

- **企业级开发** 从产品经理和设计师的角度深入浅出地介绍Android App从开发、调试到上线的企业级开发流程
- **范例丰富实用** 提供3个主流App与11个趣味开发范例，每个范例均给出设计思路与示例代码
- **技术先进，涉及面广** 包括卫星导航、Socket通信、多点触控、百叶窗动画、蓝牙技术、支付SDK、三端融合等众多热点技术



—— Android Studio 2.X/Android 7.X企业级开发 ——

Android Studio 开发实战

从零基础到App上线

欧阳焱 编著

清华大学出版社

Android StudioApp

ISBN9787302470069

.



AndroidApp

168Android StudioAppAppAppApp8App

3AppApp()App()

App()WIFI

AndroidAppAppAndroid



IBM“”iOS

21“”“App”



AppAppApp

AppAndroidiOSApp

C/C++JavaJavaAndroidAndroidADTAndroid Studio

AndroidAndroidAndroidApp

3“App”412“”

1~8Android StudioAppAppAppAppAppApp

9~16AppApp“”

1~8App9~16App“9”“11”“12

”“13”App“10”“14”“15”“16

”

SocketSDKAndroid

“”BUG

Android Studio 2.2.3API 25SDK(Android 7.1.1)

(<http://blog.csdn.net/aqi00>)

<http://pan.baidu.com/s/1dFEFEhF>()booksaga@126.com

“App”

第1章 Android Studio环境搭建



Android StudioApp“Hello World”Android StudioAppApp

1.1 Android Studio

AndroidLinuxAndroidiOSAndroid20169

Android85%iOS

AndroidAppEclipseEclipseADTEclipseJavaAndroid2015

ADTAndroid Studio

Android StudioAndroidEclipseADTAndroid StudioAndroid
Studio20161262.2.3WindowsMac OS XLinux

Android StudioEclipse

(1)Android Studiov7designEclipse

(2)SDKNDKAndroid StudioEclipse

(3)Android Studiodrawable

1.2 Android Studio

Android StudioAppAndroid Studio

1.2.1

Android StudioApp

(1)4G8G

(2)CPU1.5GHz

(3)10G

(4)USB

(5)WindowsWindows 7Windows XPjdk1.8Java 8Windows

1.2.2

Android StudioAndroidJDKSDKNDK

1. JDK

JDKJavaJava Development KitJavaAndroidJavaJDK

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>JDK1.8

AndroidJDKAndroid 5.0jdk1.7Android 7.0jdk1.8

JDK1.61.7SDK

(1)Android N requires the IDE to be running with Java 1.8 or later

(2)com/android/dx/command/dexer/MainUnsupported major.minor

version 52.0

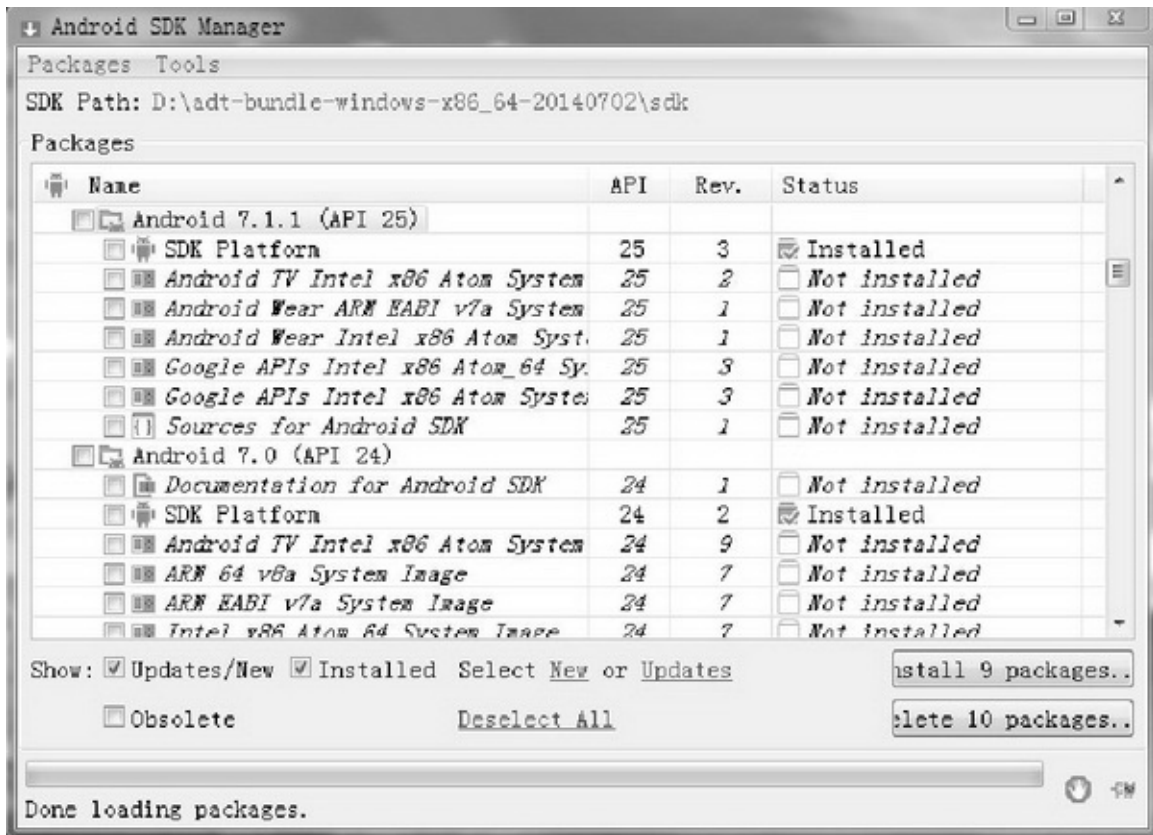
(3)AppcompileSdkVersion 'android-24' requires JDK 1.8 or later to compile.

JDKJAVA_HOMEJDKD:\Program


```
Files(x86)\Java\jdk1.8.0_102CLASSPATH
.;%JAVA_HOME%\lib\tools.jar;%JAVA_HOME%\lib\dt.jar;%JAVA_HOME%\b
Path;%JAVA_HOME%\bin
```

2. SDK

```
SDKAndroidSoftware Development KitSDKApp
build-toolsAndroid
docs
extras\androidandroid-support-v4.jarv7v13
platformsAndroid
platform-toolstoolssqlite3.exeddms.batdraw9patch.bat
samplesAndroiddemo
sourcesAndroidAPI
system-images
SDKAndroid Studiohttp://sdk.android-studio.org/Android Studio
SDKANDROID_HOME SDKD:\Android\sdkPath
;%ANDROID_HOME%\tools
SDKSDKSDK Manager.exe1-1
```

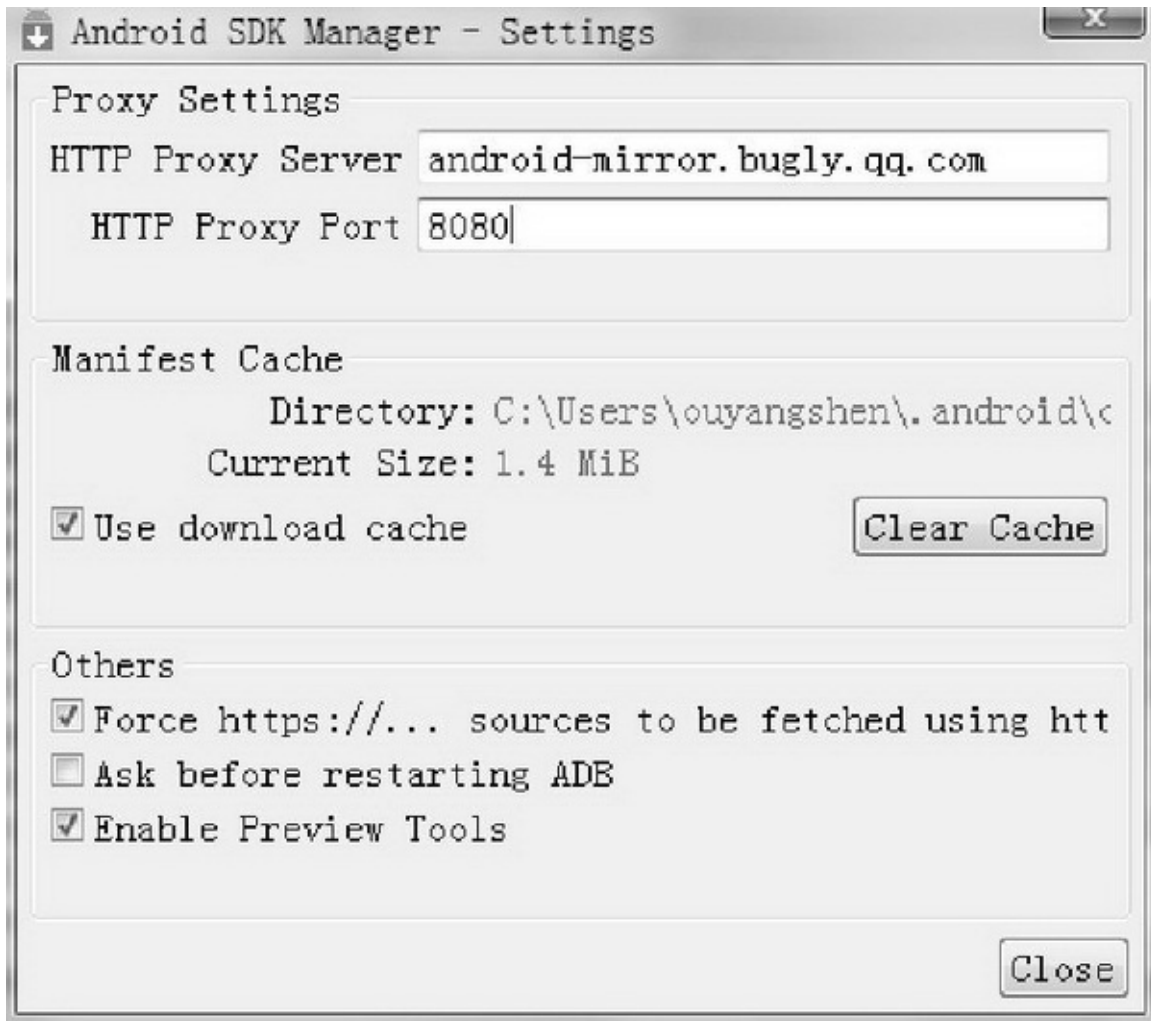


1-1 SDK Manager

Install ** packages Accept License Install

Tools → Options HTTP Proxy Server HTTP Proxy Port

Force https://... sources to be fetched using http://... 1-2



1-2SDK Manager

Buglyandroid-mirror.bugly.qq.com8080

3. NDK

NDKC/C++Native Development KitJNIC/C++soJava

JNiso

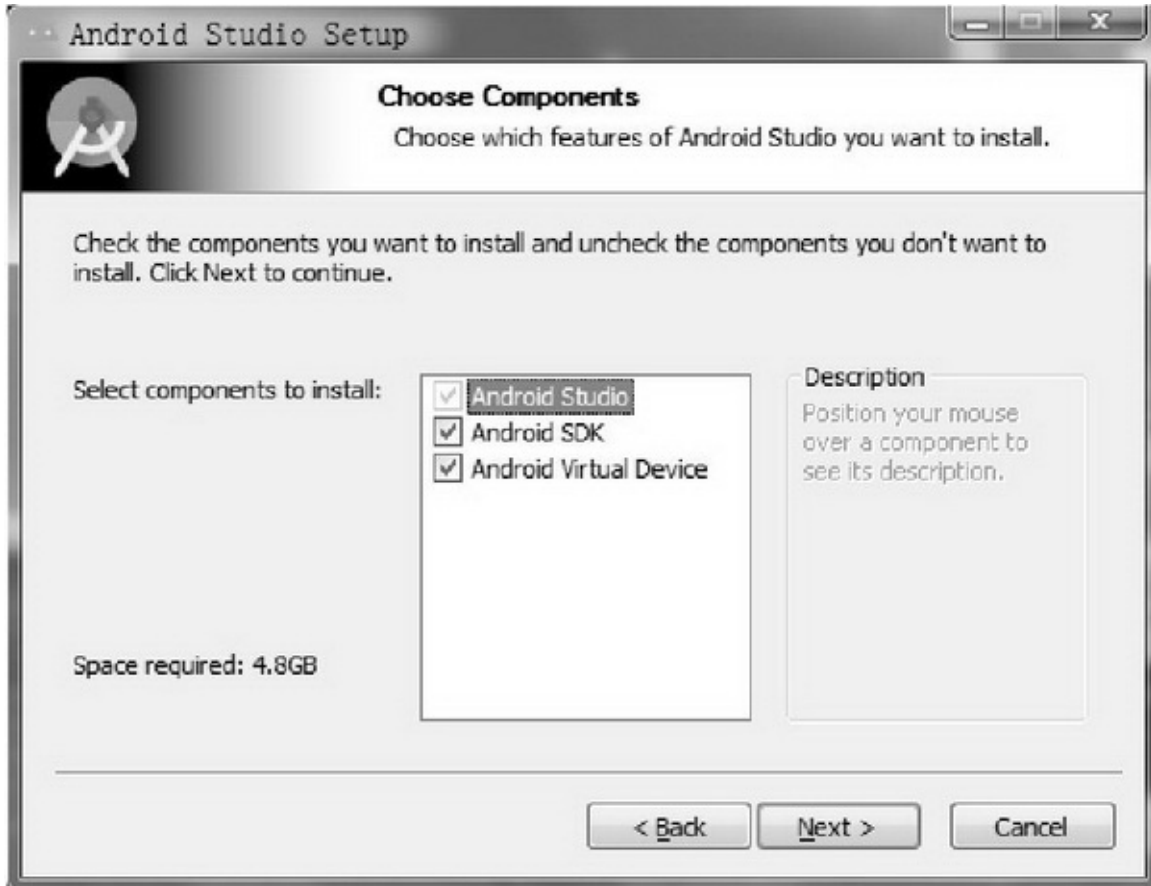
NDK14JNINDKNDK_ROOTNDKD:\android-ndk-r12b

Path;%NDK_ROOT%

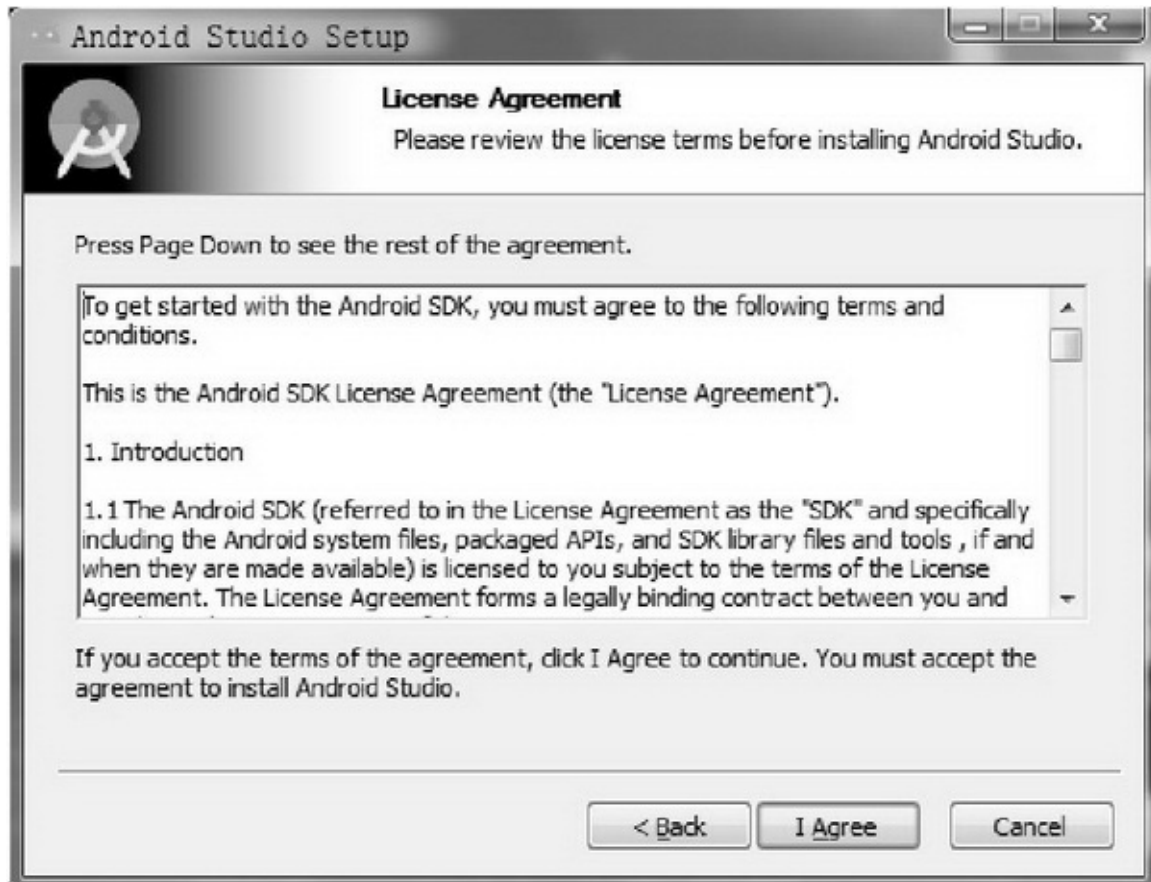
1.2.3 Android Studio 2016128 Android Studio

<https://developer.android.google.cn/studio/index.html> Android

Android Studio 1-3 Next Agree 1-4 Android
Studio SDK Android Studio SDK
Studio SDK (D) Next



1-3 Android Studio



1-4

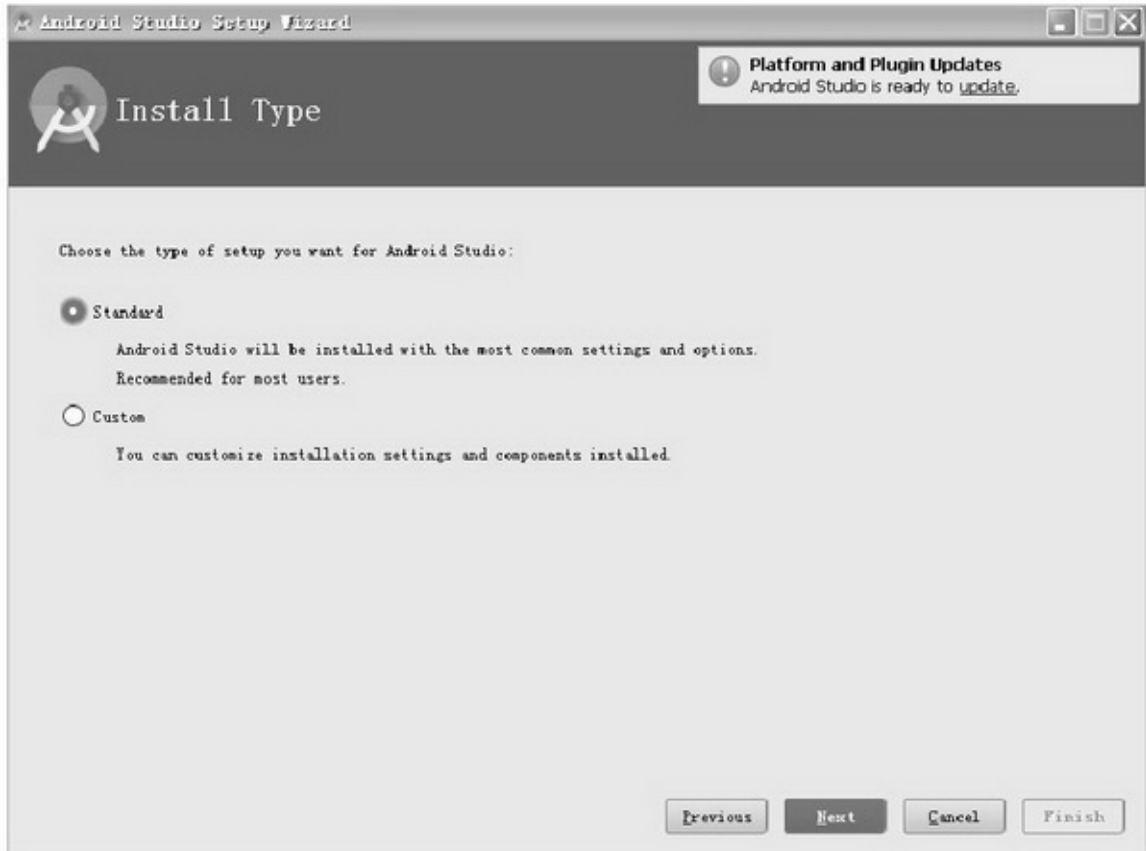
NextInstall

Android Studio1-5Next1-6StandardNextSDK

NextFinishCancelFinishWelcome to Android Studio

1-7Start a new Android Studio projectAndroid







1-7 Android Studio

(1) Your Android SDK is missing... SDK ANDROID_HOME

(2) Android Studio Unable to access Android SDK add-on list Cancel

Android Studio File → Project Structure → SDK Location JDK SDK

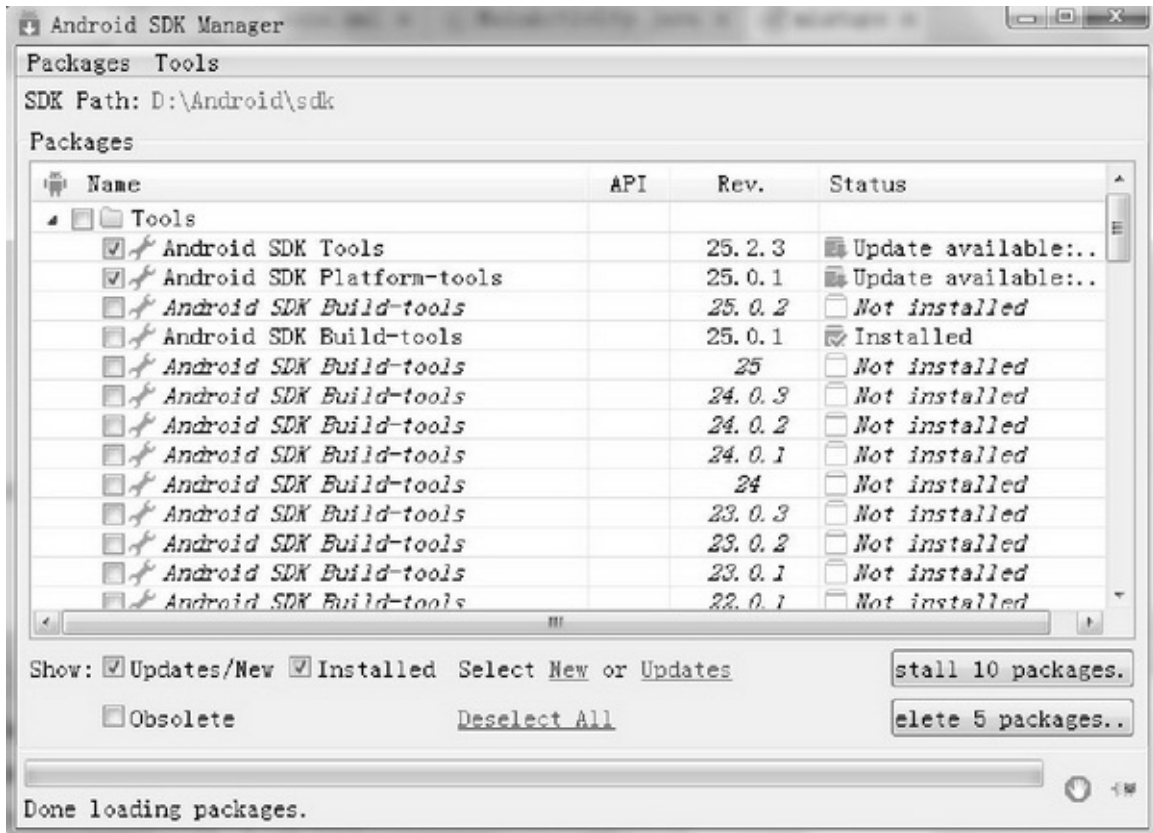
NDK Android Studio

(3) Android Studio Android Studio

Android Studio 2.2.3 SDK Android 7.1.1 (API 25) 25.0.1 SDK App

SDK

SDK SDK Manager.exe API 25 SDK 1-8



1-8SDK Manager

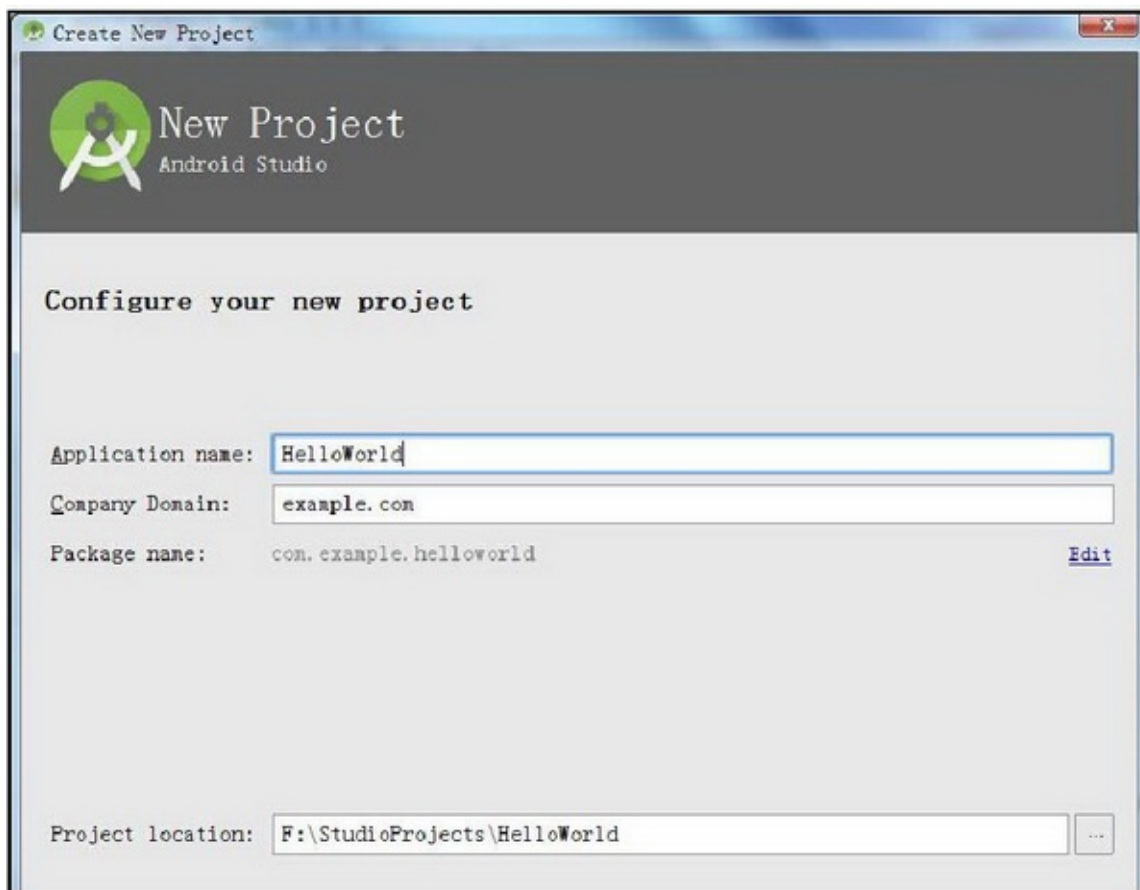
ToolsAndroid7.0(API 24)SDK PlatformAndroid6.0(API 23)SDK
 PlatformAndroid5.1.1(API 22)SDK PlatformAndroid5.0.1(API 21)SDK
 PlatformAndroid4.4.2(API 19)SDK Platform“Install ** packages...”

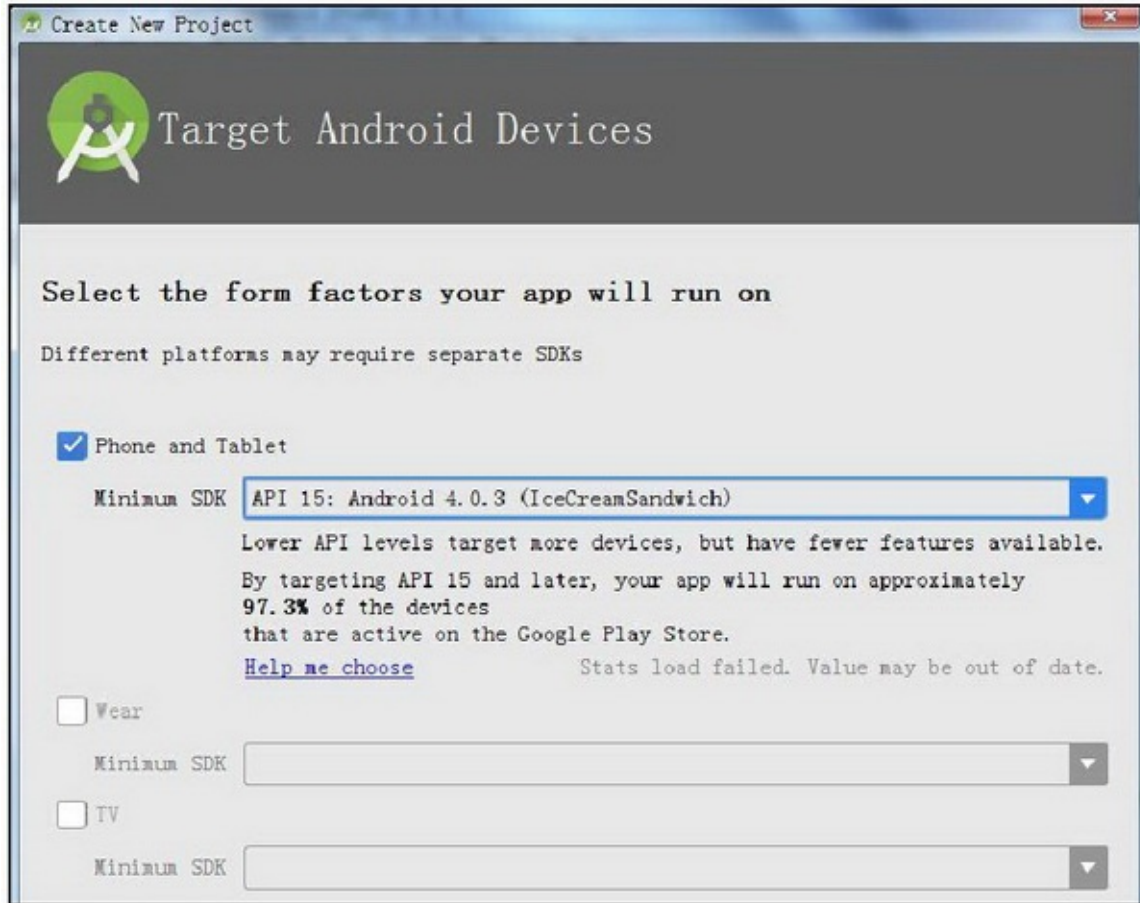
1.3 Hello World Android Studio App

Hello World

1.3.1

Android Studio File → New → New Project Create New Project 1-9
Application name Company Domain Next
1-10 App App SDK Minimum SDK Phone and Tablet
API 15 (97.3%) Next 1-11 Empty Activity Next



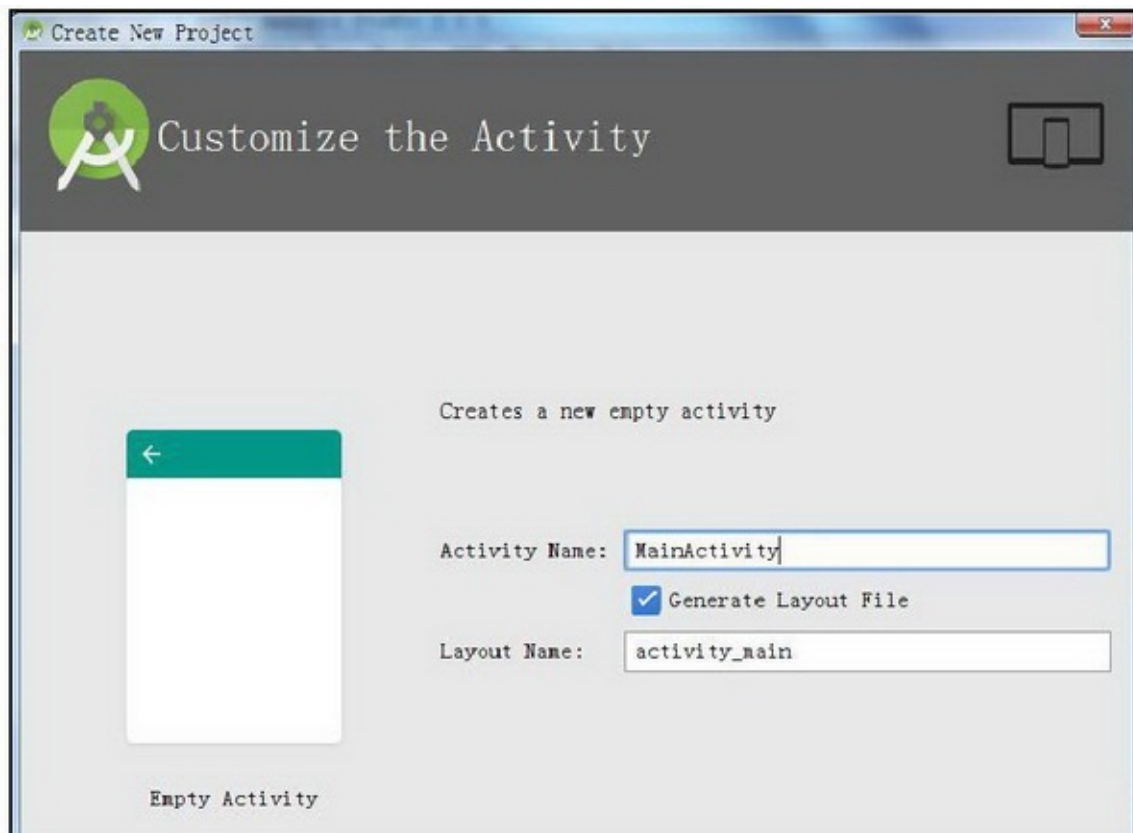


1-10

1-12(Activity Name)(Layout Name)OK



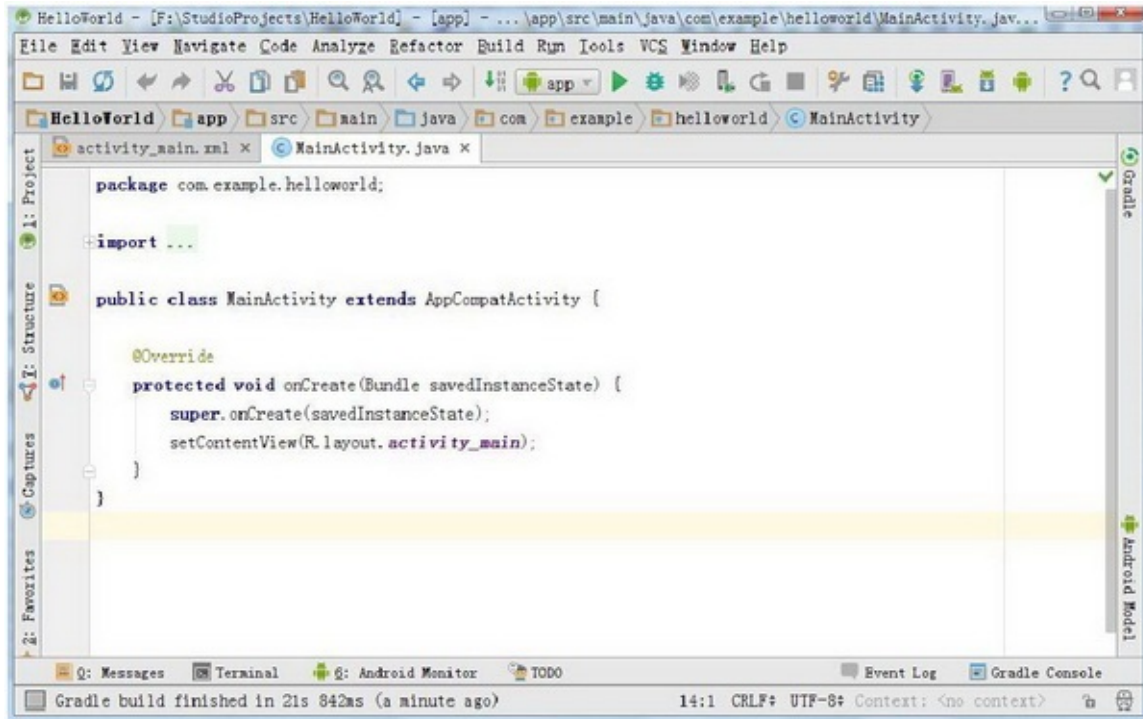
1-11Activity



1-12

Android Studio activity_main.xml MainActivity.java MainActivity.java

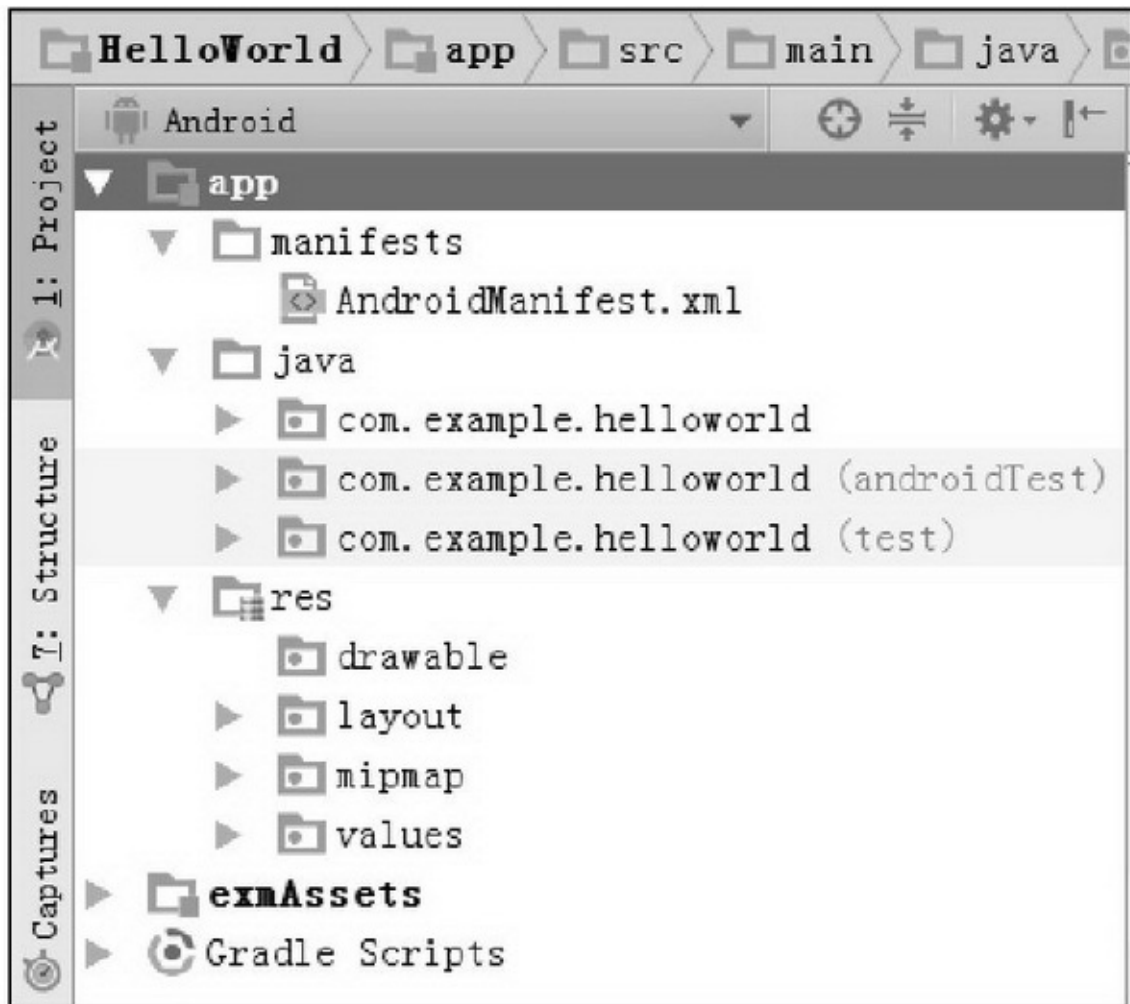
1-13



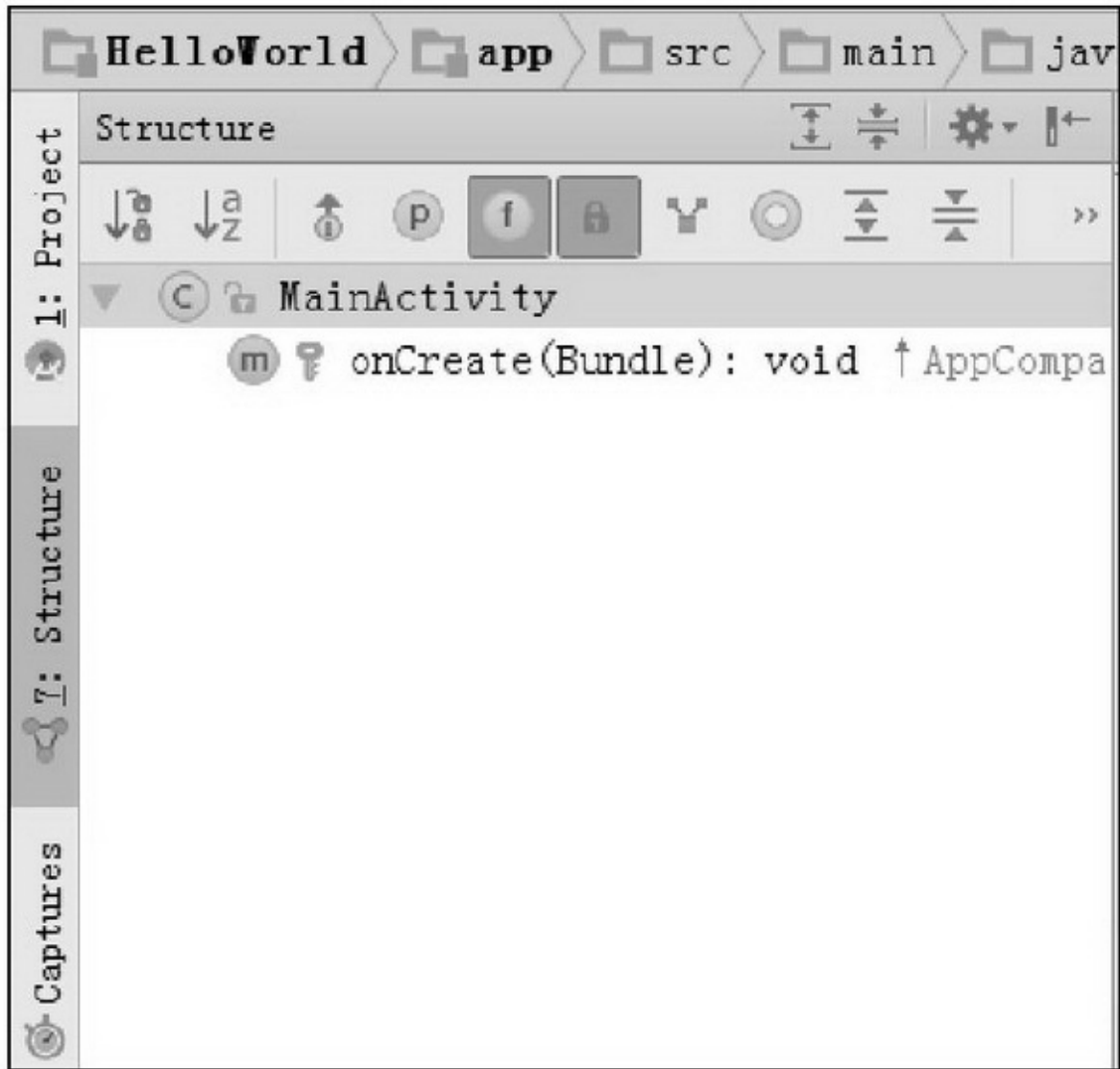
1-13 MainActivity

MainActivity.java Project Structure Captures Favorites Project

1-14 Structure 1-15

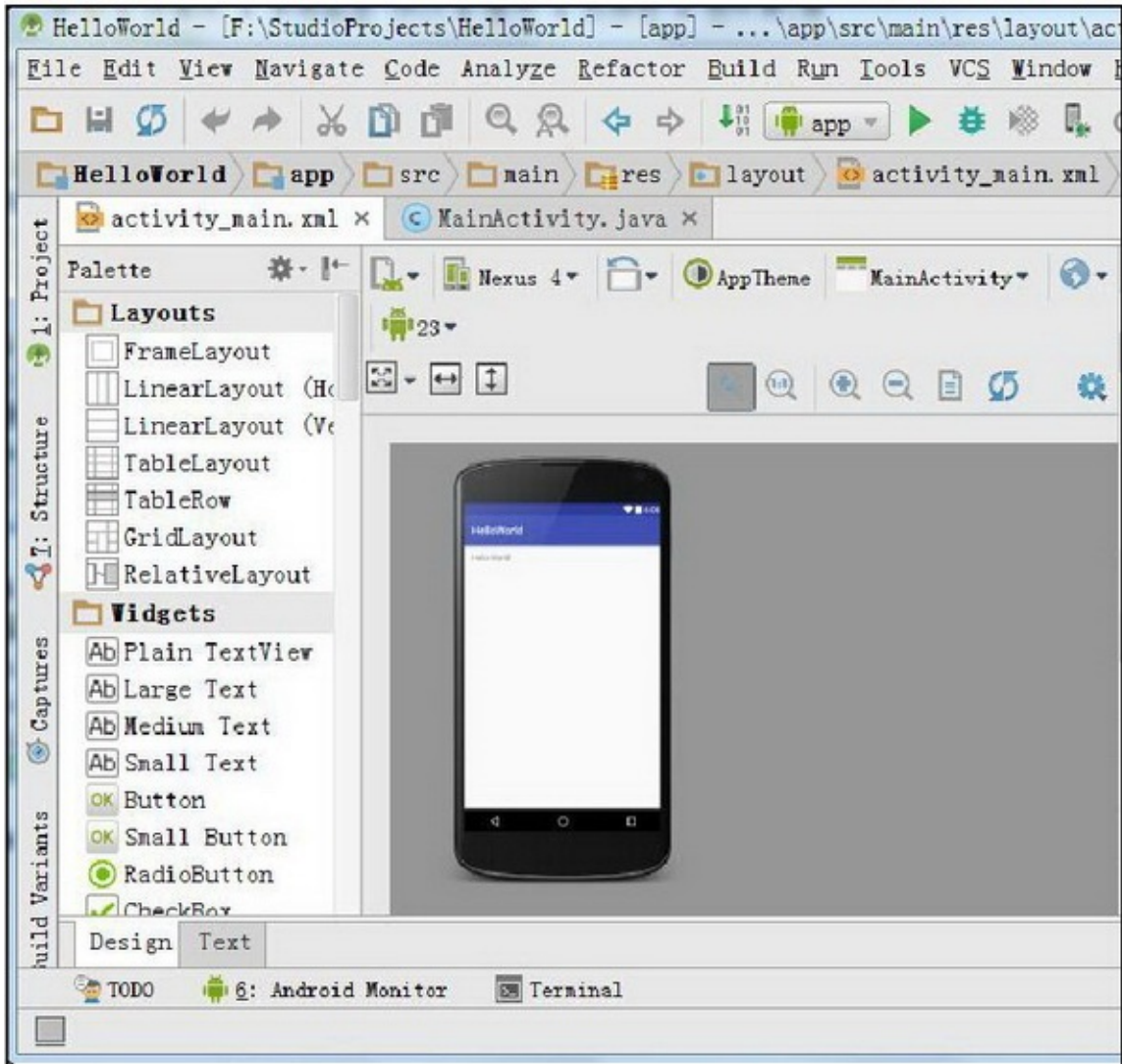


1-14HelloWorld

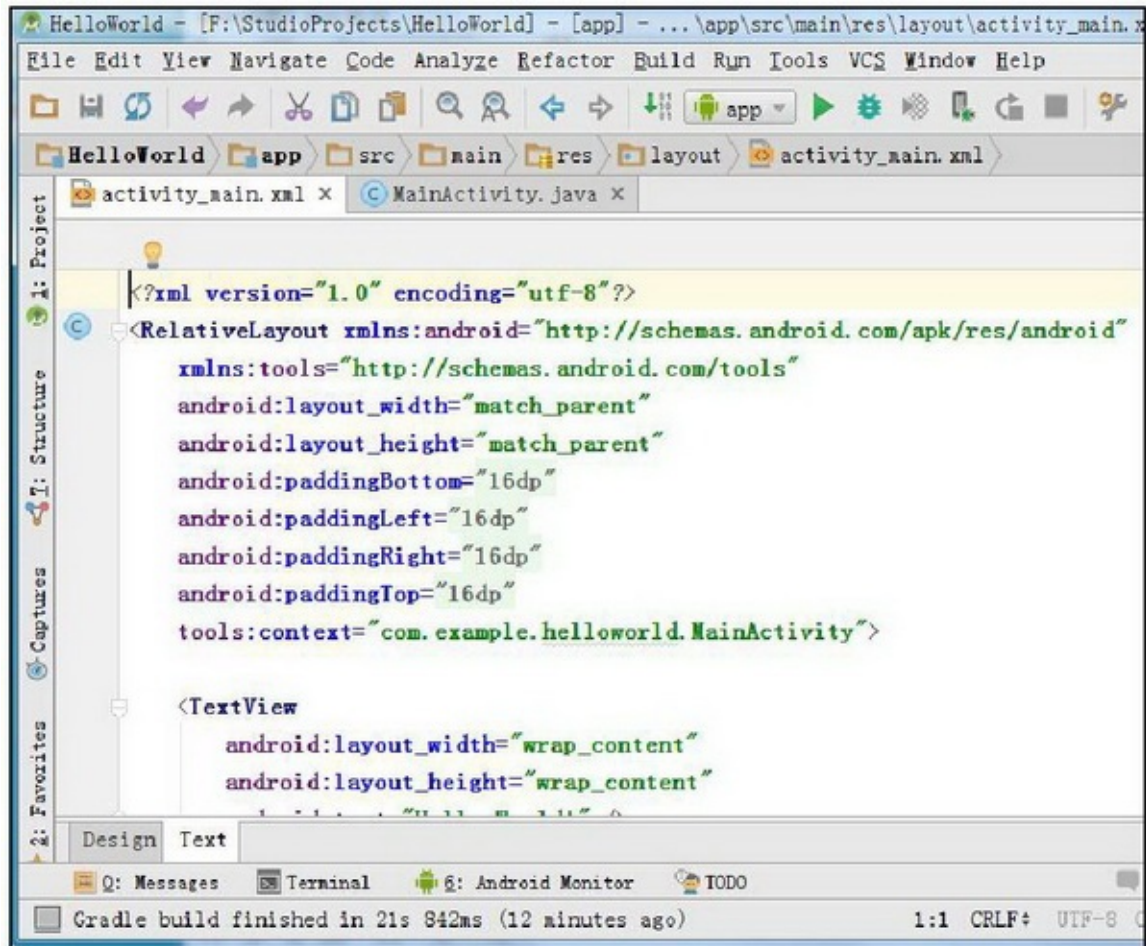


1-15MainActivity

activity_main.xml
1-16PalettePaletteDesign(
)Text()
1-17XMLApp



1-16activity.xml



1-17activity.xml

activity_main.xml

(1)Android N requires the IDE to be running with Java 1.8 or later

jdk1.8(jdk1.6jdk1.7)

jdk1.8JDK

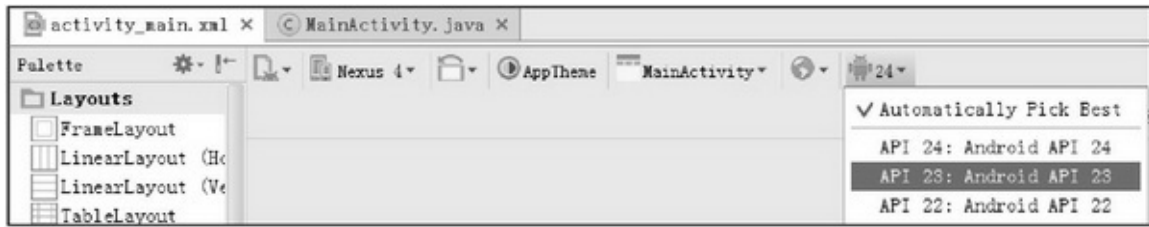
(2)Rendering Problems Exception raised during rendering

com/android/util/PropertiesMap (Details)Rendering Problems Exception raised

during renderingcom.android.ide.common.rendering.api.LayoutlibCallback

Android Studiobug

API 231-18



1-18

1.3.2/

Android StudioEclipseAndroid Studio3

(1)Build → Make Project(2)Build → Make Module ***

(3)Build → Clean ProjectBuild → Rebuild Project



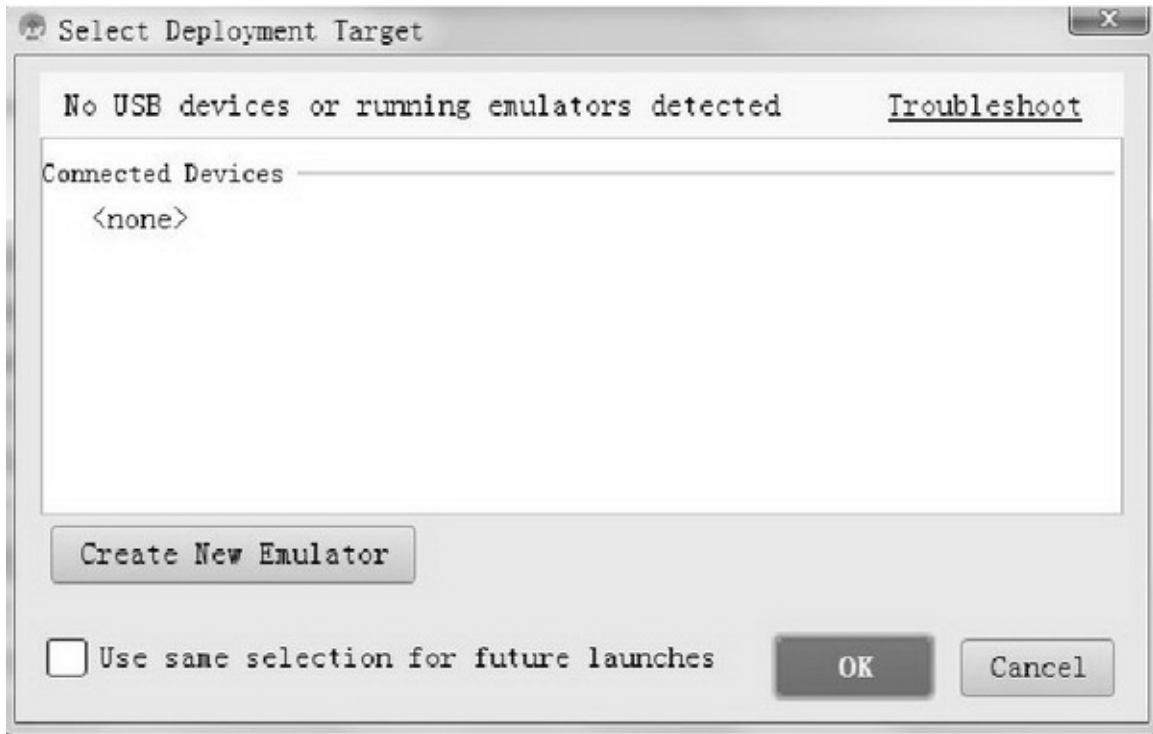
1-19

1-194AVD ManagerSDK ManagerSDK

1.3.3

AppAppRun → Run 'app'(Shift+F10)Android Studio

Select Deployment Target1-20



1-20App

Create New Emulator1-21Next


Virtual Device Configuration

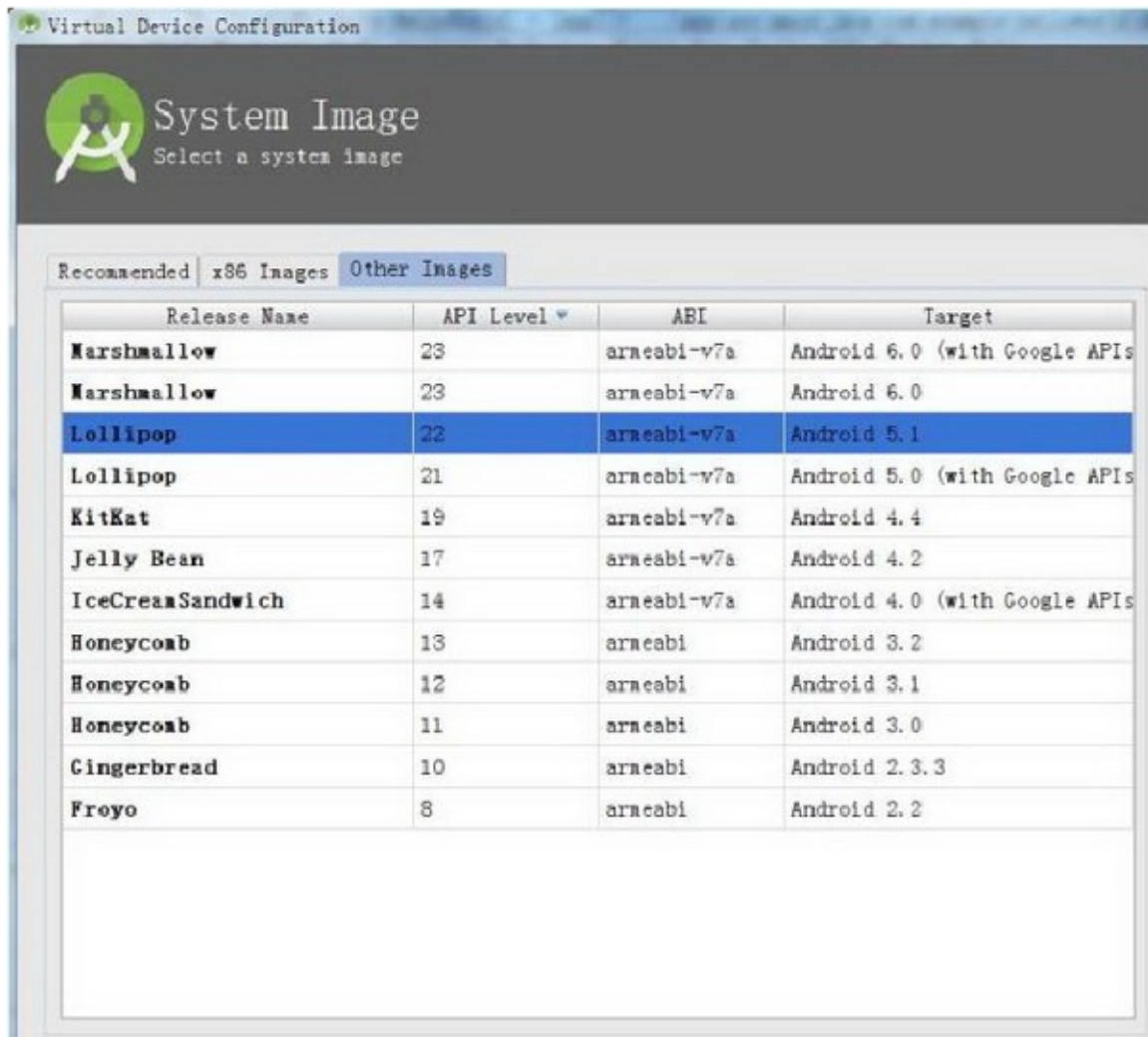
Select Hardware

Choose a device definition

Q-

Category	Name	Size	Resolution	Density
TV	Nexus S	4.0"	480x800	hdpi
Wear	Nexus One	3.7"	480x800	hdpi
Phone	Nexus 6P	5.7"	1440x2560	560dpi
Tablet	Nexus 6	5.96"	1440x2560	560dpi
	Nexus 5X	5.2"	1080x1920	420dpi
	Nexus 5	4.95"	1080x1920	xxhdpi
	Nexus 4	4.7"	768x1280	xhdpi
	Galaxy Nexus	4.65"	720x1280	xhdpi
	Android Wear Square	1.65"	280x280	hdpi
	Android Wear Round	1.65"	320x320	hdpi
	5.4" HWCA	5.4"	480x854	mdpi

New Hardware Profile Import Hardware Profiles 



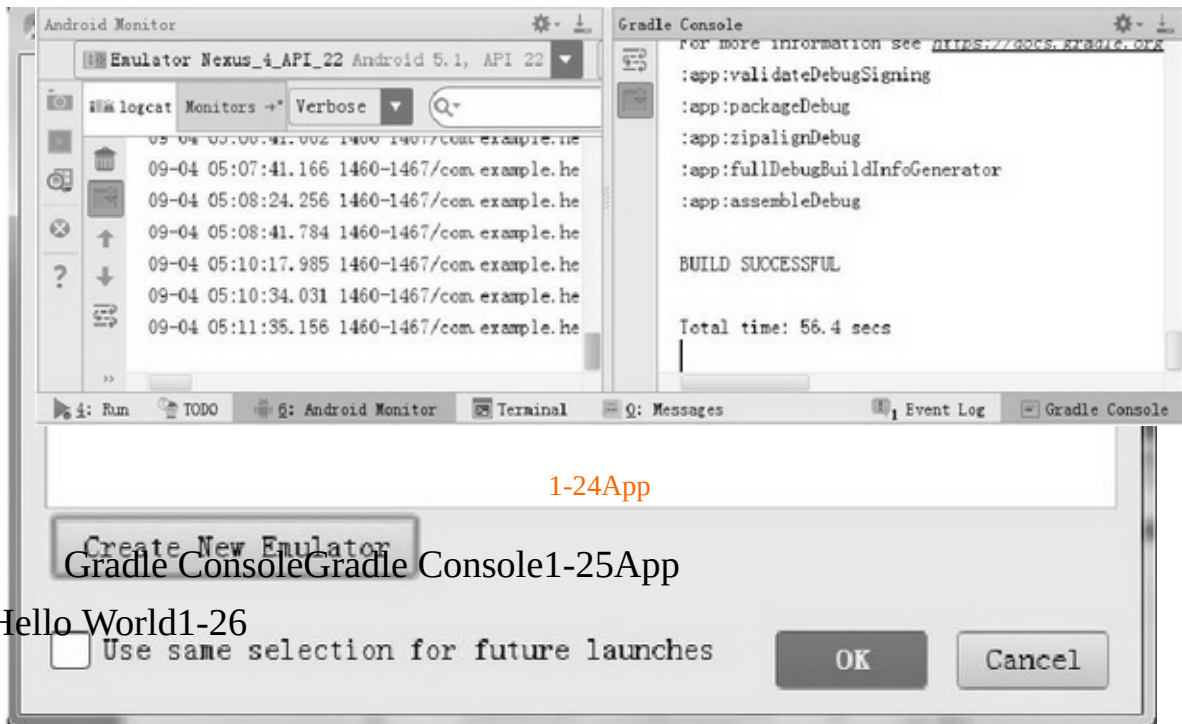
1-22SDK

SDK1-223Other ImagesLollipop(Android 5.1)Android 5.1
 5SDKAPI 19
 NextFinish

1.3.4App

1-23

Run → Run 'app'Nexus 4 API 22-23
 OKAndroid Studio1-24logcatAppGradle
 ConsoleApp



5554:Nexus_4_API_22



1-25



1-26HelloWorld

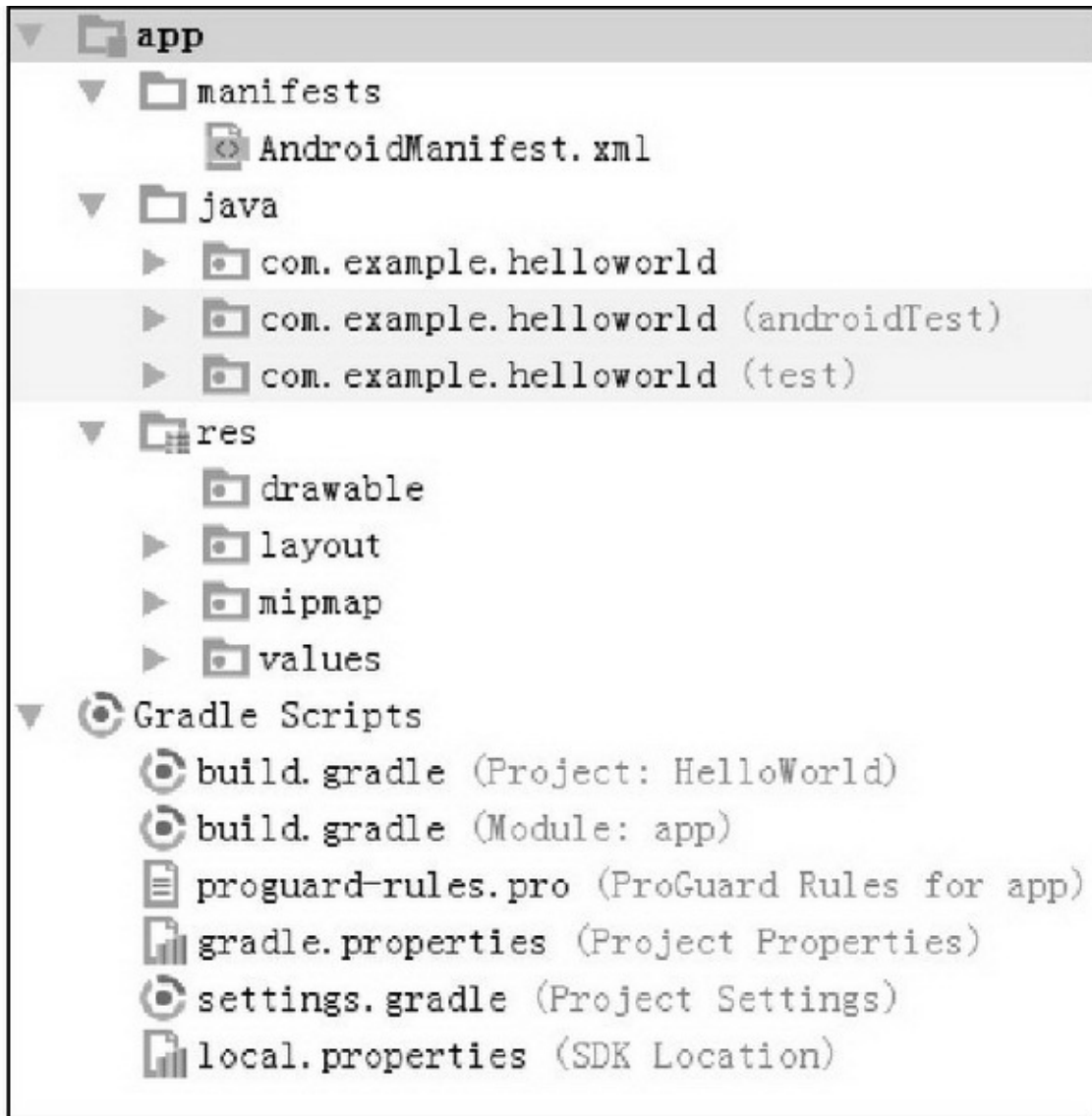
AppHello World AppAndroid StudioApp—Hello
WorldAndroid

1.4App

App(Hello Wolrd)App

1.4.1

Android StudioFile → New → New ProjectEclipseworkspace
File → New → New ModuleAppEclipseprojectAndroid StudioNew
ProjectAppNew ModuleApp
1-27HelloWorld



1-27HelloWorld

appGradle Scriptsapp3

(1)manifestsxmlAndroidManifest.xmlApp

(2)java3com.example.hellorworldAppjavajava

(3)resAppres4

drawable

layoutApp

mipmap

valuesstrings.xmldimens.xmlcolors.xmlstyles.xml

Gradle Scripts

(1)build.gradleApp

(2)proguard-rules.projava

(3)gradle.properties

(4)settings.gradleinclude ':app'App

(5)local.propertiesSDKNDK

1.4.2build.gradle build.gradlebuild.gradlebuild.gradle

```
apply plugin: 'com.android.application'

android {
    // 指定编译用的 SDK 版本号，如 25 表示使用 Android 7.1 编译
    compileSdkVersion 25
    // 指定编译工具的版本号。这里的头两位数字必须与 compileSdkVersion 保持一致，具体的版本号
    // 可在 sdk 安装目录的 sdk\build-tools 下找到
    buildToolsVersion "25.0.2"

    defaultConfig {
        // 指定该模块的应用编号，即 App 的包名。该参数为自动生成，无须修改
        applicationId "com.example.helloworld"
        // 指定 App 适合运行的最小 SDK 版本号，如 15 表示至少要在 Android 4.0.3 上运行
        minSdkVersion 15
        // 指定目标设备的 SDK 版本号，即该 App 最希望在哪个版本的 Android 上运行
        targetSdkVersion 25
        // 指定 App 的应用版本号
        versionCode 1
        // 指定 App 的应用版本名称
        versionName "1.0"
    }
    buildTypes {
        release {
            // 指定是否开启代码混淆功能。true 表示开启混淆，false 表示无须混淆。
            minifyEnabled false
            // 指定代码混淆规则文件的文件名
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}

// 指定 App 编译的依赖信息
dependencies {
    // 指定引用 jar 包的路径
    compile fileTree(dir: 'libs', include: ['*.jar'])
    // 指定单元测试编译用的 junit 版本号
    testCompile 'junit:junit:4.12'
    // 指定编译 Android 的高版本支持库，如 AppCompatActivity 必须指定编译 appcompat-v7 库
    compile 'com.android.support:appcompat-v7:25.1.0'
}
```

1.4.3 AppManifest.xml AndroidManifest.xml AppXML manifest package Appmanifest

(1) uses-sdk android:minSdkVersion android:targetSdkVersion Eclipse
App build.gradle Android Studio uses-sdk

(2) uses-permission App

(3) application App

android:allowBackup true false

android:icon App

android:label App

android:supportRtl true /

android:theme App

application activity service receiver provider

1.4.4

Hello World Android Studio activity.xml MainActivity.java
activity.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.helloworld.MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!" />

</RelativeLayout>
```

xml RelativeLayout TextView Hello World App Hello World

TextViewandroid:textHello World""I Love Android

Run → Run 'app'App

MainActivity.javajavaMainActivity

```
public class MainActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
    }  
}
```

MainActivity.javaMainActivityonCreateonCreate

setContentViewactivity_mainAppactivity.xml""

activity.xmlTextViewandroid:id="@+id/tv_hello"MainActivity.java

setContentView

//tv_helloTextView

TextView tv_hello = (TextView) findViewById(R.id.tv_hello); //TextView

tv_hello.setText("");

//TextView

tv_hello.setTextColor(Color.RED);

//TextView

tv_hello.setTextSize(30);

Run → Run 'app'App1-28

App

5554:Nexus_4_API_22



1-28HelloWorld

1.5

AndroidAndroid2

1.5.1

EclipsejavaAndroid StudioAndroid StudioApp

Ctrl+S

Ctrl+Z

Ctrl+Shift+ZCtrl+YEclipseUEStudioAndroid StudioCtrl+Y

Ctrl+Y

Ctrl+C

Ctrl+X

Ctrl+V

Ctrl+A

Delete

Ctrl+F

Ctrl+R

Ctrl+/()

Ctrl+Shift+/ (“/*” “*/”)

Ctrl+Alt+LQQ(QQ)QQ

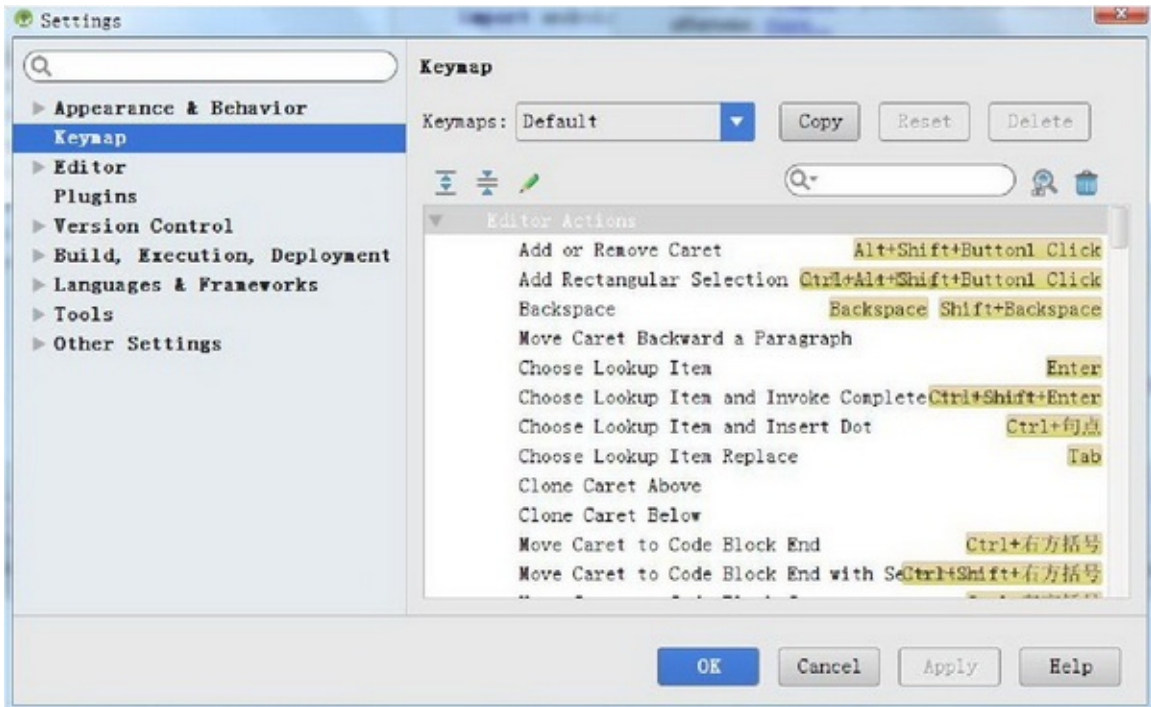
Shift+F6F2WnidowsEclipse

Alt+Enter

Shift+F10

Ctrl+F5

Android StudioFile → SettingsKeymap1-29



1-29

Add Keyboard ShortcutOK

1.5.2SVN

AppAppAppSVN

Android StudioSVN(Subversion)SVN

步骤 01

TortoiseSVN

TortoiseSVNcommand line client toolsbinsvn.exe

步骤 02

Android StudioTortoiseSVN

Android StudioFile → Settings → Version Control → Subversion → user
command line clientsvn.exe

步骤 03

Android StudioSVN

Android StudioVCS → Checkout from Version Control → Subversion

RepositoriesSVNOKCheckout

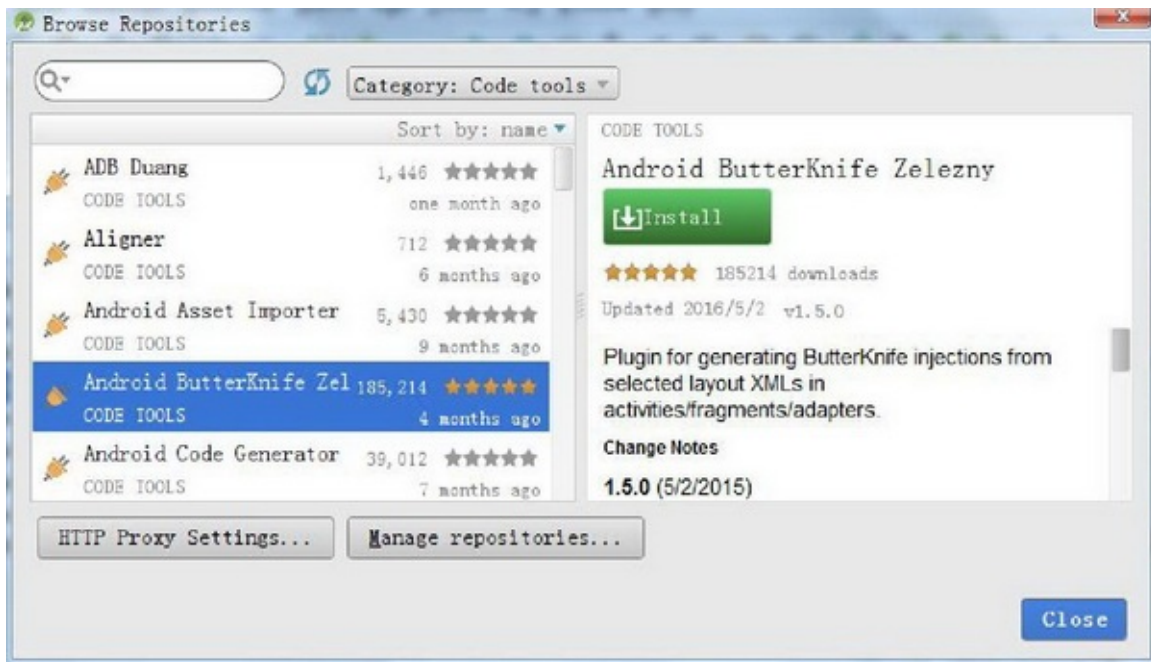
SVNSVNSVN/

(1)SVNSubversion → Commit File...SVN

(2)SVNSubversion → Update File...SVN

1.5.3

Android StudioEclipseFile → Settings → Plugins → Browser
repositories...1-30



1-30

CategoryCode toolsInstallStudio5Studio

1. Android Parcelable code generator

ParcelableAlt+InsertParcelable1-31Parcelable

2. Android Code Generator

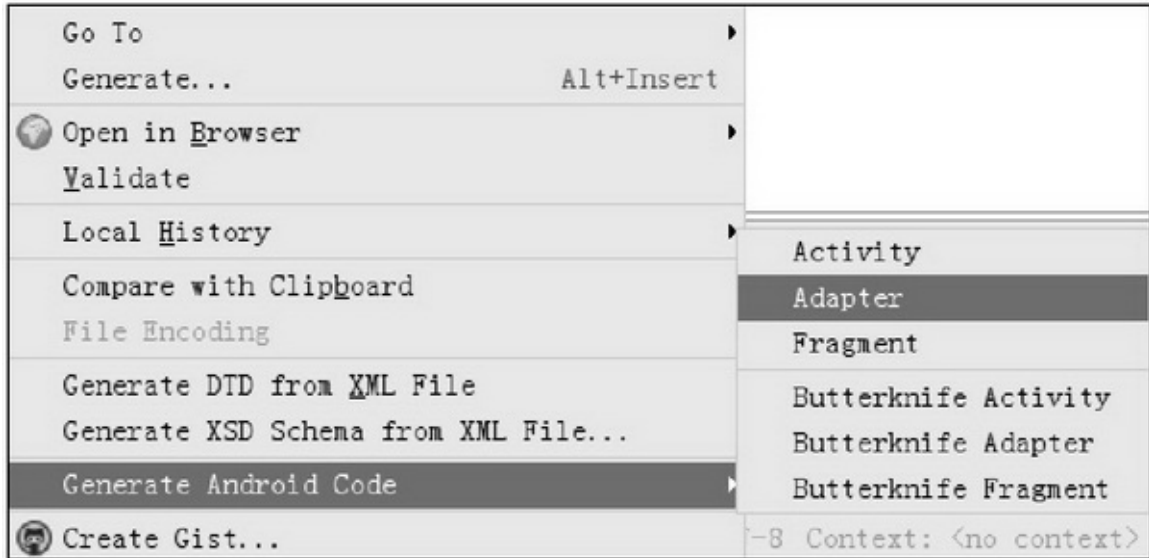
ActivityFragmentAdapterMenuGenerate Android Code1-32xml

```
public class Person {  
    public String id;  
    public String name;  
    public int age;  
}
```

Generate

GsonFormat	Alt+S
------------	-------

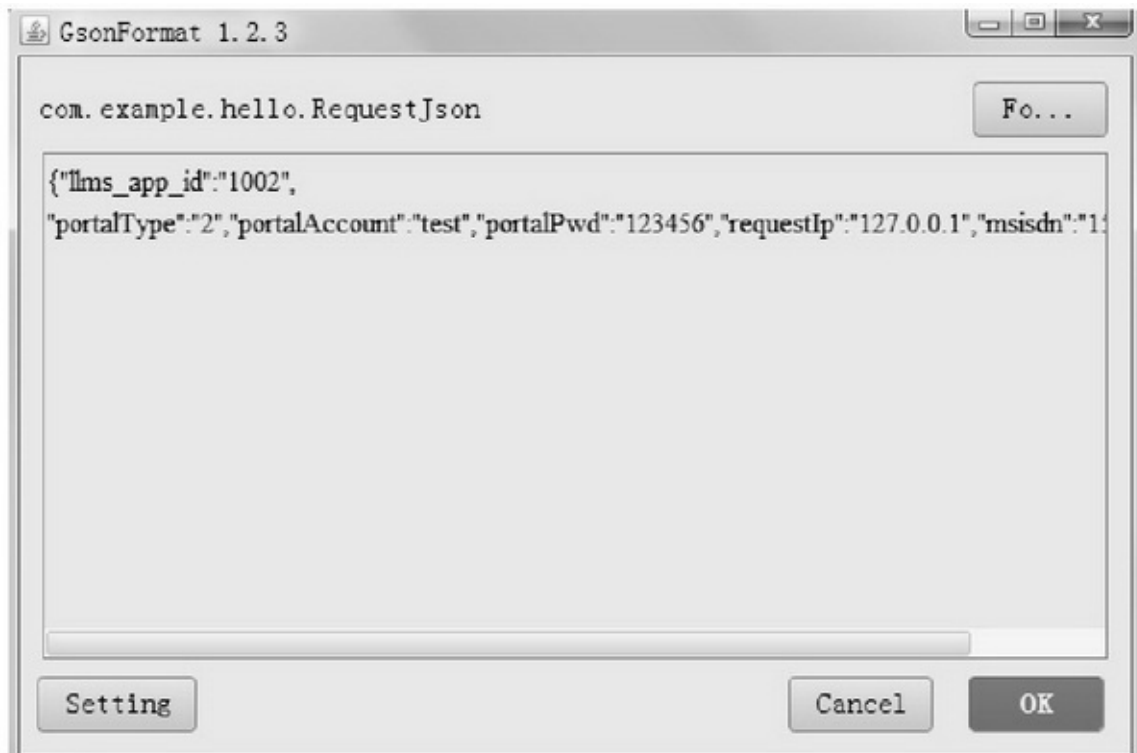
- Constructor
- Getter
- Setter
- Getter and Setter
- equals() and hashCode()
- toString()
- Override Methods... Ctrl+O
- Delegate Methods...
- Copyright
- Parcelable**



1-32Generate Android Code

3. GsonFormat

jsonsetgetAlt+SjsonjsonOKGsonFormat1-33



4. Android Postfix Completion

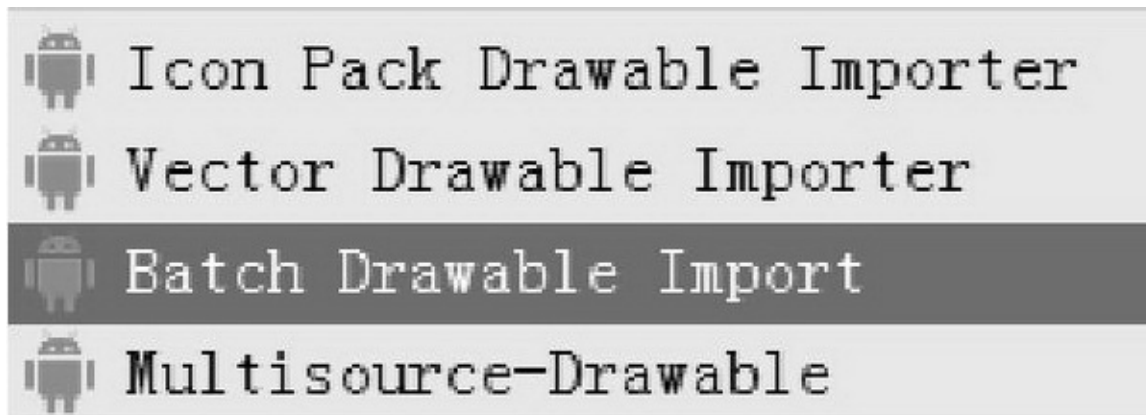
ToastLog.toastToast.makeText.logLog.d1-
34

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    //获取名字为tv_hello的TextView控件
    TextView tv_hello = (TextView) findViewById(R.id.tv_hello);
    //给TextView控件设置文字内容
    tv_hello.setText("今天天气真热啊，火辣辣的");
    //给TextView控件设置文字颜色
    tv_hello.setTextColor(Color.RED);
    //给TextView控件设置文字大小
    tv_hel
        log                               Log.d(TAG, expr)
        logd                               if(BuildConfig.DEBUG) Log.d(TAG, expr);
    "测试日志".log
}

```

5. Android Drawable Importer



1-35Drawable

New*** Drawable Importer1-35

Batch Drawable ImportOKdrawabe

1.5.4ADT

Android StudioAppAppADTADTADTAndroid Studio

ADTAndroid StudioFile → New → Import ModuleADT

FinishAndroid StudioADTApp

ADT“Error:(1, 1)\uffeff”BOMutf8EclipseAndroid

StudioBOMutf8UEStudioBOMutf8Android Studio

1.6

App—— Android StudioAndroid Studio3JDKSDKNDK
HelloWorldAppAppbuild.gradleApp
AndroidManifest.xmlSVNADTAndroid Studio
Android StudioAndroid StudioApp

第2章 初级控件



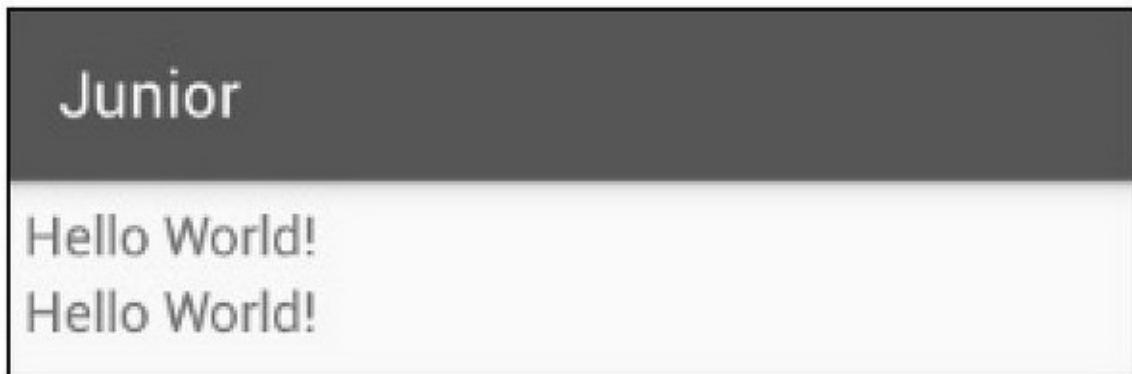
Android“”

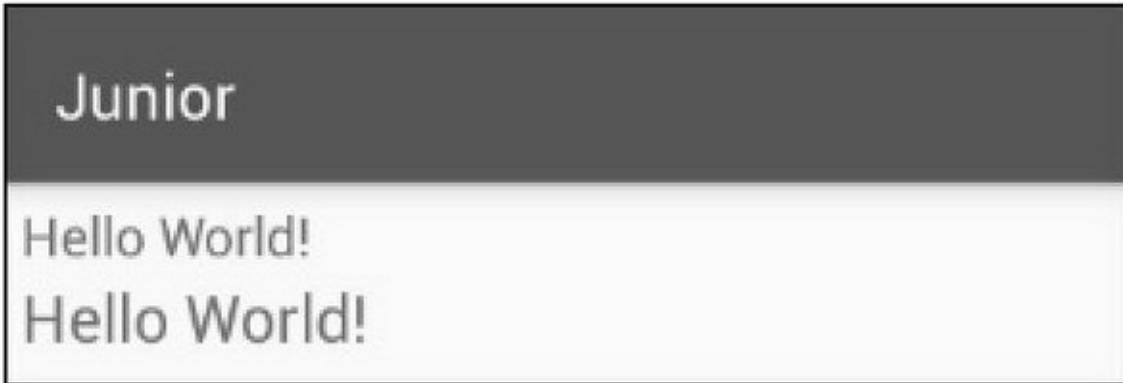
2.1

2.1.1

“”AndroidApp

Androidpx()in()mm()pt(1/72)dp()dip(
dp)sp()pxdpsp
px(5)px
dpdpdp
spdp()dpspdpsp18dp18sp
2-118dp18sp2-2





2-2

dpsp
dpppxDisplayMetricsdensitydensity=1.0dppxdensity=1.5
dp3pxdensity=2.0dppx

```
public class Utils {  
    //根据手机的分辨率从 dp 的单位转成为 px(像素)  
    public static int dip2px(Context context, float dpValue) {  
        final float scale = context.getResources().getDisplayMetrics().density;  
        return (int) (dpValue * scale + 0.5f);  
    }  
  
    //根据手机的分辨率从 px(像素) 的单位 转成为 dp  
    public static int px2dip(Context context, float pxValue) {  
        final float scale = context.getResources().getDisplayMetrics().density;  
        return (int) (pxValue / scale + 0.5f);  
    }  
}
```

XMLspdpAndroidpxLayoutParamwidthheight
setMarginssetPaddingpxdpdppx
int dip_10 = Utils.dip2px(this, 10L);
TextView tv_padding = (TextView) findViewById(R.id.tv_padding);
tv_padding.setPadding(dip_10, dip_10, dip_10, dip_10); 2.1.2
AndroidalphaRGB()FFEEDDCCFFEEDD
CCFF00RGB
XML(FF)(00)

```

TextView tv_code_six = (TextView) findViewById(R.id.tv_code_six);
tv_code_six.setBackgroundColor(0x00ff00);
TextView tv_code_eight = (TextView) findViewById(R.id.tv_code_eight);
tv_code_eight.setBackgroundColor(0xff00ff00); 2-3

```



2-3

Android3

1.

Android12Color2-1

Color 类中的颜色类型	说明	Color 类中的颜色类型	说明
BLACK	黑色	GREEN	绿色
DKGRAY	深灰	BLUE	蓝色
GRAY	灰色	YELLOW	黄色
LTGRAY	浅灰	CYAN	青色
WHITE	白色	MAGENTA	玫红
RED	红色	TRANSPARENT	透明

2-1

2.

```
“#”android:textColor="#000000"(setTextColor(0xff00ff00);)
```

```
Color.rgb(int red, int green, int blue)Color.argb(int alpha, int red, int green, int blue)
```

3.colors.xml

```
res/values/colors.xmlXML“@color/”XML
```

getResources().getColor(R.color.)

2.1.3

App()DisplayMetricsDisplayMetrics

widthPixelspx

heightPixelspx

densitydppx

```
public class DisplayUtil {
    public static int getSreenWidth(Context ctx) {
        WindowManager wm = (WindowManager) ctx.getSystemService(Context.WINDOW_SERVICE);
        DisplayMetrics dm = new DisplayMetrics();
        wm.getDefaultDisplay().getMetrics(dm);
        return dm.widthPixels;
    }

    public static int getSreenHeight(Context ctx) {
        WindowManager wm = (WindowManager) ctx.getSystemService(Context.WINDOW_SERVICE);
        DisplayMetrics dm = new DisplayMetrics();
        wm.getDefaultDisplay().getMetrics(dm);
        return dm.heightPixels;
    }

    public static float getSreenDensity(Context ctx) {
        WindowManager wm = (WindowManager) ctx.getSystemService(Context.WINDOW_SERVICE);
        DisplayMetrics dm = new DisplayMetrics();
        wm.getDefaultDisplay().getMetrics(dm);
        return dm.density;
    }
}
```

2-45720*12802

Junior

当前屏幕的宽度是720px，高度
是1280px，像素密度是2.000000

2.2

AndroidViewLinearLayoutScrollView

2.2.1 View

ViewAndroidViewView

XML

id

layout_widthdpmatch_parentwrap_content()

layout_heightlayout_width

layout_margin()layout_marginTop

layout_marginBottomlayout_marginLeftlayout_marginRight

minWidth

minHeight

background

layout_gravity2-2“|”

2-2

XML 中的对齐方式	Gravity 类中的对齐方式	说明
left	LEFT	靠左对齐
right	RIGHT	靠右对齐
top	TOP	向上对齐
bottom	BOTTOM	向下对齐
center	CENTER	居中对齐
center_horizontal	CENTER_HORIZONTAL	水平方向居中
center_vertical	CENTER_VERTICAL	垂直方向居中

paddingpaddingToppaddingBottompaddingLeftpaddingRight

visibility2-3

XML 中的可视类型	View 类中的可视类型	说明
visible	VISIBLE	可见。默认值
invisible	INVISIBLE	不可见。虽然看不到但还占着位置
gone	GONE	消失。不仅看不到而且不占位置了

setLayoutParamsLayoutParams.MATCH_PARENT

LayoutParams.WRAP_CONTENTsetMargins

setMinimumWidth

setMinimumHeight

setBackgroundColor

setBackgroundDrawable

setBackgroundResourceid

setPadding

setVisibility2-3

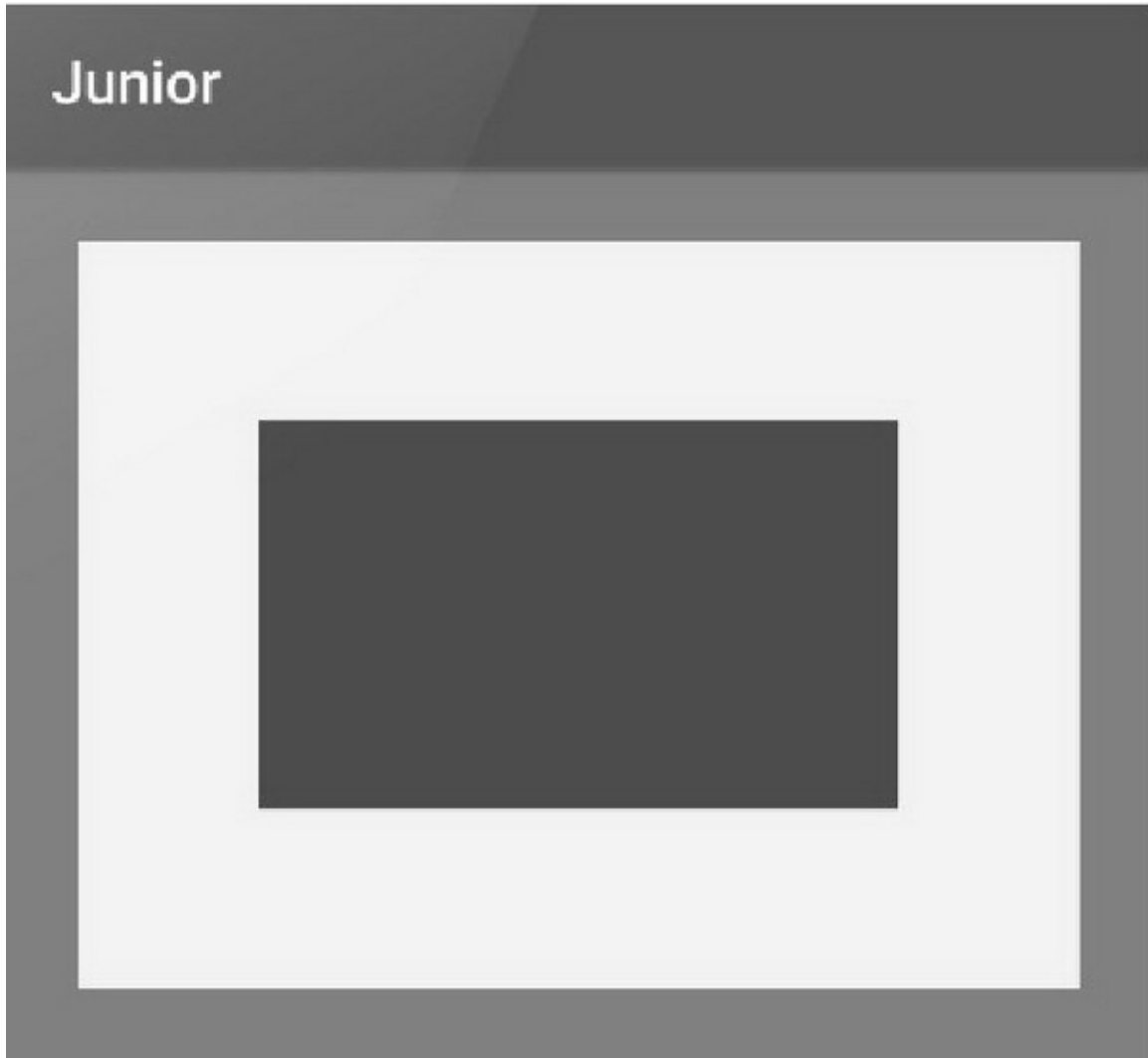
marginpaddingmarginpadding

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="300dp"
    android:background="#00aaff"
    android:orientation="vertical"
    android:padding="5dp" >

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_margin="20dp"
        android:background="#ffff99"
        android:padding="60dp" >

        <View
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:background="#ff0000" />

    </LinearLayout>
</LinearLayout>
```



2-5marginpadding

```
2-5LinearLayoutmargin20dppadding60dpmarginpadding  
layout_marginpadding  
ViewGroupAndroid()
```

```
ViewGroup33  
addView  
removeView  
removeAllViews
```


2.2.2LinearLayout LinearLayoutLinearLayout

LinearLayoutLinearLayoutView/ViewGroupXML

orientationhorizontalverticalhorizontalApp

orientation

gravitylayout_gravity

layout_weightlayout_weightLinearLayout

layout_weightlayout_width0dplayout_weightlayout_height0dp

LinearLayout

setOrientationLinearLayout.HORIZONTAL

LinearLayout.VERTICAL

setGravity2-2

layout_gravitygravitylayout_gravitygravityXML

layout_gravitybottomgravityleftlayout_gravitytopgravityright

```

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="300dp"
    android:background="#fff99"
    android:orientation="horizontal"
    android:padding="5dp" >

    <LinearLayout
        android:layout_width="0dp"
        android:layout_height="200dp"
        android:layout_weight="1"
        android:layout_gravity="bottom"
        android:gravity="left"
        android:background="#ff0000"
        android:layout_margin="10dp"
        android:padding="10dp"
        android:orientation="vertical">

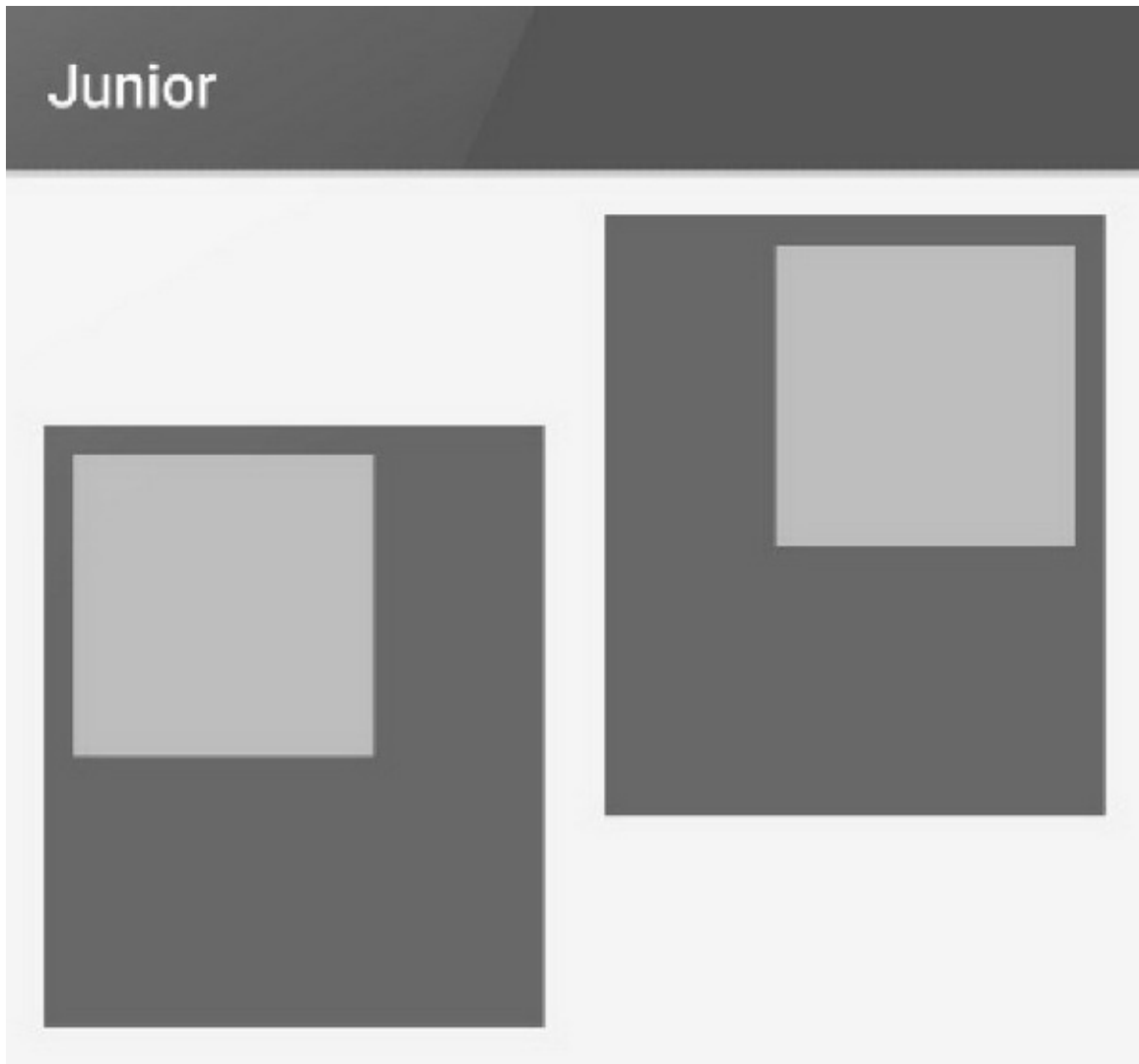
        <View
            android:layout_width="100dp"
            android:layout_height="100dp"
            android:background="#00fff" />
    </LinearLayout>

    <LinearLayout
        android:layout_width="0dp"
        android:layout_height="200dp"
        android:layout_weight="1"
        android:layout_gravity="top"
        android:gravity="right"
        android:background="#ff0000"
        android:layout_margin="10dp"
        android:padding="10dp"
        android:orientation="vertical">

        <View
            android:layout_width="100dp"
            android:layout_height="100dp"
            android:background="#00fff" />
    </LinearLayout>
</LinearLayout>

```

2-6layout_gravitygravitylayout_gravity
gravity



2-6layout_gravitygravity

2.2.3ScrollView AndroidScrollView

ScrollViewHorizontalScrollView3

- (1)layout_widthmatch_parentlayout_heightwrap_content
- (2)layout_widthwrap_contentlayout_heightmatch_parent
- (3)Caused byjava.lang.IllegalStateExceptionScrollView can host only one direct child

ScrollViewHorizontalScrollViewXML

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:orientation="vertical">
```

```
    <HorizontalScrollView  
        android:layout_width="wrap_content"  
        android:layout_height="200dp">
```

```
        <LinearLayout  
            android:layout_width="wrap_content"  
            android:layout_height="match_parent"  
            android:orientation="horizontal">
```

```
            <View
```

```

        android:layout_width="400dp"
        android:layout_height="match_parent"
        android:background="#aaffff" />

        <View
            android:layout_width="400dp"
            android:layout_height="match_parent"
            android:background="#ffff00" />
    </LinearLayout>
</HorizontalScrollView>

<ScrollView
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">

        <View
            android:layout_width="match_parent"
            android:layout_height="400dp"
            android:background="#00ff00" />

        <View
            android:layout_width="match_parent"
            android:layout_height="400dp"
            android:background="#fffaa" />
    </LinearLayout>
</ScrollView>
</LinearLayout>

```

ScrollView layout_height="match_parent" fillViewport
 android:layout_height="match_parent"
 android:fillViewport="true"

2.3

AndroidTextViewButtonImageViewImageButton

2.3.1TextView TextView2-4

2-4TextView

XML 中的属性	TextView 类的设置方法	说明
text:	setText	设置文本内容
textColor	setTextColor	设置文本颜色
textSize	setTextSize	设置文本大小
textAppearance	setTextAppearance	设置文本风格，风格定义在 res/styles.xml
gravity	setGravity	设置文本的对齐方式，对应的方法是 setGravity。取值说明见表 2-2

1App“Hello World”TextView

1.

“*****”

XML2-5

2-5

XML 中的属性	跑马灯用到的设置方法	说明
singleLine	setSingleLine	指定文本是否单行显示
ellipsize	setEllipsize	指定文本超出范围后的省略方式，省略方式的取值说明见表 2-6
focusable	setFocusable	指定是否获得焦点，跑马灯效果要求设置为 true
focusableInTouchMode	setFocusableInTouchMode	指定在触摸时是否获得焦点，跑马灯效果要求设置为 true

2-6

XML 中的省略方式	TruncateAt 类中省略方式	说明
start	START	省略号在开头
middle	MIDDLE	省略号在中间
end	END	省略号在末尾
marquee	MARQUEE	跑马灯显示

XML

```

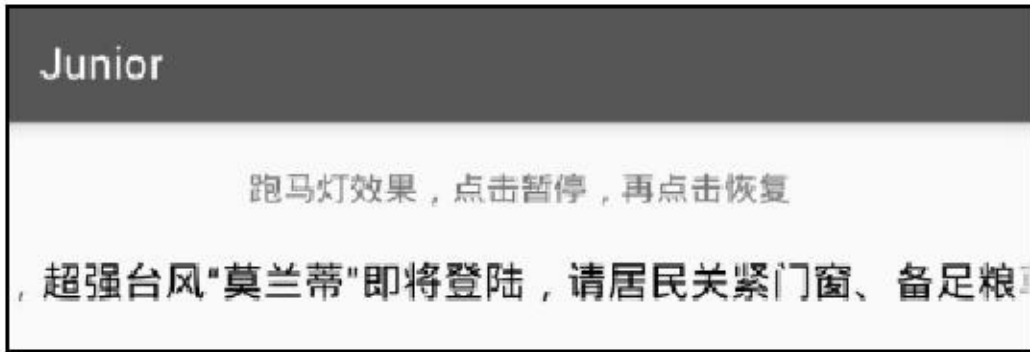
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
        android:gravity="center"
        android:text="跑马灯效果，点击暂停，再点击恢复" />

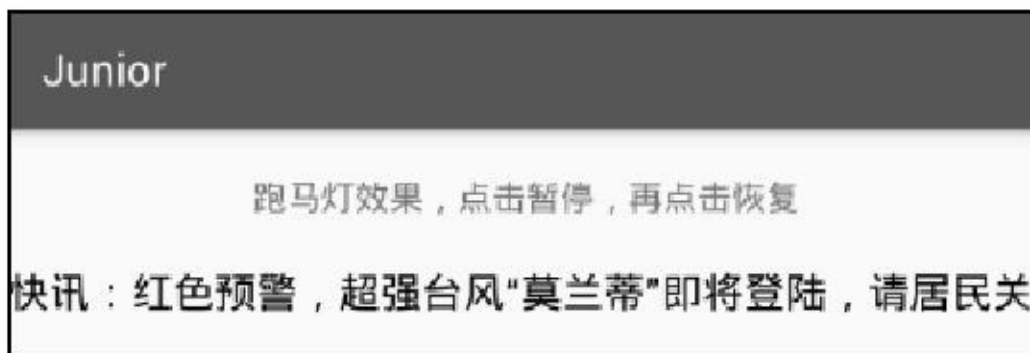
    <TextView
        android:id="@+id/tv_marquee"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
        android:singleLine="true"
        android:ellipsize="marquee"
        android:focusable="true"
        android:focusableInTouchMode="true"
        android:textColor="#000000"
        android:textSize="17sp"
        android:text="快讯：红色预警，超强台风“莫兰蒂”即将登陆，请居民关紧门窗、备足粮
草，做好防汛救灾准备！" />
</LinearLayout>

```

2-72-8



2-7



2-8

2.

XML2-7

12-7

XML 中的属性	聊天室用到的设置方法	说明
gravity	setGravity	指定文本的对齐方式，取值 left bottom，表示靠左对齐且靠下对齐
lines	setLines	指定文本的行数
maxLines	setMaxLines	指定文本的最大行数
scrollbars	无	指定滚动条的方向，取值 vertical，如果不指定将不显示滚动条
无	setMovementMethod	设置文本的移动方式，可设置 ScrollingMovementMethod，如果不设置将无法拉动文本

2-92-102-102-93

Junior

聊天室效果，点击添加聊天记录，长按删除聊天记录

13:10:11 你吃饭了吗？
13:10:14 我中奖啦！
13:10:16 我们去看电影吧
13:10:17 我中奖啦！
13:10:19 今天天气真好呀。
13:10:21 你吃饭了吗？
13:10:24 我们去看电影吧
13:10:27 晚上干什么好呢？

2-9

Junior

聊天室效果，点击添加聊天记录，长按删除聊天记录

13:10:17 我中奖啦！
13:10:19 今天天气真好呀。
13:10:21 你吃饭了吗？
13:10:24 我们去看电影吧
13:10:27 晚上干什么好呢？
13:11:28 晚上干什么好呢？
13:11:31 晚上干什么好呢？
13:11:34 我们去看电影吧

2-103

XML

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <TextView
        android:id="@+id/tv_control"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
        android:gravity="center"
        android:text="聊天室效果，点击添加聊天记录，长按删除聊天记录" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="200dp"
        android:orientation="vertical">

        <TextView
            android:id="@+id/tv_bbs"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_marginTop="20dp"
            android:scrollbars="vertical"
            android:textColor="#000000"
            android:textSize="17sp" />

    </LinearLayout>
</LinearLayout>
```

```

public class BbsActivity extends AppCompatActivity implements OnClickListener, OnLongClickListener {
    private TextView tv_bbs;
    private TextView tv_control;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_bbs);
        tv_control = (TextView) findViewById(R.id.tv_control);
        tv_control.setOnClickListener(this);
        tv_control.setOnLongClickListener(this);
        tv_bbs = (TextView) findViewById(R.id.tv_bbs);
        tv_bbs.setOnClickListener(this);
        tv_bbs.setOnLongClickListener(this);
        tv_bbs.setGravity(Gravity.LEFT|Gravity.BOTTOM);
        tv_bbs.setLines(8);
        tv_bbs.setMaxLines(8);
        tv_bbs.setMovementMethod(new ScrollingMovementMethod());
    }

    private String[] mChatStr = { "你吃饭了吗？", "今天天气真好呀。",
        "我中奖啦！ ", "我们去看电影吧", "晚上干什么好呢？ ", };

    @Override
    public void onClick(View v) {
        if (v.getId() == R.id.tv_control || v.getId() == R.id.tv_bbs) {
            int random = (int)(Math.random()*10) % 5;
            String newStr = String.format("%s\n%s %s",
                tv_bbs.getText().toString(), DateUtil.getNowTime(), mChatStr[random]);
            tv_bbs.setText(newStr);
        }
    }

    @Override
    public boolean onLongClick(View v) {
        if (v.getId() == R.id.tv_control || v.getId() == R.id.tv_bbs) {
            tv_bbs.setText("");
        }
        return true;
    }
}

```

2.3.2 Button TextView UIButton Button

background background@null Button Button TextView
TextView View Button

Android Button ImageButton

步骤

01

View.OnClickListener View.OnLongClickListener

Activity

步骤

02

onClick onLongClick

步骤

03

setOnClickListenersetOnLongClickListener

Button

```
public class ClickActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_click);
        Button btn_click = (Button) findViewById(R.id.btn_click);
        btn_click.setOnClickListener(new MyOnClickListener());
        btn_click.setOnLongClickListener(new MyOnLongClickListener());
    }

    class MyOnClickListener implements View.OnClickListener {
        @Override
        public void onClick(View v) {
            if (v.getId() == R.id.btn_click) {
                Toast.makeText(ClickActivity.this, "您点击了控件 : " + ((TextView) v).getText(),
                    Toast.LENGTH_SHORT).show();
            }
        }
    }

    class MyOnLongClickListener implements View.OnLongClickListener {
        @Override
        public boolean onLongClick(View v) {
            if (v.getId() == R.id.btn_click) {
                Toast.makeText(ClickActivity.this, "您长按了控件 : " + ((TextView) v).getText(),
                    Toast.LENGTH_SHORT).show();
            }
            return true;
        }
    }
}
```

2.3.3 ImageView ImageView

scaleTypefitCenter2-8

srcsrcscaleTypescaleTypefitXY

XML 中的拉伸类型	ScaleType 类中的拉伸类型	说明
fitXY	FIT_XY	拉伸图片使其正好填满视图（图片可能被拉伸变形）
fitStart	FIT_START	拉伸图片使其位于视图上部
fitCenter	FIT_CENTER	拉伸图片使其位于视图中间
fitEnd	FIT_END	拉伸图片使其位于视图下部
center	CENTER	保持图片原尺寸，并使其位于视图中间
centerCrop	CENTER_CROP	拉伸图片使其充满视图，并位于视图中间
centerInside	CENTER_INSIDE	使图片位于视图中间（只压不拉）。当图片尺寸大于视图时，centerInside 等同于 fitCenter；当图片尺寸小于视图时，centerInside 等同于 center

ImageView

setScaleType2-8

setImageDrawableDrawable

setImageResourceID

setImageBitmap

ImageViewcenter4fitCentercentercenterCrop

centerInsideImageView

```

public class ScaleActivity extends AppCompatActivity implements OnClickListener {
    private ImageView iv_scale;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_scale);
        iv_scale = (ImageView) findViewById(R.id.iv_scale);
        findViewById(R.id.btn_center).setOnClickListener(this);
        findViewById(R.id.btn_fitCenter).setOnClickListener(this);
        findViewById(R.id.btn_centerCrop).setOnClickListener(this);
        findViewById(R.id.btn_centerInside).setOnClickListener(this);
        findViewById(R.id.btn_fitXY).setOnClickListener(this);
        findViewById(R.id.btn_fitStart).setOnClickListener(this);
        findViewById(R.id.btn_fitEnd).setOnClickListener(this);
    }

    @Override
    public void onClick(View v) {
        if (v.getId() == R.id.btn_center) {
            iv_scale.setScaleType(ImageView.ScaleType.CENTER);
        } else if (v.getId() == R.id.btn_fitCenter) {
            iv_scale.setScaleType(ImageView.ScaleType.FIT_CENTER);
        } else if (v.getId() == R.id.btn_centerCrop) {
            iv_scale.setScaleType(ImageView.ScaleType.CENTER_CROP);
        } else if (v.getId() == R.id.btn_centerInside) {
            iv_scale.setScaleType(ImageView.ScaleType.CENTER_INSIDE);
        } else if (v.getId() == R.id.btn_fitXY) {
            iv_scale.setScaleType(ImageView.ScaleType.FIT_XY);
        } else if (v.getId() == R.id.btn_fitStart) {
            iv_scale.setScaleType(ImageView.ScaleType.FIT_START);
        } else if (v.getId() == R.id.btn_fitEnd) {
            iv_scale.setScaleType(ImageView.ScaleType.FIT_END);
        }
    }
}

```

fitCenter2-11
center2-12
centerCrop2-13
centerInside2-14

Junior



2-11fitCenter

Junior



2-12center

Junior



2-13centerCrop

Junior



2-14centerInside

AndroidImageViewImageView

(View)

setDrawingCacheEnabledtruefalse

isDrawingCacheEnabled

setDrawingCacheQuality

getDrawingCacheBitmap

setDrawingCacheBackgroundColor()

步驟

01

setDrawingCacheEnabled

步驟

02

getDrawingCache

步骤

03

setDrawingCacheEnabled

setDrawingCacheEnabled

```
public class CaptureActivity extends AppCompatActivity implements OnClickListener, OnLongClickListener {
    private TextView tv_capture;
    private ImageView iv_capture;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_capture);
        tv_capture = (TextView) findViewById(R.id.tv_capture);
        iv_capture = (ImageView) findViewById(R.id.iv_capture);
        tv_capture.setDrawingCacheEnabled(true);
        findViewById(R.id.btn_chat).setOnClickListener(this);
        findViewById(R.id.btn_chat).setOnLongClickListener(this);
        findViewById(R.id.btn_capture).setOnClickListener(this);
    }

    private String[] mChatStr = { "你吃饭了吗？", "今天天气真好呀。",
        "我中奖啦！", "我们去看电影吧。", "晚上干什么好呢？" };

    @Override
    public boolean onLongClick(View v) {
        if (v.getId() == R.id.btn_chat) {
            tv_capture.setText("");
        }
        return true;
    }

    @Override
    public void onClick(View v) {
        if (v.getId() == R.id.btn_chat) {
            int random = (int)(Math.random()*10) % 5;
            String newStr = String.format("%s\n%s %s",
                tv_capture.getText().toString(), DateUtil.getNowTime(), mChatStr[random]);
            tv_capture.setText(newStr);
        } else if (v.getId() == R.id.btn_capture) {
```

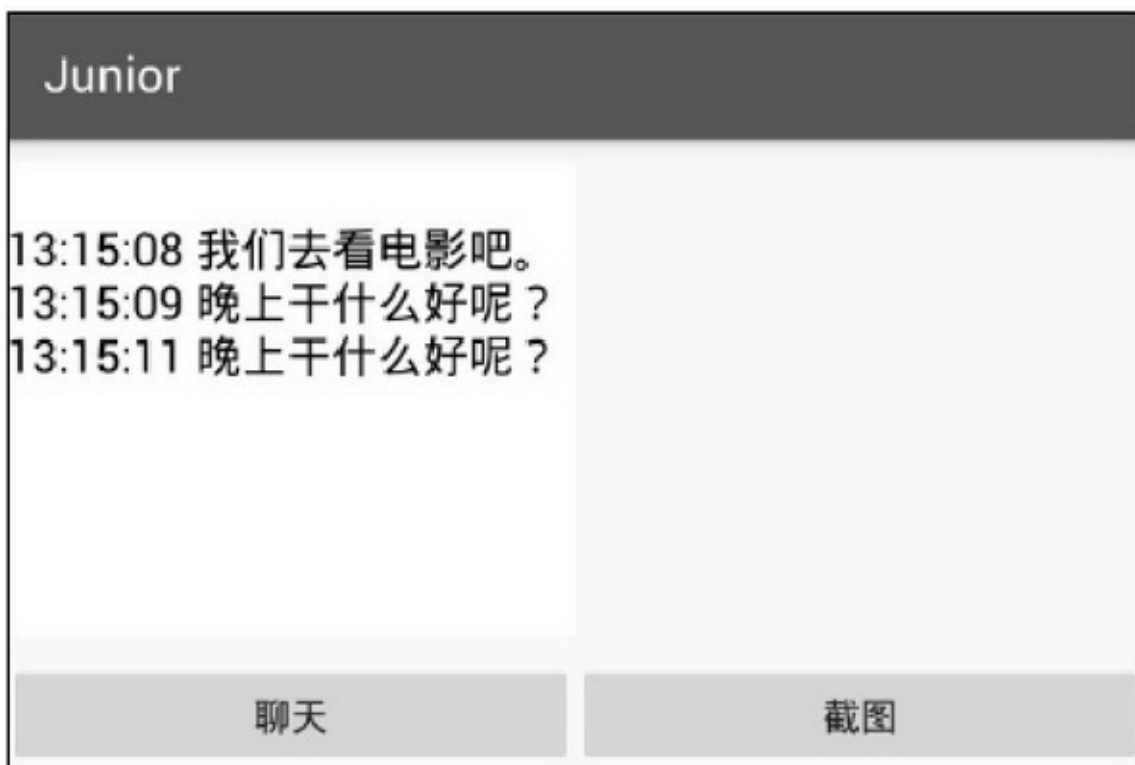
```

        Bitmap bitmap = tv_capture.getDrawingCache();
        iv_capture.setImageBitmap(bitmap);
        // 注意截图完毕后不能马上关闭绘图缓存，因为界面渲染需要时间
        // 如果立即关闭缓存，渲染界面就会找不到位图对象，从而报错
        // java.lang.IllegalArgumentException: Cannot draw recycled bitmaps
        mHandler.postDelayed(mResetCache, 200);
    }
}

private Handler mHandler = new Handler();
private Runnable mResetCache = new Runnable() {
    @Override
    public void run() {
        tv_capture.setDrawingCacheEnabled(false);
        tv_capture.setDrawingCacheEnabled(true);
    }
};
}

```

2-152-162-152-16





2-16

2.3.4 ImageButton ImageButtonImageViewButton, ImageView

ImageButtonImageButton

ImageButtonButtonImageButton

(1)Button()ImageButton

(2)ImageButtonButton()

(3)ButtonImageButton

ButtonImageButtonButtonImageButton

ButtonImageButtonButtonImageButton

LinearLayoutImageViewTextView

TextViewTextViewButtonXML5

drawableTop

drawableBottom

drawableLeft

drawableRight

drawablePadding

setCompoundDrawables

setCompoundDrawablePadding

```
public class IconActivity extends AppCompatActivity implements OnClickListener {
    private Button btn_icon;
    private Drawable drawable;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_icon);
        btn_icon = (Button) findViewById(R.id.btn_icon);
        drawable = getResources().getDrawable(R.mipmap.ic_launcher);
        // 必须设置图片大小, 否则不显示图片
        drawable.setBounds(0, 0, drawable.getMinimumWidth(), drawable.getMinimumHeight());
        findViewById(R.id.btn_left).setOnClickListener(this);
        findViewById(R.id.btn_top).setOnClickListener(this);
        findViewById(R.id.btn_right).setOnClickListener(this);
        findViewById(R.id.btn_bottom).setOnClickListener(this);
    }

    @Override
    public void onClick(View v) {
        if (v.getId() == R.id.btn_left) {
            btn_icon.setCompoundDrawables(drawable, null, null, null);
        } else if (v.getId() == R.id.btn_top) {
            btn_icon.setCompoundDrawables(null, drawable, null, null);
        } else if (v.getId() == R.id.btn_right) {
            btn_icon.setCompoundDrawables(null, null, drawable, null);
        } else if (v.getId() == R.id.btn_bottom) {
            btn_icon.setCompoundDrawables(null, null, null, drawable);
        }
    }
}
```

2-17()2-18()2-19()2-20()



2-17



2-18



2-19



2-20

2.4

AndroidStateListDrawableShapeDrawable()

2.4.1 Drawable AndroidDrawable()

drawableresdrawable\res\drawableXMLdrawable

drawable-ldpi(240×320)

drawable-mdpi(320×480)

drawable-hdpi(480×800)4~4.5()

drawable-xhdpi(720×1280)5~5.5

drawable-xxhdpi(1080×1920)6~6.5

drawable-xxxhdpi(1440×2560)7

AndroidAppdrawable-hdpi.png(

480×800)480×800App720×1280AppAndroidbg.pngbg.png

720×1280

drawabledrawableAndroid Postfix Completion1“1.5.3”

XMLdrawable“@drawable/****”backgroundImageView

ImageButtonsrcTextViewButtondrawableTopdrawable

drawable

(1)setBackgroundResourcessetImageResourcedrawableID

“R.drawable.****”

(2)setBackgroundDrawablessetImageDrawablessetCompoundDrawables

DrawableDrawable

Drawable drawable = getResources().getDrawable(R.drawable.apple);

2.4.2

drawableButtonDrawableStateListDrawableXML

drawable

```
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:state_pressed="true" android:drawable="@drawable/button_pressed" />
    <item android:drawable="@drawable/button_normal" />
</selector>
```

XMLstate_pressedtruebutton_selectedbutton_normal
Buttonbackgrounddrawable2-21()2-22()



2-21



2-22

StateListDrawableButtonStateListDrawable2-9

2-9

状态类型	说明	常用的控件
state_pressed	是否按下	按钮 Button
state_checked	是否勾选	单选框 RadioButton、复选框 CheckBox
state_focused	是否获取焦点	文本编辑框 EditText
state_selected	是否选中	各控件均可

2.4.3

XMLXML—— shapeApp
 shape6corners()gradient()padding()size()
 solid()stroke()

1. shape

shapeXMLshape
 shape2-10

2-10

形状类型	说明
rectangle	矩形。默认值
oval	椭圆。此时 corners 节点会失效
line	直线。此时必须设置 stroke 节点，否则会报错
ring	圆环

2. corners

cornersshape4cornerscorners
 bottomLeftRadius
 bottomRightRadius
 topLeftRadius
 topRightRadius
 radius(4radius)

3. gradien

gradienshapegradiengradien

angle0990618032700/12

type2-11

2-11

渐变类型	说明
linear	线性渐变, 默认值
radial	放射渐变, 起始颜色就是圆心颜色
sweep	滚动渐变, 即一个线段以某个端点为圆心做 360 度旋转

centerXXandroid:type="linear"

centerYYandroid:type="linear"

gradientRadiusandroid:type="radial"

centerColor

startColor

endColor

useLeveltruefalse

4. padding

paddingshapepaddingpadding

bottom

left

right

top

5. size

sizeshape()size size

height

width

6. solid

solidshapesolidsolid

color

7. stroke

strokestrokestroke

color

dashGap

dashWidth

widthdashGapdashWidth0

3corners()solid()stroke()shape()shapeXML

```
<shape xmlns:android="http://schemas.android.com/apk/res/android" >
  <solid android:color="#ffdd66" />
  <stroke
    android:width="1dp"
    android:color="#ffa6a6" />
  <corners
    android:bottomLeftRadius="10dp"
    android:bottomRightRadius="10dp"
    android:topLeftRadius="10dp"
    android:topRightRadius="10dp" />
</shape>
```



2-23shape

3

(1)TextView

(2)ImageView

(3)shapeshapeline

View1dpXML

```
<View  
    android:layout_width="match_parent"  
    android:layout_height="1dp"  
    android:background="#000000" />
```

2.4.4

ImageView2-24

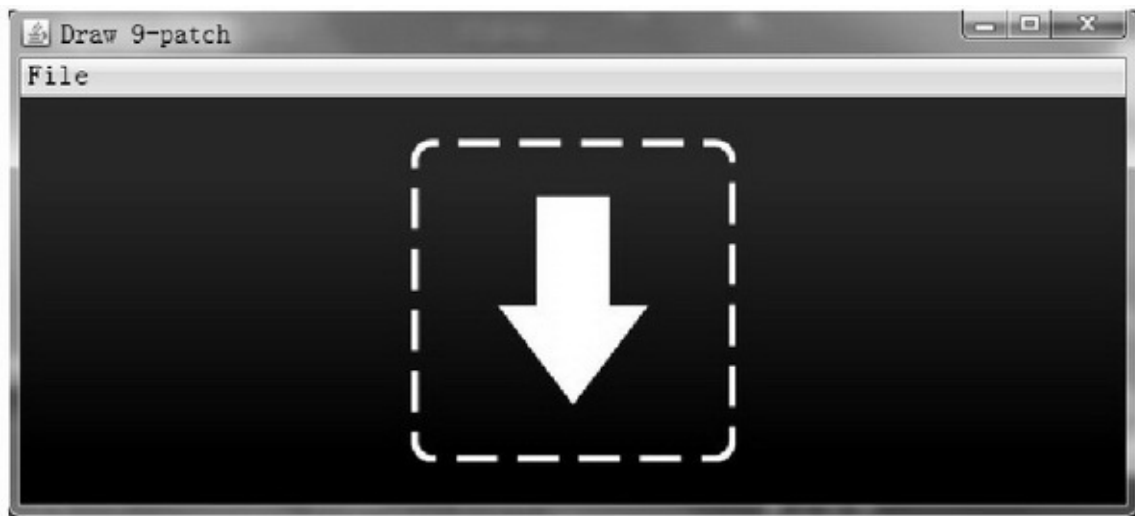
Androidpng“.9”3×3shapestrokewidth(

1dp)shape1dpshape

AndroidSDKSDKsdk\tools\draw9patch.bat2-25

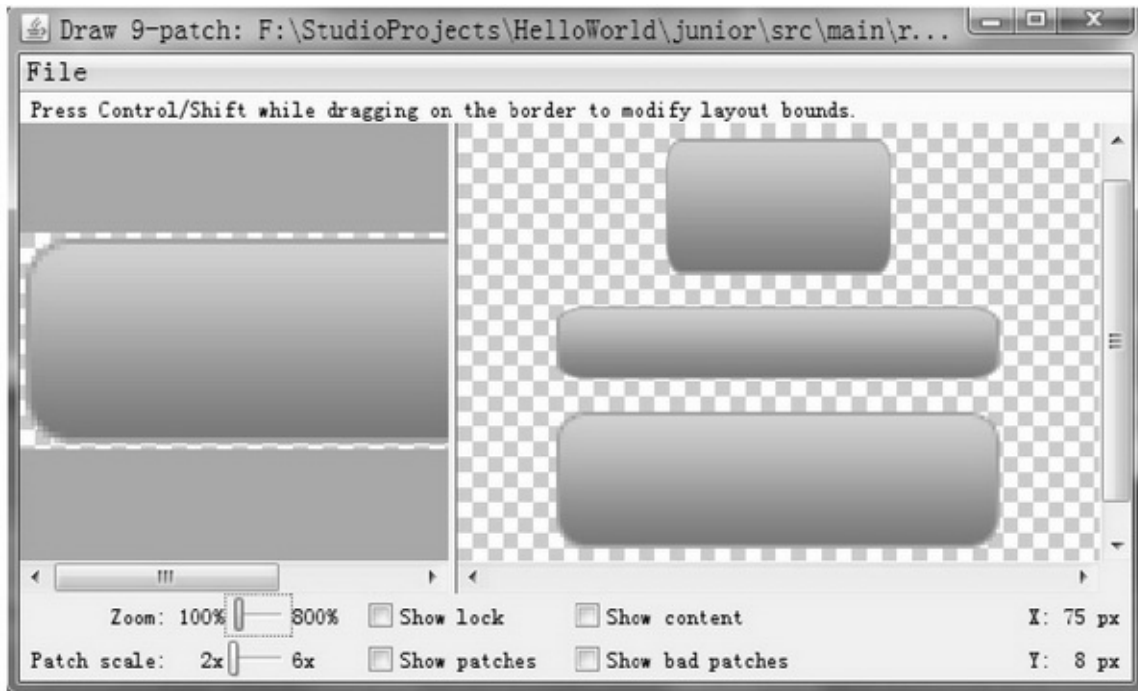


2-24



2-25

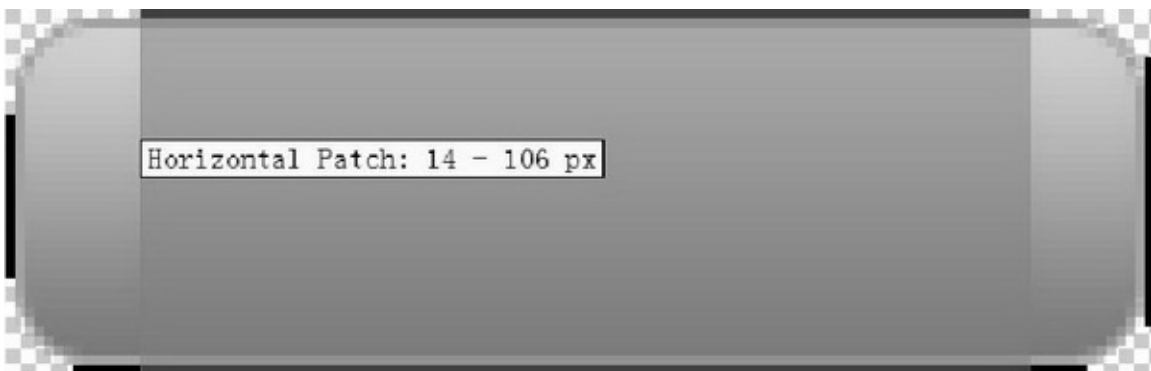
PNG2-26



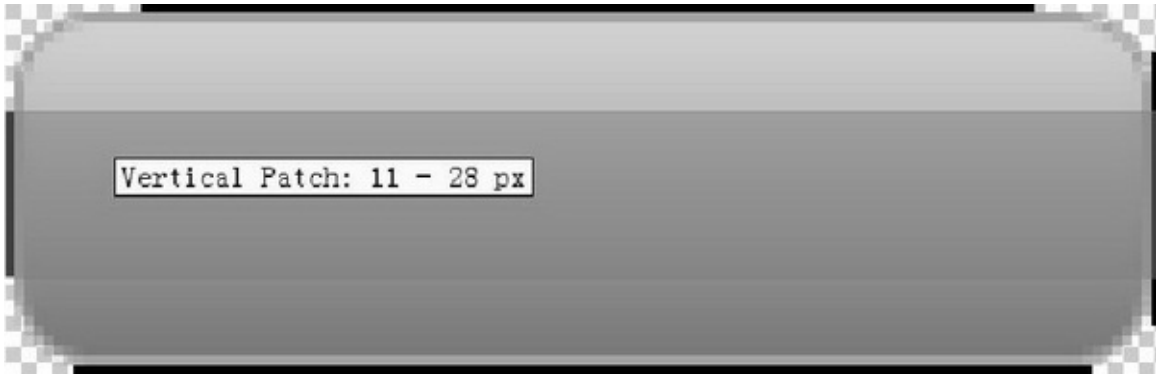
2-26

2-27

2-28



2-27



2-28

2-29 Horizontal Padding `android:paddingLeft`
`android:paddingRight`

2-30 Vertical Padding `android:paddingTop`
`android:paddingBottom`



2-29



2-30

XMLdraw9patch.batpadding

2.5

AndroidApp

2.5.1

Windwos2-31

()App2-32





2-32

LinearLayoutLinearLayout4LinearLayout

ScrollViewScrollView

TextView""TextViewTextViewTextView

ButtonButton

ImageView

ImageButton"√"ImageButton

shapeshape
“0”

2.5.2 Log/Toast Android Cprintf Java System.out.println App

1. Log

LogAndroid Studio logcat Log
Log.e
Log.w
Log.i
Log.d
Log.v

2. Toast

Toast “0” “0”

Toast

Toast.makeText(MainActivity.this, "",
Toast.LENGTH_SHORT).show(); valuesstyles.xml
style="@style/btn_cal" styles.xml

```
<style name="btn_cal">  
    <item name="android:layout_width">0dp</item>  
    <item name="android:layout_height">match_parent</item>  
    <item name="android:layout_weight">1</item>  
    <item name="android:gravity">center</item>  
    <item name="android:textColor">@color/black</item>  
    <item name="android:textSize">30sp</item>  
    <item name="android:background">@drawable/btn_nine_selector</item>  
</style>
```

2.5.3



2-33

AppApp2-33

3

步骤 01

CalculatorActivity.java activity_calculator.xml

AndroidManifest.xml
activity AppCompatActivity
NotFoundException
application

步骤 02

<activity android:name=".CalculatorActivity" />

res/layout/activity_calculator.xml

步骤 03

CalculatorActivityMainActivity.onCreate
setContentView(R.layout.activity_calculatoractivity_calculator.xml)
App

```
public void onClick(View v) {
    int resid = v.getId();
    String inputText;
    if (resid == R.id.ib_sqrt) {
        inputText = "√";
    } else {
        inputText = ((TextView) v).getText().toString();
    }
    Log.d(TAG, "resid="+resid+",inputText="+inputText);
    if (resid == R.id.btn_clear) {
        clear("");
    } else if (resid == R.id.btn_cancel) {
        if (operator.equals("") == true) {
            if (firstNum.length() == 1) {
                firstNum = "0";
            } else if (firstNum.length() > 0) {
                firstNum = firstNum.substring(0, firstNum.length() - 1);
            } else {
                Toast.makeText(this, "没有可取消的数字了", Toast.LENGTH_SHORT).show();
            }
        }
    }
}
```



```

        return;
    }
    showText = firstNum;
    tv_result.setText(showText);
} else {
    if (nextNum.length() == 1) {
        nextNum = "";
    } else if (nextNum.length() > 0) {
        nextNum = nextNum.substring(0, nextNum.length() - 1);
    } else {
        Toast.makeText(this, "没有可取消的数字了", Toast.LENGTH_SHORT).show();
        return;
    }
    showText = showText.substring(0, showText.length() - 1);
    tv_result.setText(showText);
}
} else if (resid == R.id.btn_equal) {
    if (operator.length() == 0 || operator.equals("=") == true) {
        Toast.makeText(this, "请输入运算符", Toast.LENGTH_SHORT).show();
        return;
    } else if (nextNum.length() <= 0) {
        Toast.makeText(this, "请输入数字", Toast.LENGTH_SHORT).show();
        return;
    }
    if (caculate() == true) {
        operator = inputText;
        showText = showText + "=" + result;
        tv_result.setText(showText);
    } else {
        return;
    }
} else if (resid == R.id.btn_plus || resid == R.id.btn_minus
    || resid == R.id.btn_multiply || resid == R.id.btn_divide) {
    if (firstNum.length() <= 0) {
        Toast.makeText(this, "请输入数字", Toast.LENGTH_SHORT).show();
        return;
    }
    if (operator.length() == 0 || operator.equals("=") == true || operator.equals("√") == true) {
        operator = inputText; // 操作符
        showText = showText + operator;
        tv_result.setText(showText);
    } else {
        Toast.makeText(this, "请输入数字", Toast.LENGTH_SHORT).show();
    }
}

```

```

        return;
    }
} else if (resid == R.id.ib_sqrt) {
    if (firstNum.length() <= 0) {
        Toast.makeText(this, "请输入数字", Toast.LENGTH_SHORT).show();
        return;
    } else if (Double.parseDouble(firstNum) < 0) {
        Toast.makeText(this, "开根号的数值不能小于 0", Toast.LENGTH_SHORT).show();
        return;
    }
    result = String.valueOf(Math.sqrt(Double.parseDouble(firstNum)));
    firstNum = result;
    nextNum = "";
    operator = inputText;
    showText = showText + "√=" + result;
    tv_result.setText(showText);
    Log.d(TAG, "result="+result+",firstNum="+firstNum+",operator="+operator);
} else {
    if (operator.equals("=") == true) {
        operator = "";
        firstNum = "";
        showText = "";
    }
    if (resid == R.id.btn_dot) {
        inputText = ".";
    }
    if (operator.equals("") == true) {
        firstNum = firstNum + inputText;
    } else {
        nextNum = nextNum + inputText;
    }
    showText = showText + inputText;
    tv_result.setText(showText);
}
}

private String operator = ""; // 操作符
private String firstNum = ""; // 前一个操作数
private String nextNum = ""; // 后一个操作数
private String result = ""; // 当前计算结果
private String showText = ""; // 显示的文本内容
private boolean caculate() { // 开始加减乘除四则运算
    if (operator.equals("+") == true) {
        result = String.valueOf(Arith.add(firstNum, nextNum));
    }
}

```

```
    } else if (operator.equals("-") == true) {
        result = String.valueOf(Arith.sub(firstNum, nextNum));
    } else if (operator.equals("×") == true) {
        result = String.valueOf(Arith.mul(firstNum, nextNum));
    } else if (operator.equals("÷") == true) {
        if ("0".equals(nextNum)) {
            Toast.makeText(this, "被除数不能为零", Toast.LENGTH_SHORT).show();
            return false;
        } else {
            result = String.valueOf(Arith.div(firstNum, nextNum));
        }
    }
    firstNum = result;
    nextNum = "";
    return true;
}

private void clear(String text) { // 清空并初始化
    showText = text;
    tv_result.setText(showText);
    operator = "";
    firstNum = "";
    nextNum = "";
    result = "";
}
```

2.6

App0000

“”AppLogToastApp

3

(1)

(2)

(3)

第3章 中级控件



AppActivity“App”

3.1

AndroidRelativeLayoutFrameLayout

3.1.1RelativeLayout

RelativeLayoutRelativeLayout(RelativeLayout)

3-1

3-1

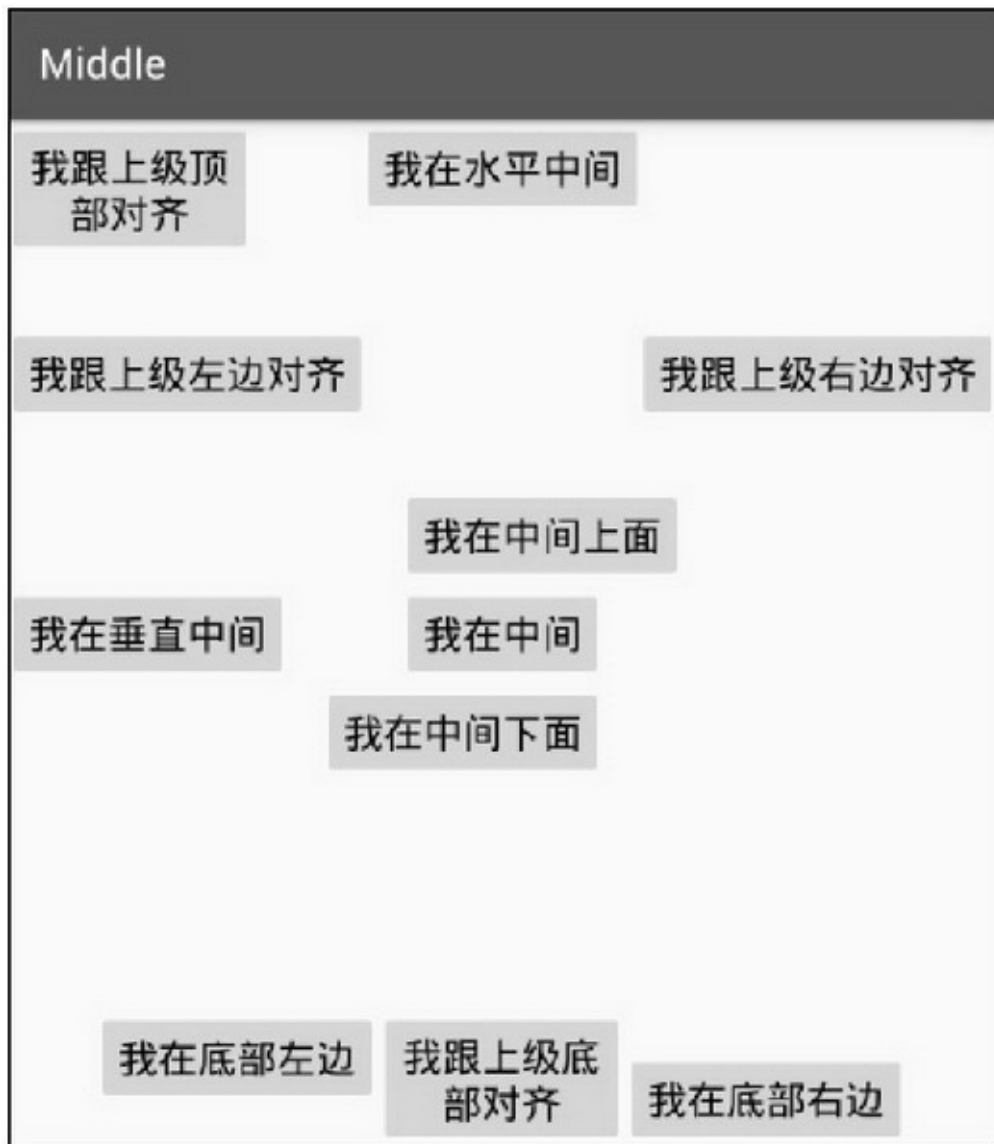
XML 中的相对位置属性	RelativeLayout 类的相对位置	相对位置说明
layout_toLeftOf	LEFT_OF	当前视图在指定视图的左边
layout_toRightOf	RIGHT_OF	当前视图在指定视图的右边
layout_above	ABOVE	当前视图在指定视图的上方
layout_below	BELOW	当前视图在指定视图的下方
layout_alignLeft	ALIGN_LEFT	当前视图与指定视图的左侧对齐
layout_alignRight	ALIGN_RIGHT	当前视图与指定视图的右侧对齐
layout_alignTop	ALIGN_TOP	当前视图与指定视图的顶部对齐
layout_alignBottom	ALIGN_BOTTOM	当前视图与指定视图的底部对齐
layout_centerInParent	CENTER_IN_PARENT	当前视图在上级视图中间
layout_centerHorizontal	CENTER_HORIZONTAL	当前视图在上级视图的水平方向居中
layout_centerVertical	CENTER_VERTICAL	当前视图在上级视图的垂直方向居中
layout_alignParentLeft	ALIGN_PARENT_LEFT	当前视图与上级视图的左侧对齐
layout_alignParentRight	ALIGN_PARENT_RIGHT	当前视图与上级视图的右侧对齐
layout_alignParentTop	ALIGN_PARENT_TOP	当前视图与上级视图的顶部对齐
layout_alignParentBottom	ALIGN_PARENT_BOTTOM	当前视图与上级视图的底部对齐

RelativeLayoutXML

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="500dp" >
    <Button
        android:id="@+id/btn_center"
        style="@style/btn_relative"
```

```
        android:layout_centerInParent="true"
        android:text="我在中间" />
<Button
    android:id="@+id/btn_center_horizontal"
    style="@style/btn_relative"
    android:layout_centerHorizontal="true"
    android:text="我在水平中间" />
<Button
    android:id="@+id/btn_center_vertical"
    style="@style/btn_relative"
    android:layout_centerVertical="true"
    android:text="我在垂直中间" />
<Button
    android:id="@+id/btn_parent_left"
    style="@style/btn_relative"
    android:layout_marginTop="100dp"
    android:layout_alignParentLeft="true"
    android:text="我跟上级左边对齐" />
<Button
    android:id="@+id/btn_parent_top"
    style="@style/btn_relative"
    android:layout_width="120dp"
    android:layout_alignParentTop="true"
    android:text="我跟上级顶部对齐" />
<Button
    android:id="@+id/btn_parent_right"
    style="@style/btn_relative"
    android:layout_marginTop="100dp"
    android:layout_alignParentRight="true"
    android:text="我跟上级右边对齐" />
<Button
    android:id="@+id/btn_parent_bottom"
    style="@style/btn_relative"
    android:layout_width="120dp"
    android:layout_alignParentBottom="true"
    android:layout_centerHorizontal="true"
    android:text="我跟上级底部对齐" />
<Button
    android:id="@+id/btn_left_bottom"
    style="@style/btn_relative"
    android:layout_toLeftOf="@+id/btn_parent_bottom"
    android:layout_alignTop="@+id/btn_parent_bottom"
    android:text="我在底部左边" />
```

```
<Button
    android:id="@+id/btn_right_bottom"
    style="@style/btn_relative"
    android:layout_toRightOf="@+id/btn_parent_bottom"
    android:layout_alignBottom="@+id/btn_parent_bottom"
    android:text="我在底部右边" />
<Button
    android:id="@+id/btn_above_center"
    style="@style/btn_relative"
    android:layout_above="@+id/btn_center"
    android:layout_alignLeft="@+id/btn_center"
    android:text="我在中间上面" />
<Button
    android:id="@+id/btn_below_center"
    style="@style/btn_relative"
    android:layout_below="@+id/btn_center"
    android:layout_alignRight="@+id/btn_center"
    android:text="我在中间下面" />
</RelativeLayout>
```

3-1

3-1RelativeLayoutXML

RelativeLayout.LayoutParamsaddRule3-1ID

RelativeLayout

```

public void onClick(View v) {
    if (v.getId() == R.id.btn_add_left) {
        addNewView(RelativeLayout.LEFT_OF, RelativeLayout.ALIGN_TOP, v.getId());
    } else if (v.getId() == R.id.btn_add_above) {
        addNewView(RelativeLayout.ABOVE, RelativeLayout.ALIGN_LEFT, v.getId());
    } else if (v.getId() == R.id.btn_add_right) {
        addNewView(RelativeLayout.RIGHT_OF, RelativeLayout.ALIGN_BOTTOM, v.getId());
    } else if (v.getId() == R.id.btn_add_below) {
        addNewView(RelativeLayout.BELOW, RelativeLayout.ALIGN_RIGHT, v.getId());
    } else if (v.getId() == R.id.btn_add_center) {
        addNewView(RelativeLayout.CENTER_IN_PARENT, -1, rl_content.getId());
    } else if (v.getId() == R.id.btn_add_parent_left) {
        addNewView(RelativeLayout.ALIGN_PARENT_LEFT, RelativeLayout.CENTER_
VERTICAL, rl_content.getId());
    } else if (v.getId() == R.id.btn_add_parent_top) {
        addNewView(RelativeLayout.ALIGN_PARENT_TOP, RelativeLayout.CENTER_
HORIZONTAL, rl_content.getId());
    } else if (v.getId() == R.id.btn_add_parent_right) {
        addNewView(RelativeLayout.ALIGN_PARENT_RIGHT, -1, rl_content.getId());
    } else if (v.getId() == R.id.btn_add_parent_bottom) {
        addNewView(RelativeLayout.ALIGN_PARENT_BOTTOM, -1, rl_content.getId());
    }
}

private void addNewView(int firstAlign, int secondAlign, int referId) {
    View v = new View(this);
    v.setBackgroundColor(0xaa66ff66);
    RelativeLayout.LayoutParams rl_params = new RelativeLayout.LayoutParams(100, 100);
    rl_params.addRule(firstAlign, referId);
    if (secondAlign >= 0) {
        rl_params.addRule(secondAlign, referId);
    }
    v.setLayoutParams(rl_params);
    v.setOnLongClickListener(new OnLongClickListener() {
        @Override
        public boolean onLongClick(View vv) {
            rl_content.removeView(vv);
            return true;
        }
    });
    rl_content.addView(v);
}

```



3-2

3.1.2FrameLayout

FrameLayoutFrameLayout

foreground

foregroundGravitygravity

FrameLayout

3-33-4

Middle

给下方的帧布局添加视图





3-4

AndroidAbsoluteLayoutTableLayout3

3.2

CheckBoxSwitchiOSRadioButtonRadioGroup

3.2.1CheckBox CompoundButtonAndroid

CompoundButtonCompoundButtonCheckBox

CompoundButton
RadioButtonSwitchCompoundButton

checkedtruefalse

button

CompoundButton4

setChecked

setButtonDrawable

setOnCheckedChangeListener

isChecked

CheckBoxCompoundButtonCheckBox

setOnCheckedChangeListener

CompoundButton.OnCheckedChangeListener

```
private class CheckListener implements CompoundButton.OnCheckedChangeListener {
    @Override
    public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
        String desc = String.format("您勾选了控件%d, 状态为%b", buttonView.getId(),
isChecked);
        Toast.makeText(MainActivity.this, desc, Toast.LENGTH_LONG).show();
    }
}
```

buttoncheck_choosecheck_unchoose

```
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:state_checked="true" android:drawable="@drawable/check_choose"/>
    <item android:drawable="@drawable/check_unchoose"/>
</selector>
```

3.2.2 Switch SwitchAndroid4.1.2 SwitchCheckBox CheckBoxSwitch3-2

3-2Switch

XML 中的属性	Switch 类的设置方法	说明
textOn	setTextOn	设置右侧开启时的文本
textOff	setTextOff	设置左侧关闭时的文本
switchPadding	setSwitchPadding	设置左右两个开关按钮之间的距离
thumbTextPadding	setThumbTextPadding	设置文本左右两边的距离。如果设置了该属性，switchPadding 属性就会失效
thumb	setThumbDrawable setThumbResource	设置开关轨道的背景
track	setTrackDrawable setTrackResource	设置开关标识的图标

SwitchCheckBoxSwitch3-53-6iPhoneiOSUI
AppiOSiOSUISwitch



3-5Switch“”



3-6Switch“”

AndroidiOSStateListDrawableXML

```
<selector xmlns:android="http://schemas.android.com/apk/res/android">  
  <item android:state_checked="true" android:drawable="@drawable/switch_on"/>  
  <item android:drawable="@drawable/switch_off"/>  
</selector>
```

CheckBoxbackgroundbutton@nullbackgroundbuttonbutton
background
iOS3-73-8



3-7iOS“”



3-8iOS“”

3.2.3 RadioButton RadioGroup RadioGroup

RadioButton RadioGroup RadioGroup orientation horizontal

RadioGroup3
vertical RadioGroup RadioButton (TextView
setCheckedId
ImageView) RadioGroup
getCheckedRadioButtonId

setCheckedChangeListener

RadioButton RadioButton RadioGroup

RadioGroup.OnCheckedChangeListener RadioGroup

setCheckedChangeListener

RadioGroup

```

class RadioListener implements RadioGroup.OnCheckedChangeListener {
    @Override
    public void onCheckedChanged(RadioGroup group, int checkedId) {
        Toast.makeText(MainActivity.this, "您选中了控件"+checkedId,
Toast.LENGTH_LONG).show();
    }
}

```

RadioButton button3-9 drawableLeft(button@null)
drawablePadding 3-10

Middle

请选择您的婚姻状况

未婚

已婚

哇哦，你的前途不可限量

3-9button

Middle

请选择您的婚姻状况

未婚

已婚

哇哦，祝你早生贵子

3-10drawableLeft

3CheckBoxbuttoniOSbackgroundRadioButtondrawableLeft3

button

background

drawableLeft



3-11dialog

3.3.2 ArrayAdapter Spinner.setAdapter

Spinner.setAdapter

`ArrayAdapter(R.layout.item_select, TextView`

```
ArrayAdapter<String> starAdapter = new ArrayAdapter<String>(this,  
R.layout.item_select, starArray);
```

`R.layout.item_dropdown`

```
starAdapter.setDropDownViewResource(R.layout.item_dropdown);  
ArrayAdapterSpinnersetAdapter
```

3.3.3 SimpleAdapter ArrayAdapter SimpleAdapter

```
SimpleAdapterArrayAdapter  
SimpleAdapterMap345  
SimpleAdapter
```

```
int[] iconArray = {R.drawable.shuixing, R.drawable.jinxing, R.drawable.diqiu,  
    R.drawable.huoxing, R.drawable.muxing, R.drawable.tuxing };  
List<Map<String, Object>> list = new ArrayList<Map<String, Object>>();  
for (int i = 0; i < iconArray.length; i++) {  
    Map<String, Object> item = new HashMap<String, Object>();  
    item.put("icon", iconArray[i]);  
    item.put("name", starArray[i]);  
    list.add(item);  
}  
SimpleAdapter starAdapter = new SimpleAdapter(this, list, R.layout.item_select,  
    new String[] { "icon", "name" }, new int[] { R.id.iv_icon, R.id.tv_name });  
starAdapter.setDropDownViewResource(R.layout.item_simple);
```

0

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal" >

    <ImageView
        android:id="@+id/iv_icon"
        android:layout_width="0dp"
        android:layout_height="50dp"
        android:layout_weight="1"
        android:gravity="center" />

    <TextView
        android:id="@+id/tv_name"
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="3"
        android:gravity="center"
        android:textSize="17sp"
        android:textColor="#ff0000" />

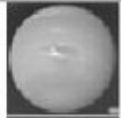
</LinearLayout>
```

3-12

Middle

水星

请选择行星



水星



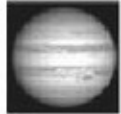
金星



地球



火星



木星



土星

3.4

AndroidEditTextAutoCompleteTextViewEditText4

3.4.1 EditText EditText EditText

inputTypesetInputType3-3“|”

maxLength

hintsetHint

textColorHintsetHintTextColor

3-3

输入类型	说明
text	文本
textPassword	文本密码。显示时用星号“*”代替
number	整型数
numberSigned	带符号的数字。允许在开头带负号“-”
numberDecimal	带小数点的数字
numberPassword	数字密码。显示时用星号“*”代替
datetime	时间日期格式。除了数字外，还允许输入横线、斜杆、空格、冒号
date	日期格式。除了数字外，还允许输入横线“-”和斜杆“/”
time	时间格式。除了数字外，还允许输入冒号“:”

4

1.

EditText

cursorVisiblesetCursorVisible

textCursorDrawable

cursorVisiblefalsesetTextCursorDrawable3-13

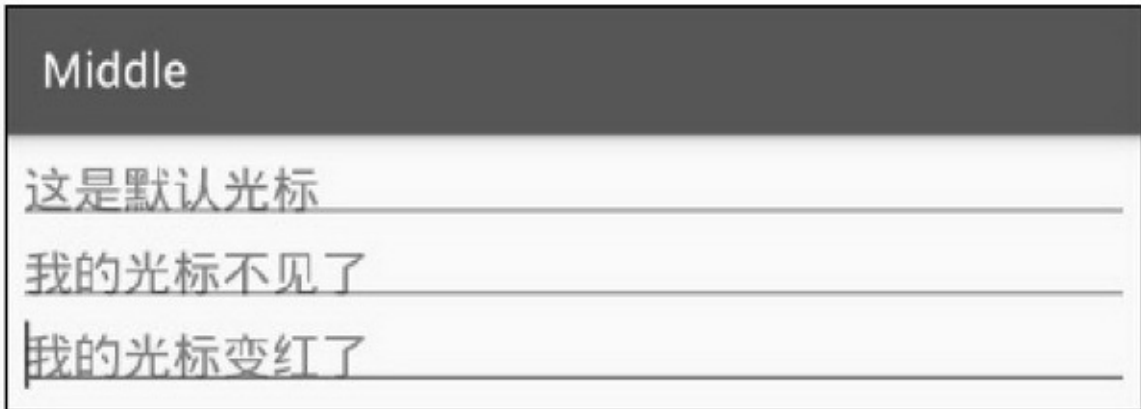
2.

EditTextbackgroundbackground@nullbackground

XML()shape_edit_focusshape_edit_normal

```
<selector xmlns:android="http://schemas.android.com/apk/res/android">  
  <item android:state_focused="true" android:drawable="@drawable/shape_edit_focus"/>  
  <item android:drawable="@drawable/shape_edit_normal"/>  
</selector>
```

3-14



3-13EditText



3-14EditText

3.

EditTextfocusablefocusableInTouchMode

true

“”“”“”11App

3

(1)

maxLengthEditText

```
public static int getMaxLength(EditText et) {
    int length = 0;
    try {
        InputFilter[] inputFilters = et.getFilters();
        for (InputFilter filter : inputFilters) {
            Class<?> c = filter.getClass();
            if (c.getName().equals("android.text.InputFilter$LengthFilter")) {
                Field[] f = c.getDeclaredFields();
                for (Field field : f) {
                    if (field.getName().equals("mMax")) {
                        field.setAccessible(true);
                        length = (Integer) field.get(filter);
                    }
                }
            }
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
    return length;
}
```

(2)

TextWatcher3

beforeTextChanged

onTextChanged

afterTextChanged

afterTextChangedTextWatcherEditTextaddTextChangedListener

```

private class HideTextWatcher implements TextWatcher {
    private EditText mView;
    private int mMaxLength;
    private CharSequence mStr;

    public HideTextWatcher(EditText v) {
        super();
        mView = v;
        mMaxLength = ViewUtil.getMaxLength(v);
    }

    @Override
    public void beforeTextChanged(CharSequence s, int start, int count, int after) {
    }

    @Override
    public void onTextChanged(CharSequence s, int start, int before, int count) {
        mStr = s;
    }

    @Override
    public void afterTextChanged(Editable s) {
        if (mStr == null || mStr.length() == 0)
            return;
        if (mStr.length() == 11 && mMaxLength == 11) {
            ViewUtil.hideAllInputMethod(EditHideActivity.this);
        } else if (mStr.length() == 6 && mMaxLength == 6) {
            ViewUtil.hideOneInputMethod(EditHideActivity.this, mView);
        }
    }
}
}

```

(3)

INPUT_METHOD_SERVICE

①toggleSoftInput

```

public static void hideAllInputMethod(Activity act) {
    InputMethodManager imm = (InputMethodManager)
act.getSystemService(Context.INPUT_METHOD_SERVICE);
    if (imm.isActive() == true) { //软键盘如果已经打开就要关闭
        imm.toggleSoftInput(0, InputMethodManager.HIDE_NOT_ALWAYS);
    }
}
}

```

②hideSoftInputFromWindow

```
public static void hideOneInputMethod(Activity act, View v) {  
    InputMethodManager imm = (InputMethodManager)  
act.getSystemService(Context.INPUT_METHOD_SERVICE);  
    imm.hideSoftInputFromWindow(v.getWindowToken(), 0);  
}
```

3-151011113-16

4.

()EditTextTextWatcher



3-1510

Middle

15960238696

输入6位时自动隐藏输入法

点击外部空白处隐藏输入法

3-1611

```

private class JumpTextWatcher implements TextWatcher {
    private EditText mThisView = null;
    private View mViewNext = null;

    public JumpTextWatcher(EditText vThis, View vNext) {
        super();
        mThisView = vThis;
        if (vNext != null) {
            mViewNext = vNext;
        }
    }

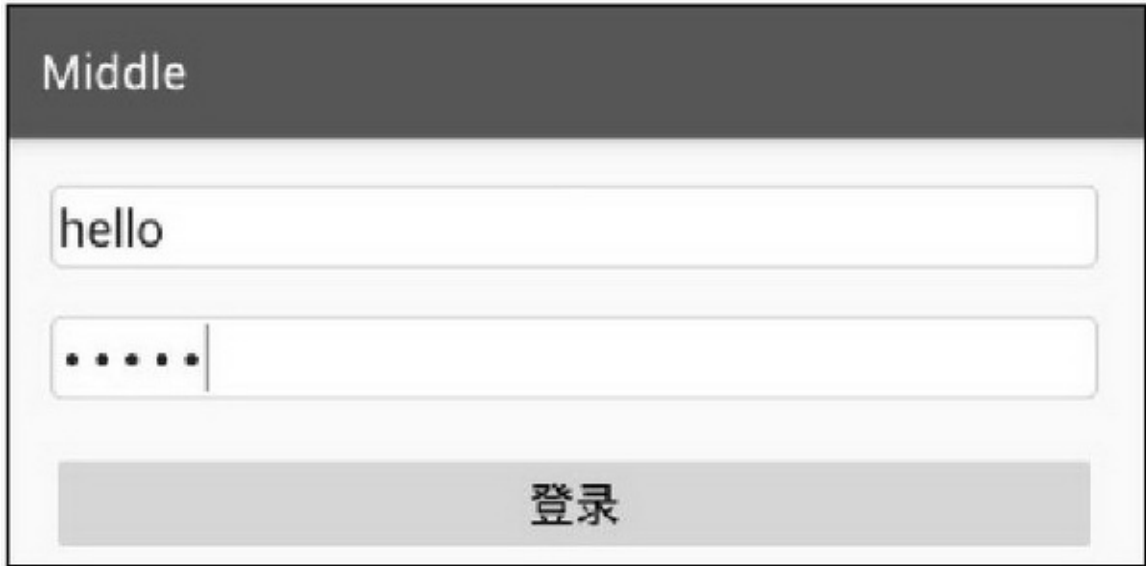
    @Override
    public void beforeTextChanged(CharSequence s, int start, int count, int after) {
    }

    @Override
    public void onTextChanged(CharSequence s, int start, int before, int count) {
    }

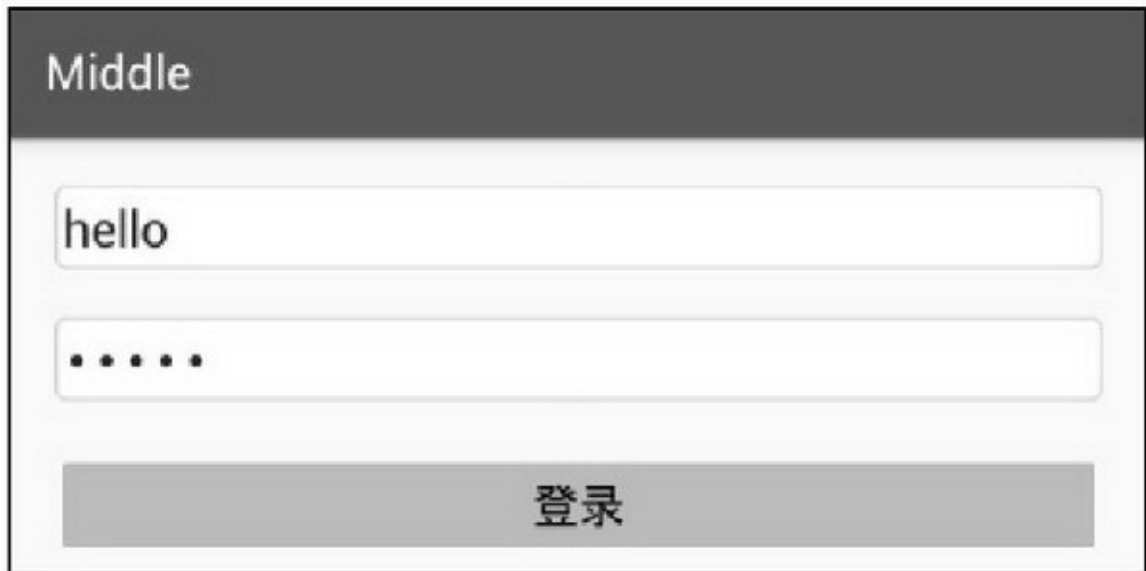
    @Override
    public void afterTextChanged(Editable s) {
        String str = s.toString();
        if (str.indexOf("\r") >= 0 || str.indexOf("\n") >= 0) { //发现输入回车符或换行符
            mThisView.setText(str.replace("\r", "").replace("\n", "")); //去掉回车符和换行符
            if (mViewNext != null) {
                mViewNext.requestFocus(); //让下一个视图获得焦点，即将光标移到下个视图
                if (mViewNext instanceof EditText) {
                    EditText et = (EditText)mViewNext;
                    //让光标自动移到编辑框内部的文本末尾
                    //方式一：直接调用 EditText 的 setSelection 方法
                    et.setSelection(et.getText().length());
                    //方式二：调用 Selection 类的 setSelection 方法
                    //Editable edit = et.getText();
                    //Selection.setSelection(edit, edit.length());
                }
            }
        }
    }
}

```

3-17“”()3-18



3-17



3-18

3.4.2 AutoCompleteTextView App

AutoCompleteTextView EditText TextWatcher Spinner EditText
AutoCompleteTextView 3-4
EditText

3-4

XML 中的属性	AutoCompleteTextView 类的设置方法	说明
completionHint	setCompletionHint	设置下拉列表底部的提示文字
completionThreshold	setThreshold	设置至少输入多少个字符才会显示提示
dropDownHorizontalOffset	setDropDownHorizontalOffset	设置下拉列表与文本框之间的水平偏移
dropDownVerticalOffset	setDropDownVerticalOffset	设置下拉列表与文本框之间的垂直偏移
dropDownHeight	setDropDownHeight	设置下拉列表的高度
dropDownWidth	setDropDownWidth	设置下拉列表的宽度
无	setAdapter	设置下拉列表的数据适配器

AutoCompleteTextView

```
String[] hintArray = {"第一", "第一次", "第一次写代码", "第一次领工资", "第二", "第二个"};
ArrayAdapter<String> adapter = new ArrayAdapter<String>(
    this, R.layout.item_dropdown, hintArray);
AutoCompleteTextView ac_text= (AutoCompleteTextView) findViewById(R.id.ac_text);
ac_text.setAdapter(adapter);
```

3-19



3-19

3.5 Activity

AndroidActivityActivityIntentIntent

3.5.1 Activity

ActivityActivityActivityonCreatestartActivity

ActivityActivity

ActivityActivityActivity

onCreate

onStart

onResume

onPause

onStop

onDestroy

onRestart

Activity3HOMEApp

```
private void refreshLife(String desc) {
    Log.d(TAG, desc);
    mStr = String.format("%s%s %s %s\n", mStr, DateUtil.getNowTimeDetail(), TAG, desc);
    tv_life.setText(mStr);
}

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_act_***);
    tv_life = (TextView) findViewById(R.id.tv_life);
    refreshLife("onCreate");
}

@Override
protected void onStart() {
    refreshLife("onStart");
    super.onStart();
}

@Override
protected void onStop() {
    refreshLife("onStop");
    super.onStop();
}

@Override
protected void onResume() {
```

```
        refreshLife("onResume");
        super.onResume();
    }

    @Override
    protected void onPause() {
        refreshLife("onPause");
        super.onPause();
    }

    @Override
    protected void onRestart() {
        refreshLife("onRestart");
        super.onRestart();
    }

    @Override
    protected void onDestroy() {
        refreshLife("onDestroy");
        super.onDestroy();
    }
}
```

1.

Middle	
跳到下个页面	
20:30:20.916 ActJumpActivity onCreate	1
20:30:20.916 ActJumpActivity onStart	
20:30:20.916 ActJumpActivity onResume	
20:30:22.524 ActJumpActivity onPause	2
20:30:22.972 ActJumpActivity onStop	
20:30:40.657 ActJumpActivity	
20:30:22.568 ActNextActivity onCreate	
20:30:22.568 ActNextActivity onStart	
20:30:22.568 ActNextActivity onResume	
20:30:40.661 ActJumpActivity onActivityResult	3
20:30:40.661 ActJumpActivity onRestart	
20:30:40.661 ActJumpActivity onStart	
20:30:40.661 ActJumpActivity onResume	

3-20

ActJumpActivityActNextActivityActNextActivityActJumpActivity3-
201ActJumpActivity2ActNextActivity3ActJumpActivity

logcat

ActJumpActivityonCreate → onStart → onResume

11:30:18.352D/ActJumpActivity(2315)onCreate 11:30:18.352

D/ActJumpActivity(2315)onStart 11:30:18.352D/ActJumpActivity(2315)
onResume ActJumpActivityActNextActivityonPause →

onCreate → onStart → onResume → onStop

11:30:32.668D/ActJumpActivity(2315)onPause 11:30:32.688

D/ActNextActivity(2315)onCreate 11:30:32.688D/ActNextActivity(2315)
onStart 11:30:32.688D/ActNextActivity(2315)onResume 11:30:33.116
D/ActJumpActivity(2315)onStop ActNextActivityActJumpActivity(finish
)onPause → onRestart → onStart → onResume → onStop → onDestroy
11:30:40.740D/ActNextActivity(2315)onPause 11:30:40.752
D/ActJumpActivity(2315)onRestart 11:30:40.752
D/ActJumpActivity(2315)onStart 11:30:40.752D/ActJumpActivity(2315)
onResume 11:30:41.160D/ActNextActivity(2315)onStop 11:30:41.164
D/ActNextActivity(2315)onDestroy

Middle
20:34:11.106 ActRotateActivity onCreate 20:34:11.106 ActRotateActivity onStart 20:34:11.106 ActRotateActivity onResume
Middle
20:36:24.558 ActRotateActivity onCreate 20:36:24.558 ActRotateActivity onStart 20:36:24.558 ActRotateActivity onResume
Middle
20:40:11.632 ActRotateActivity onCreate 20:40:11.632 ActRotateActivity onStart 20:40:11.640 ActRotateActivity onResume

onPauseonCreate/onRestart → onStart → onResumeonStop(
onDestroy)

2.

ActRotateActivity333-21

logcat

21:02:10.179 D/ActRotateActivityonCreate 21:02:10.179

D/ActRotateActivityonStart 21:02:10.179 D/ActRotateActivityonResume

21:02:13.227 D/ActRotateActivityonPause 21:02:13.227

D/ActRotateActivityonStop 21:02:13.227 D/ActRotateActivityonDestroy

21:02:13.247 D/ActRotateActivityonCreate 21:02:13.247

D/ActRotateActivityonStart 21:02:13.247 D/ActRotateActivityonResume

21:02:16.239 D/ActRotateActivityonPause 21:02:16.239

D/ActRotateActivityonStop 21:02:16.239 D/ActRotateActivityonDestroy

21:02:16.279 D/ActRotateActivityonCreate 21:02:16.279

D/ActRotateActivityonStart 21:02:16.279 D/ActRotateActivityonResume

onPauseonStoponDestroyonCreateonStartonResume

3.HOMEApp

Middle

20:27:42.586 ActHomeActivity onCreate

20:27:42.586 ActHomeActivity onStart

20:27:42.586 ActHomeActivity onResume

20:27:47.538 ActHomeActivity onPause

20:27:47.690 ActHomeActivity onStop

20:27:51.682 ActHomeActivity onRestart

20:27:51.682 ActHomeActivity onStart

20:27:51.682 ActHomeActivity onResume

3-22HOME

ActHomeActivityHOMEHOME()App3-22
()(App)logcat

3.5.2Intent

IntentIntentActivityActivityServiceActivity
BroadcastIntentAndroid3

(1)Intent

(2)Intent

(3)Intent

IntentIntent3-5

3-5Intent

元素名称	设置方法	说明与用途
Component	setComponent	组件，用于指定 Intent 的来源与目的
Action	setAction	动作，用于指定 Intent 的操作行为
Data	setData	即 Uri，用于指定动作要操纵的数据路径
Category	addCategory	类别，用于指定 Intent 的操作类别
Type	setType	数据类型，用于指定 Data 类型的定义
Extras	putExtra	扩展信息，用于指定装载的参数信息
Flags	setFlags	标志位，用于指定 Intent 的运行模式（启动标志）

IntentIntentIntent

1.Intent

Intent3

(1)

Intent intent = new Intent(this, ActResponseActivity.class); (2)setClass

Intent intent = new Intent();

intent.setClass(this, ActResponseActivity.class); (3)setComponent

Intent intent = new Intent();


```
ComponentName component = new ComponentName(this,
ActResponseActivity.class); intent.setComponent(component);
```

2.Intent

Intent3-6

3-6

Intent 类的系统动作常量名	系统动作的常量值	说明
ACTION_MAIN	android.intent.action.MAIN	App 启动时的入口
ACTION_VIEW	android.intent.action.VIEW	显示数据给用户
ACTION_EDIT	android.intent.action.EDIT	显示可编辑的数据
ACTION_CALL	android.intent.action.CALL	拨号
ACTION_DIAL	android.intent.action.DIAL	打电话
ACTION_SEND	android.intent.action.SEND	发短信
ACTION_ANSWER	android.intent.action.ANSWER	接电话
ACTION_SEARCH	android.intent.action.SEARCH	导航栏上 SearchView 的搜索动作

```
setActionIntent(String action)Intent
```

```
UriCategoryUriIntent(String action, Uri uri)setData(setData
```

```
setUri)CategoryaddCategoryaddsetIntentCategory
```

Uri

```
Intent intent = new Intent();
```

```
intent.setAction(Intent.ACTION_CALL);
```

```
Uri uri = Uri.parse("tel:"+ "15960238696"); intent.setData(uri);
```

```
startActivity(intent);
```

```
IntentAndroidAndroidManifest.xmlintent-filterXML
```

```
MainAcitivityactivityactioncategoryandroid.intent.action.MAINApp
```

```
android.intent.category.LAUNCHERApp
```

```

<activity
    android:name=".MainActivity"
    android:label="@string/app_name" >
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>

```

3.5.3Activity

IntentsetDataExtrasIntentputExtraStringintdouble
 ParcelableSerializableputExtraAndroidBundle
 BundleBundle
 BundleMapBundleputExtrasBundlegetExtras

```

Intent intent = new Intent(MainActivity.this, FirstActivity.class); Bundle
bundle = new Bundle();
bundle.putString("name", "");
bundle.putInt("age", 30);
bundle.putDouble("height", 170.0f);
intent.putExtra(bundle);
startActivity(intent);

```

```

Intent intent = getIntent();
Bundle bundle = intent.getExtras();
String name = bundle.getString("name", ""); int age = bundle.getInt("age",
0);
double height = bundle.getDouble("height", 0.0f);

```

3.5.4Activity
Intent()startActivity

步骤 01 startActivityForResult(Intent intent, intrequestCode)

步骤 02

步骤 03

setResultSetResult()

```
Intent intent = new Intent();  
Bundle bundle = new Bundle();  
bundle.putString("job", "");  
intent.putExtras(bundle);  
setResult(Activity.RESULT_OK, intent);  
finish(); //
```

步骤 04

onActivityResult

```
@Override  
public void onActivityResult(int requestCode, int resultCode, Intent intent) {  
    Log.d(TAG, "onActivityResult. requestCode="+requestCode+", resultCode="+resultCode);  
    Bundle resp = intent.getExtras();  
    String job = resp.getString("job");  
    Toast.makeText(this, "您目前的职业是"+job, Toast.LENGTH_LONG).show();  
}
```

Activity

```

public class ActRequestActivity extends AppCompatActivity implements OnClickListener {
    private EditText et_request;
    private TextView tv_request;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_act_request);
        findViewById(R.id.btn_act_request).setOnClickListener(this);
        et_request = (EditText) findViewById(R.id.et_request);
        tv_request = (TextView) findViewById(R.id.tv_request);
    }

    @Override
    public void onClick(View v) {
        if (v.getId() == R.id.btn_act_request) {
            Intent intent = new Intent();
            intent.setClass(this, ActResponseActivity.class);
            intent.putExtra("request_time", DateUtil.getNowTime());
            intent.putExtra("request_content", et_request.getText().toString());
            startActivityForResult(intent, 0);
        }
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        if (data != null) {
            String response_time = data.getStringExtra("response_time");
            String response_content = data.getStringExtra("response_content");
            String desc = String.format("收到返回消息 : \n 应答时间为%s\n 应答内容为%s",
                response_time, response_content);
            tv_request.setText(desc);
        }
    }
}

```

```

public class ActResponseActivity extends AppCompatActivity implements OnClickListener {
    private EditText et_response;
    private TextView tv_response;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_act_response);
        findViewById(R.id.btn_act_response).setOnClickListener(this);
        et_response = (EditText) findViewById(R.id.et_response);
        tv_response = (TextView) findViewById(R.id.tv_response);
        Bundle bundle = getIntent().getExtras();
        String request_time = bundle.getString("request_time");
        String request_content = bundle.getString("request_content");
        String desc = String.format("收到请求消息 : \n 请求时间为%s\n 请求内容为%s",
            request_time, request_content);
        tv_response.setText(desc);
    }

    @Override
    public void onClick(View v) {
        if (v.getId() == R.id.btn_act_response) {
            Intent intent = new Intent();
            Bundle bundle = new Bundle();
            bundle.putString("response_time", DateUtil.getNowTime());
            bundle.putString("response_content", et_response.getText().toString());
            intent.putExtras(bundle);
            setResult(Activity.RESULT_OK, intent);
            finish();
        }
    }
}

```

3-233-243-253-233-243-25

Middle

你吃了没？去我家吃饭，走走

传送请求参数

3-23

Middle

收到请求消息：

请求时间为20:25:55

请求内容为你吃了没？去我家吃饭，走走

我家饭菜都做好了，还是你来我家吃饭，来来

返回应答参数

3-24

Middle

你吃了没？去我家吃饭，走走

传送请求参数

收到返回消息：

应答时间为20:27:10

应答内容为我家饭菜都做好了，还是你来我家吃饭，来来

3.6App AppApp

3.6.1

AppAppApp3-26RelativeLayout
EditTextRadioButtonCheckBoxSpinner
AppActivityActivityAndroid
AppApp3-27
6
RadioButton
Spinner
EditText
CheckBox
RelativeLayoutTextViewLinearLayout
FrameLayout

Middle

购房总价： 万

按揭部分： %

计算贷款总额

其中贷款部分为：***万

还款方式： 等额本息 等额本金

商业贷款： 万

公积金贷款： 万

贷款年限： ▾

基准利率： ▾

计算还款明细

还款总额为：***万
其中利息总额为：***万
月供(每月还款额)为：***

Middle

密码登录

验证码登录

我是：

个人用户

手机号码：请输入手机号码

登录密码：请输入密码

忘记密码

记住密码

登录



3-28

“”3-28

AppApponActivityResult
onActivityResult
ActivityonRestart()App

3.6.2AlertDialog AppApp

“”“”(AlertDialog)

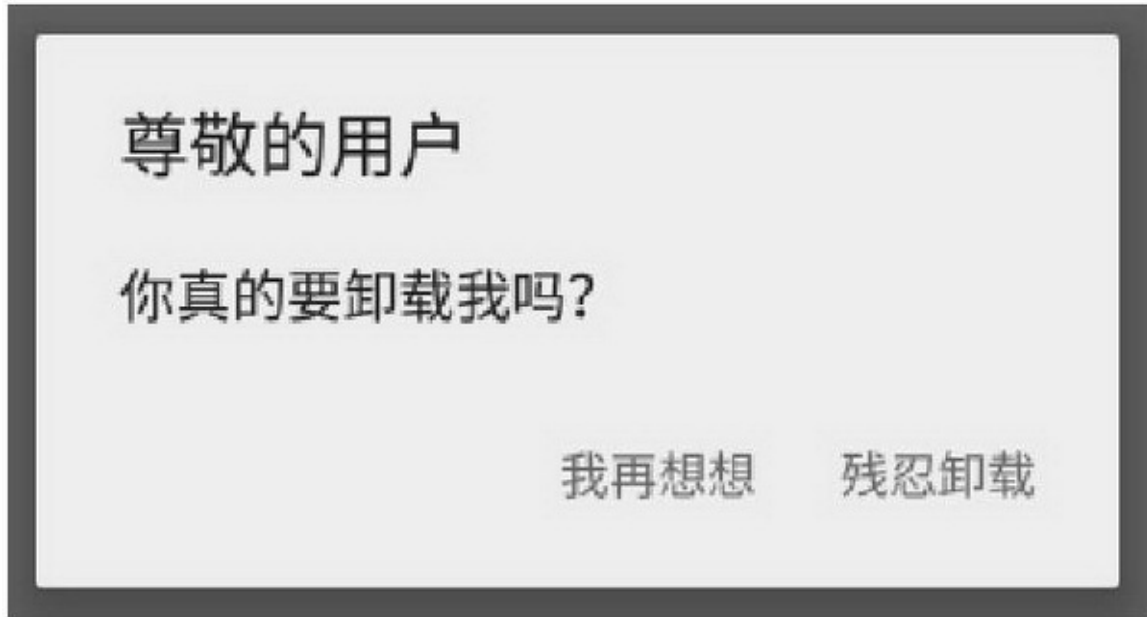
AlertDialogAndroidAlertDialogAlertDialog.Builder
AlertDialog.Builder
setIcon
setTitle
setMessage
setPositiveButton
setNegativeButton

setNeutralButton

AlertDialog.Builder.createAlertDialog.AlertDialog.show

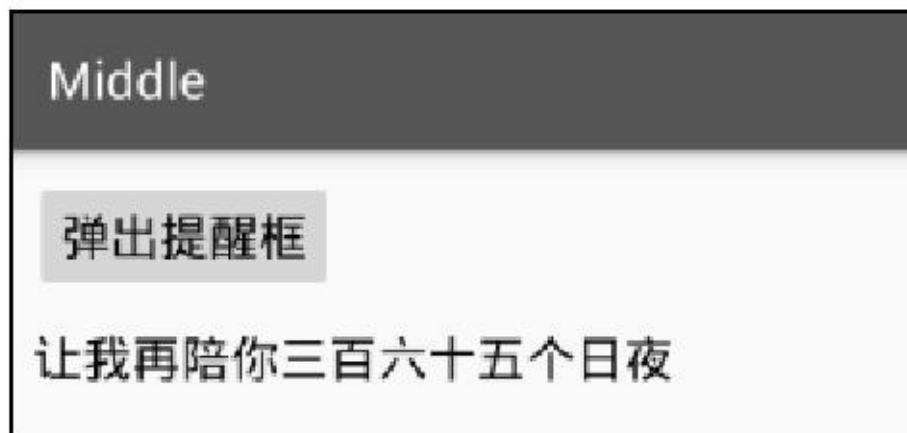
```
public void onClick(View v) {
    if (v.getId() == R.id.btn_alert) {
        AlertDialog.Builder builder = new AlertDialog.Builder(this);
        builder.setTitle("尊敬的用户");
        builder.setMessage("你真的要卸载我吗? ");
        builder.setPositiveButton("残忍卸载", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                tv_alert.setText("虽然依依不舍, 但是只能离开了");
            }
        });
        builder.setNegativeButton("我再想想", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                tv_alert.setText("让我再陪你三百六十五个日夜");
            }
        });
        AlertDialog alert = builder.create();
        alert.show();
    }
}
```

3-29



3-29AlertDialog

3-30“”3-31“”



3-30“”



3-31“”

3.6.3

3

步骤 01

LoginMainActivity.javaactivity_login.xml

LoginForgetActivity.javaactivity_login_forget.xmlAndroidManifest.xml
activity

```
<activity android:name=".LoginMainActivity" /> <activity
```

```
android:name=".LoginForgetActivity" /> 步骤 02 res/layout  
activity_login.xmlactivity_login_forget.xml
```

步骤 03

LoginMainActivityLoginForgetActivity

LoginMainActivity.java

```
public void onClick(View v) {
    String phone = et_phone.getText().toString();
    if (v.getId() == R.id.btn_forget) {
        if (phone==null || phone.length()<11) {
            Toast.makeText(this, "请输入正确的手机号", Toast.LENGTH_SHORT).show();
            return;
        }
        if (rb_password.isChecked() == true) {
            Intent intent = new Intent(this, LoginForgetActivity.class);
            intent.putExtra("phone", phone);
            startActivityForResult(intent, mRequestCode);
        } else if (rb_verifycode.isChecked() == true) {
            mVerifyCode = String.format("%06d", (int)(Math.random()*1000000%1000000));
            AlertDialog.Builder builder = new AlertDialog.Builder(this);
            builder.setTitle("请记住验证码");
            builder.setMessage("手机号"+phone+", 本次验证码是"+mVerifyCode+", 请输入验证码");
            builder.setPositiveButton("好的", null);
            AlertDialog alert = builder.create();
            alert.show();
        }
    }
}
```

```

    }
} else if (v.getId() == R.id.btn_login) {
    if (phone == null || phone.length() < 11) {
        Toast.makeText(this, "请输入正确的手机号", Toast.LENGTH_SHORT).show();
        return;
    }
    if (rb_password.isChecked() == true) {
        if (et_password.getText().toString().equals(mPassword) != true) {
            Toast.makeText(this, "请输入正确的密码", Toast.LENGTH_SHORT).show();
            return;
        } else {
            loginSuccess();
        }
    } else if (rb_verifycode.isChecked() == true) {
        if (et_password.getText().toString().equals(mVerifyCode) != true) {
            Toast.makeText(this, "请输入正确的验证码", Toast.LENGTH_SHORT).show();
            return;
        } else {
            loginSuccess();
        }
    }
}
}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    if (requestCode == mRequestCode && data != null) {
        //用户密码已改为新密码
        mPassword = data.getStringExtra("new_password");
    }
}

//从修改密码页面返回登录页面，要清空密码的输入框
@Override
protected void onRestart() {
    et_password.setText("");
    super.onRestart();
}

private void loginSuccess() {
    String desc = String.format("您的手机号码是%s，类型是%s。恭喜你通过登录验证，点击“确定”按钮返回上个页面", et_phone.getText().toString(), typeArray[mType]);
    AlertDialog.Builder builder = new AlertDialog.Builder(this);
    builder.setTitle("登录成功");

```

```

        builder.setMessage(desc);
        builder.setPositiveButton("确定返回", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                finish();
            }
        });
        builder.setNegativeButton("我再看看", null);
        AlertDialog alert = builder.create();
        alert.show();
    }
}

```

LoginForgetActivity.java


```

public class LoginForgetActivity extends AppCompatActivity implements OnClickListener {
    private EditText et_password_first;
    private EditText et_password_second;
    private EditText et_verifycode;
    private String mVerifyCode;
    private String mPhone;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login_forget);
        et_password_first = (EditText) findViewById(R.id.et_password_first);
        et_password_second = (EditText) findViewById(R.id.et_password_second);
        et_verifycode = (EditText) findViewById(R.id.et_verifycode);
        findViewById(R.id.btn_verifycode).setOnClickListener(this);
        findViewById(R.id.btn_confirm).setOnClickListener(this);
        mPhone = getIntent().getStringExtra("phone");
    }

    @Override
    public void onClick(View v) {
        if (v.getId() == R.id.btn_verifycode) {
            if (mPhone == null || mPhone.length() < 11) {
                Toast.makeText(this, "请输入正确的手机号", Toast.LENGTH_SHORT).show();
                return;
            }
            mVerifyCode = String.format("%06d", (int) (Math.random() * 1000000 % 1000000));
            AlertDialog.Builder builder = new AlertDialog.Builder(this);
            builder.setTitle("请记住验证码");
            builder.setMessage("手机号"+mPhone+", 本次验证码是"+mVerifyCode+", 请输入验证码");
            builder.setPositiveButton("好的", null);
        }
    }
}

```

```
AlertDialog alert = builder.create();
alert.show();
} else if (v.getId() == R.id.btn_confirm) {
    String password_first = et_password_first.getText().toString();
    String password_second = et_password_second.getText().toString();
    if (password_first==null || password_first.length()<6 ||
        password_second==null || password_second.length()<6) {
        Toast.makeText(this, "请输入正确的新密码", Toast.LENGTH_SHORT).show();
        return;
    } else if (password_first.equals(password_second) != true) {
        Toast.makeText(this, "两次输入的新密码不一致", Toast.LENGTH_SHORT).show();
        return;
    } else if (et_verifycode.getText().toString().equals(mVerifyCode) != true) {
        Toast.makeText(this, "请输入正确的验证码", Toast.LENGTH_SHORT).show();
        return;
    } else {
        Toast.makeText(this, "密码修改成功", Toast.LENGTH_SHORT).show();
        Intent intent = new Intent();
        intent.putExtra("new_password", password_first);
        setResult(Activity.RESULT_OK, intent);
        finish();
    }
}
}
}
```

3.7

App()()Activity(
)“App”AppActivity

3

(1)

(2)

(3)Activity

第4章 数据存储



AndroidSharedPreferencesSQLiteSDAppApplication

“”

4.1 Shared Preferences

Android——SharedPreferences“App”

4.1.1

SharedPreferencesAndroidKey-ValueJavaPropertiesKey-Value
PropertiesKey=ValueSharedPreferencesXMLSharedPreferences
/data/data//shared_prefs/.xmlXML

第4章 数据存储



AndroidSharedPreferencesSQLiteSDAppApplication

“”

4.1 Shared Preferences Android——Shared Preferences “App”

4.1.1

SharedPreferencesAndroidKey-ValueJavaPropertiesKey-Value
PropertiesKey=ValueSharedPreferencesXMLSharedPreferences
/data/data//shared_prefs/.xmlXML

```
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>  
<map>  
  <string name="name">Mr Lee</string>  
  <int name="age" value="30" />  
  <boolean name="married" value="true" />  
  <float name="weight" value="100.0" />  
</map>
```

XMLSharedPreferences

(1)

(2)

(3)App

AppApp

SharedPreferencesMapputgetgetSharedPreferences

SharedPreferences sps= getSharedPreferences("share",

Context.MODE_PRIVATE); getSharedPreferencesshare.xml

MODE_PRIVATE

Editor

SharedPreferences.Editor editor = sps.edit(); editor.putString("name", "Mr
Lee");

editor.putInt("age", 30);

```
editor.putBoolean("married", true);
editor.putFloat("weight", 100f);
editor.commit();
get
String name = sps.getString("name", ""); int age = sps.getInt("age", 0);
boolean married = sps.getBoolean("married", false); float weight =
sps.getFloat("weight", 0); SharedPreferences4-1EditText
App4-2
```

The screenshot displays a form titled "Storage" with the following fields and values:

- 姓名 : Jack
- 年龄 : 30
- 身高 : 175
- 体重 : 70
- 婚否 : 已婚

At the bottom of the form is a button labeled "保存到共享参数".

Storage

共享参数中保存的信息如下：

married的取值为true

weight的取值为70.000000

height的取值为175

count的取值为4

update_time的取值为2016-10-01 18:25:18

age的取值为30

name的取值为Jack

first的取值为false

4-2

4.1.2

“App”“”App

3

(1)SharedPreferences.onCreate()getSharedPreferences

(2)“”loginSuccess

```
if (bRemember) {  
    SharedPreferences.Editor editor = mShared.edit();  
    editor.putString("phone", et_phone.getText().toString());  
    editor.putString("password", et_password.getText().toString());  
    editor.commit();  
}
```

(3)App.onCreate

```
String phone = mShared.getString("phone", ""); String password =  
mShared.getString("password", ""); et_phone.setText(phone);  
et_password.setText(password);  
“”App4-34-44-3“”4-4
```

The image shows a screenshot of a mobile application's login screen. At the top, there is a dark header with the word "Storage" in white. Below the header, there are two radio buttons: "密码登录" (Password Login) which is selected, and "验证码登录" (Verification Code Login). Underneath, there is a label "我是：" (I am:) followed by a dropdown menu currently showing "个人用户" (Personal User). There are two input fields: "手机号码：" (Mobile Number:) containing the text "15960238696", and "登录密码：" (Login Password:) which is masked with dots. To the right of the password field is a button labeled "忘记密码" (Forgot Password). At the bottom left, there is a checked checkbox labeled "记住密码" (Remember Password). A large, light gray button labeled "登录" (Login) is centered at the bottom of the form.

Storage

密码登录 验证码登录

我是： 个人用户 ▼

手机号码： 15960238696

登录密码： [忘记密码](#)

记住密码

登录

4.2 SQLite Android—— SQLite SQLite“App”

4.2.1 SQLite

SQLite iOS Android SQLite SQLite sql Oracle

(1) IF NOT EXISTS CREATE TABLE IF NOT EXISTS

table_name

(2) IF EXISTS DROP TABLE IF EXISTS table_name

(3) ALTER TABLE table_name ADD COLUMN ... Oracle COLUMN

(4) SQLite ALTER

(5) SQLite INTEGER VARCHAR FLOAT SQLite 010

false 1 true

(6) SQLite _id _id INTEGER PRIMARY KEY

AUTO INCREMENT NOT NULL

(7)

SQLite Database SQLite Context

```
//创建数据库，如果已存在就打开
SQLiteDatabase db = getApplicationContext().openOrCreateDatabase("test.db",
Context.MODE_PRIVATE, null);
//删除数据库
getApplicationContext().deleteDatabase("test.db");
```

SQLite Database API 3

1.

openDatabase

isOpen

close

getVersion

setVersion

2.

beginTransaction

setTransactionSuccessful

endTransactionsetTransactionSuccessful

3.

execSQLSQL

delete

update

insert

query

rawQuerySQL

4.2.2 SQLiteOpenHelper SQLiteDatabaseAndroid

—— SQLiteOpenHelperSQLite

SQLiteOpenHelper

步骤 01

SQLiteOpenHelper onCreateonUpgrade
onCreateonUpgradeonUpgrade

步骤 02

App

SQLiteOpenHelpergetReadableDatabase

getWritableDatabase

SQLiteDatabaseclose

步骤 03

SQLiteContentValuesMapputgetContentValuesContentValues

HashMap“private HashMap<String, Object> mValues;”ContentValues

SQLiteDatabaseinsertupdate

CursorSQLiteDatabasequeryrawQueryCursorCursor3

1.

close

isClosed

isFirst

isLast

2.

moveToFirst

moveToLast

moveToNext

moveToPrevious

move

moveToPosition

3.

getCount

getInt

getFloat

getString

getType

4-54-6SQLiteSQLite4-7

Storage

姓名：

年龄：

身高：

体重：

婚否：

Storage

姓名： Lucy

年龄： 25

身高： 165

体重： 50

婚否： 未婚

保存到数据库

Storage

删除所有记录

数据库查询到2条记录，详情如下：

第1条记录信息如下：

姓名为Jack

年龄为30

身高为175

体重为70.000000

婚否为true

更新时间为2016-10-01 18:29:27

第2条记录信息如下：

姓名为Lucy

年龄为25

身高为165

体重为50.000000

婚否为false

更新时间为2016-10-01 18:29:48

4-7SQLite

SQLiteOpenHelper

```
public class UserDBHelper extends SQLiteOpenHelper {  
    private static final String TAG = "UserDBHelper";  
    private static final String DB_NAME = "user.db";  
}
```

```
private static final int DB_VERSION = 1;
private static UserDBHelper mHelper = null;
private SQLiteDatabase mDB = null;
private static final String TABLE_NAME = "user_info";

private UserDBHelper(Context context) {
    super(context, DB_NAME, null, DB_VERSION);
}

private UserDBHelper(Context context, int version) {
    super(context, DB_NAME, null, version);
}

public static UserDBHelper getInstance(Context context, int version) {
    if (version > 0 && mHelper == null) {
        mHelper = new UserDBHelper(context, version);
    } else if (mHelper == null) {
        mHelper = new UserDBHelper(context);
    }
    return mHelper;
}

public SQLiteDatabase openReadLink() {
    if (mDB == null || mDB.isOpen() != true) {
        mDB = mHelper.getReadableDatabase();
    }
    return mDB;
}

public SQLiteDatabase openWriteLink() {
    if (mDB == null || mDB.isOpen() != true) {
        mDB = mHelper.getWritableDatabase();
    }
    return mDB;
}

public void closeLink() {
    if (mDB != null && mDB.isOpen() == true) {
        mDB.close();
        mDB = null;
    }
}
```

```

public String getDBName() {
    if (mHelper != null) {
        return mHelper.getDatabaseName();
    } else {
        return DB_NAME;
    }
}

@Override
public void onCreate(SQLiteDatabase db) {
    Log.d(TAG, "onCreate");
    String drop_sql = "DROP TABLE IF EXISTS " + TABLE_NAME + ";";
    db.execSQL(drop_sql);
    String create_sql = "CREATE TABLE IF NOT EXISTS " + TABLE_NAME + " ("
        + "_id INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,"
        + "name VARCHAR NOT NULL," + "age INTEGER NOT NULL,"
        + "height LONG NOT NULL," + "weight FLOAT NOT NULL,"
        + "married INTEGER NOT NULL," + "update_time VARCHAR NOT NULL"
        //演示数据库升级时要先注释下面这行代码
        + ",phone VARCHAR" + ",password VARCHAR" + ");";
    Log.d(TAG, "create_sql:" + create_sql);
    db.execSQL(create_sql);
}

@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    Log.d(TAG, "onUpgrade oldVersion=" + oldVersion + ", newVersion=" + newVersion);
    if (newVersion > 1) {
        //Android 的 ALTER 命令不支持一次添加多列，只能分多次添加
        String alter_sql = "ALTER TABLE " + TABLE_NAME + " ADD COLUMN " + "phone
VARCHAR;";
        Log.d(TAG, "alter_sql:" + alter_sql);
        db.execSQL(alter_sql);
        alter_sql = "ALTER TABLE " + TABLE_NAME + " ADD COLUMN " + "password
VARCHAR;";
        Log.d(TAG, "alter_sql:" + alter_sql);
        db.execSQL(alter_sql);
    }
}

public int delete(String condition) {
    int count = mDB.delete(TABLE_NAME, condition, null);
    return count;
}

```

```

}

public int deleteAll() {
    int count = mDB.delete(TABLE_NAME, "1=1", null);
    return count;
}

public long insert(UserInfo info) {
    ArrayList<UserInfo> infoArray = new ArrayList<UserInfo>();
    infoArray.add(info);
    return insert(infoArray);
}

public long insert(ArrayList<UserInfo> infoArray) {
    long result = -1;
    for (int i = 0; i < infoArray.size(); i++) {
        UserInfo info = infoArray.get(i);
        ArrayList<UserInfo> tempArray = new ArrayList<UserInfo>();
        // 如果存在同名记录，就更新记录。注意条件语句的等号后面要用单引号括起来
        if (info.name!=null && info.name.length()>0) {
            String condition = String.format("name='%s'", info.name);
            tempArray = query(condition);
            if (tempArray.size() > 0) {
                update(info, condition);
                result = tempArray.get(0).rowid;
                continue;
            }
        }
        // 如果存在同样的手机号码，就更新记录
        if (info.phone!=null && info.phone.length()>0) {
            String condition = String.format("phone='%s'", info.phone);
            tempArray = query(condition);
            if (tempArray.size() > 0) {
                update(info, condition);
                result = tempArray.get(0).rowid;
                continue;
            }
        }
        // 如果不存在唯一性重复的记录，就插入新记录
        ContentValues cv = new ContentValues();
        cv.put("name", info.name);
        cv.put("age", info.age);
        cv.put("height", info.height);
    }
}

```

```

        cv.put("weight", info.weight);
        cv.put("married", info.married);
        cv.put("update_time", info.update_time);
        cv.put("phone", info.phone);
        cv.put("password", info.password);
        result = mDB.insert(TABLE_NAME, "", cv);
        // 添加成功后返回行号, 失败则返回-1
        if (result == -1) {
            return result;
        }
    }
    return result;
}

public int update(UserInfo info, String condition) {
    ContentValues cv = new ContentValues();
    cv.put("name", info.name);
    cv.put("age", info.age);
    cv.put("height", info.height);
    cv.put("weight", info.weight);
    cv.put("married", info.married);
    cv.put("update_time", info.update_time);
    cv.put("phone", info.phone);
    cv.put("password", info.password);
    int count = mDB.update(TABLE_NAME, cv, condition, null);
    return count;
}

public int update(UserInfo info) {
    return update(info, "rowid="+info.rowid);
}

public ArrayList<UserInfo> query(String condition) {
    String sql = String.format("select rowid,_id,name,age,height,weight,married,update_time," +
        "phone,password from %s where %s;", TABLE_NAME, condition);
    Log.d(TAG, "query sql: " + sql);
    ArrayList<UserInfo> infoArray = new ArrayList<UserInfo>();
    Cursor cursor = mDB.rawQuery(sql, null);
    if (cursor.moveToFirst()) {
        for (; cursor.moveToNext()) {
            UserInfo info = new UserInfo();
            info.rowid = cursor.getLong(0);
            info.xuhao = cursor.getInt(1);

```

```

        info.name = cursor.getString(2);
        info.age = cursor.getInt(3);
        info.height = cursor.getLong(4);
        info.weight = cursor.getFloat(5);
        //SQLite 没有布尔型，用 0 表示 false，用 1 表示 true
        info.married = (cursor.getInt(6)==0)?false:true;
        info.update_time = cursor.getString(7);
        info.phone = cursor.getString(8);
        info.password = cursor.getString(9);
        infoArray.add(info);
        if (cursor.isLast() == true) {
            break;
        }
    }
}
cursor.close();
return infoArray;
}

public UserInfo queryByPhone(String phone) {
    UserInfo info = null;
    ArrayList<UserInfo> infoArray = query(String.format("phone=%s", phone));
    if (infoArray.size() > 0) {
        info = infoArray.get(0);
    }
    return info;
}
}
}

```

4.2.3

“4.1.2”

SQLite3

(1)UserDBHelperonResumeonPasue

```

@Override
protected void onResume() {
    super.onResume();
    mHelper = UserDBHelper.getInstance(this, 2);
    mHelper.openWriteLink();
}

@Override
protected void onPause() {
    super.onPause();
    mHelper.closeLink();
}

```

(2)“”loginSuccess

```

if (bRemember) {
    UserInfo info = new UserInfo();
    info.phone = et_phone.getText().toString();
    info.password = et_password.getText().toString();
    info.update_time = DateUtil.getNowDateTime("yyyy-MM-dd HH:mm:ss");
    mHelper.insert(info);
}

```

(3)App

onClickEditTextEditTextAndroid

App.....

et_password.setOnFocusChangeListener(this); onFocusChangeListener
onFocusChange

```

@Override
public void onFocusChange(View v, boolean hasFocus) {
    String phone = et_phone.getText().toString();
    if (v.getId() == R.id.et_password) {
        if (phone.length() > 0 && hasFocus == true) {
            UserInfo info = mHelper.queryByPhone(phone);
            if (info != null) {
                et_password.setText(info.password);
            }
        }
    }
}

```

“”4-84-9

Storage

密码登录 验证码登录

我是： 个人用户 ▾

手机号码：

登录密码： [忘记密码](#)

记住密码

Storage

密码登录

验证码登录

我是：

个人用户

手机号码：15960238696

登录密码：.....|

忘记密码

记住密码

登录

4.3SD

Android——SDSDSDSD

4.3.1SD

(SD)SDSDAppSDAndroid
SDEnvironmentApp7
getRootDirectory
getDataDirectory
getDownloadCacheDirectory
getExternalStorageDirectory(SD)
getExternalStorageStateSD
4-1

4-1SD

Environment 类的存储状态常量名	常量值	常量说明
MEDIA_UNKNOWN	unknown	未知
MEDIA_REMOVED	removed	已经移除
MEDIA_UNMOUNTED	unmounted	未挂载
MEDIA_CHECKING	checking	正在检查
MEDIA_NOFS	nofs	不支持的文件系统
MEDIA_MOUNTED	mounted	已经挂载，且是可读写状态
MEDIA_MOUNTED_READ_ONLY	mounted_ro	已经挂载，且是只读状态
MEDIA_SHARED	shared	当前未挂载，但通过 USB 共享
MEDIA_BAD_REMOVAL	bad_removal	未挂载就被移除
MEDIA_UNMOUNTABLE	unmountable	无法挂载
MEDIA_EJECTING	ejecting	正在弹出

getStorageState
getExternalStoragePublicDirectorySD
4-2

4-2SD

Environment 类的目录类型	常量值	常量说明
DIRECTORY_DCIM	DCIM	相片存放目录（包括相机拍摄的照片和视频）
DIRECTORY_DOCUMENTS	Documents	文档存放目录
DIRECTORY_DOWNLOADS	Download	下载文件存放目录
DIRECTORY_MOVIES	Movies	视频存放目录
DIRECTORY_MUSIC	Music	音乐存放目录
DIRECTORY_PICTURES	Pictures	图片存放目录

SDAndroidManifest.xmlSD

```
<uses-permission  
android:name="android.permission.WRITE_EXTERNAL_STORAGE" />  
<uses-permission  
android:name="android.permission.READ_EXTERNAL_STORAGE" /> <uses-  
permission  
android:name="android.permission.MOUNT_UNMOUNT_FILESYSTEMS" />  
Environment4-10EnvironmentSD
```

Storage

系统环境（含SD卡）的信息如下：

根目录路径：/system

数据目录路径：/data

下载缓存目录路径：/cache

外部存储(即SD卡)目录路径：/storage/emulated/0

外部存储(即SD卡)状态：mounted

SD卡的相机目录路径：/storage/emulated/0/DCIM

SD卡的下载目录路径：/storage/emulated/0/

Download

SD卡的图片目录路径：/storage/emulated/0/

Pictures

SD卡的视频目录路径：/storage/emulated/0/

Movies

SD卡的音乐目录路径：/storage/emulated/0/Music

4-10SD

4.3.2

FileOutputStream
FileInputStream
FileOutputStream
FileInputStreamJava

```
public static void saveText(String path, String txt) {
    try {
        FileOutputStream fos = new FileOutputStream(path);
        fos.write(txt.getBytes());
        fos.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}

public static String openText(String path) {
    String readStr = "";
    try {
        FileInputStream fis = new FileInputStream(path);
        byte[] b = new byte[fis.available()];
        fis.read(b);
        readStr = new String(b);
        fis.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
    return readStr;
}
```

4-11AppSD4-12

Storage

姓名： Jack

年龄： 30

身高： 175

体重： 70

婚否： 已婚

保存文本到SD卡

用户注册信息文件的保存路径为：
/storage/emulated/0/
Download/20161001183721.txt



4-12

4.3.3

AndroidBitmap, AppBitmapFileOutputStreamFileInputStream
BufferedOutputStreamBufferedInputStream
BitmapcompressBitmapFactorydecodeStream

```
public static void saveImage(String path, Bitmap bitmap) {
    try {
        BufferedOutputStream bos = new BufferedOutputStream(new FileOutputStream(path));
        bitmap.compress(Bitmap.CompressFormat.JPEG, 80, bos);
        bos.flush();
        bos.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}

public static Bitmap openImage(String path) {
    Bitmap bitmap = null;
    try {
        BufferedInputStream bis = new BufferedInputStream(new FileInputStream(path));
        bitmap = BitmapFactory.decodeStream(bis);
        bis.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
    return bitmap;
}
```

4-13AppgetDrawingCacheSDSD4-14

Storage

姓名： Lucy

年龄： 25

身高： 165

体重： 50

婚否： 未婚

保存图片到SD卡

用户注册信息图片的保存路径为：
/storage/emulated/0/
Download/20161001183851.png

Storage

删除所有图片文件

文件名 : 20161001183851.png ▼

姓名 : Lucy

年龄 : 25

身高 : 165

体重 : 50

婚否 : 未婚 ▼

4-14

SDBitmapFactorydecodeStreamBitmapFactory
decodeFileBitmap
decodeResourcegetResources()drawableidR.drawable.phone

4.4 Application

AndroidApplicationApplicationApplicationApp

4.4.1 Application

ApplicationAndroidAppApplicationAndroidManifest.xmlactivity
applicationapplicationnameactivityname.MainActivityactivity
MainActivity.javaapplicationname

(1)AndroidManifest.xmlapplicationnameapplication

MainApplication.java

android:name=".MainApplication"

(2)MainApplicationApplication4

onCreateApp

onTerminateApp()

onLowMemory

onConfigurationChanged

(3)AppMainApplicationonCreate(ActivityonCreate)onTerminate

ApponTerminate

onTerminateAndroidThis method is for use in
emulated process environments. It will never be called on a production Android
device, where processes are removed by simply killing them; no user code
(including this callback) is executed when doing so

onTerminateApponTerminate

4.4.2 Application

C/C++Java()

ApplicationAppActivityApplicationApplication

Application3

(1)

(2)logo

(3)Handler

Application3

(1)ApplicationMainApplicationAppgetInstance

(2)ActivityMainApplicationgetInstanceMainApplication

MainApplication

(3)AndroidManifest.xmlApplicationapplicationandroid:name

.MainApplication

4-15AppMainApplicationMainApplication4-16

The image shows a screenshot of an Android application interface. At the top, there is a dark grey header with the word "Storage" in white. Below the header, there are five input fields, each with a label on the left and a text box on the right. The first field is labeled "姓名:" (Name) and contains the text "Xiaoming". The second field is labeled "年龄:" (Age) and contains the text "18". The third field is labeled "身高:" (Height) and contains the text "170". The fourth field is labeled "体重:" (Weight) and contains the text "60". The fifth field is labeled "婚否:" (Married?) and contains the text "未婚" (Unmarried). Below these fields, there is a large, light grey button with the text "保存到全局内存" (Save to Global Memory).

Storage

全局内存中保存的信息如下：

married的取值为未婚

weight的取值为60

height的取值为170

update_time的取值为2016-10-03 12:13:49

age的取值为18

name的取值为Xiaoming

4-16

MainApplicaton

```
public class MainApplication extends Application {
    private static MainApplication mApp;
    public HashMap<String, String> mInfoMap = new HashMap<String, String>();

    public static MainApplication getInstance() {
        return mApp;
    }

    @Override
    public void onCreate() {
        super.onCreate();
        mApp = this;
    }
}
```

`ActivityMainApplication.getInstance().mInfoMap`

4.5

App

4.5.1

4-174-18



购物车 				
图片	名称	数量	单价	总价
	Mate8 华为 HUAWEI Mate8 3GB+32GB版 移动联通4G手机 (月光银)	2	2499	4998
	iphone7 Apple iPhone 7 128GB 玫瑰金色 移动联通电信4G手机	1	5888	5888
	小米5 小米手机5 全网通 高配版 3GB内存 64GB 白色	1	1799	1799
	vivo X6S vivo X6S 金色 全网通4G 双卡 双待 4GB+64GB	1	2298	2298
总金额：14983			结 算	

4-18

4-194-20

SQLiteSQLiteSQLite

手机商场



iphone7



5888.0

加入购物车

Mate8



2499.0

加入购物车

小米5



1799.0

加入购物车

vivo X6S



2298.0

加入购物车



4-20

SharedPreferences.countApp0

SD()AppSDAppSD

SDSDSD

4.5.2 Menu App

AndroidOptionMenuWindowsContextMenu

Windowsres/layoutres/menu

Android

1.OptionMenu

3

(1)

(2)openOptionsMenu

(3)

onCreateOptionsMenuXML

onOptionsItemSelected

menuitem

```
<menu xmlns:android="http://schemas.android.com/apk/res/android" >
  <item
    android:id="@+id/menu_change_time"
    android:orderInCategory="1"
    android:title="改变时间"/>
  <item
    android:id="@+id/menu_change_color"
    android:orderInCategory="8"
    android:title="改变颜色"/>
  <item
    android:id="@+id/menu_change_bg"
    android:orderInCategory="9"
    android:title="改变背景"/>
</menu>
```

```

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.menu_option, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    int id = item.getItemId();
    if (id == R.id.menu_change_time) {
        setRandomTime();
    } else if (id == R.id.menu_change_color) {
        tv_option.setTextColor(getRandomColor());
    } else if (id == R.id.menu_change_bg) {
        tv_option.setBackgroundColor(getRandomColor());
    }
    return true;
}

private void setRandomTime() {
    String desc = DateUtil.getNowDateTime("yyyy-MM-dd HH:mm:ss") + " 这里是菜单显示文本";
    tv_option.setText(desc);
}

private int[] mColorArray = {
    Color.BLACK, Color.WHITE, Color.RED, Color.YELLOW, Color.GREEN,
    Color.BLUE, Color.CYAN, Color.MAGENTA, Color.GRAY, Color.DKGRAY };
private int getRandomColor() {
    int random = (int) (Math.random()*10 % 10);
    return mColorArray[random];
}

```

openOptionsMenu4-21



4-21

2.ContextMenu

(1)onStartregisterForContextMenuonStop

unregisterForContextMenu

(2)registerForContextMenuopenContextMenu

unregisterForContextMenu

onCreateContextMenuXML

onContextItemSelected

```
@Override
protected void onResume() {
    registerForContextMenu(tv_context);
    super.onResume();
}

@Override
protected void onPause() {
    unregisterForContextMenu(tv_context);
    super.onPause();
}

@Override
public void onCreateContextMenu(ContextMenu menu, View v, ContextMenuInfo menuInfo) {
    getMenuInflater().inflate(R.menu.menu_option, menu);
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    int id = item.getItemId();
    if (id == R.id.menu_change_time) {
        setRandomTime();
    } else if (id == R.id.menu_change_color) {
        tv_context.setTextColor(getRandomColor());
    } else if (id == R.id.menu_change_bg) {
        tv_context.setBackgroundColor(getRandomColor());
    }
    return true;
}
```

4-22

改变时间

改变颜色

改变背景

4.5.3

4ActivityApplicationApp

4(AndroidManifest.xml)

步骤 01

ShoppingCartActivity.java

activity_shopping_cart.xmlShoppingChannelActivity.java

activity_shopping_channel.xmlShoppingDetailActivity

activity_shopping_detail.xmlMainApplication.java

步骤 02

AndroidManifest.xml3

(1)3activity

<activity android:name=".ShoppingCartActivity"

android:theme="@style/AppBaseTheme" /> <activity

android:name=".ShoppingChannelActivity" /> <activity

android:name=".ShoppingDetailActivity" /> (2)applicationname

MainApplication

android:name=".MainApplication"

(3)SD3

<uses-permission

android:name="android.permission.WRITE_EXTERNAL_STORAGE" />

<uses-permission

android:name="android.permission.READ_EXTERNAL_STORAG" /> <uses-

permission

android:name="android.permission.MOUNT_UNMOUNT_FILESYSTEMS" />

步骤 03

resXML

(1)res/layoutactivity_shopping_cart.xmlactivity_shopping_channel.xml

activity_shopping_detail.xml3

(2)res/menumenue_cart.xmlmenu_goods.xml

(3)values/styles.xml

```
<style name="AppBaseTheme" parent="Theme.AppCompat.Light" />
```

步骤 04

MainApplicationShoppingCartActivityShoppingChannel-
ActivityShoppingDetailActivity
ShoppingCartActivity.java

```
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    requestWindowFeature(Window.FEATURE_NO_TITLE);  
    setContentView(R.layout.activity_shopping_cart);  
    iv_menu = (ImageView) findViewById(R.id.iv_menu);  
    tv_title = (TextView) findViewById(R.id.tv_title);  
    tv_count = (TextView) findViewById(R.id.tv_count);  
}
```



```

tv_total_price = (TextView) findViewById(R.id.tv_total_price);
ll_content = (LinearLayout) findViewById(R.id.ll_content);
ll_cart = (LinearLayout) findViewById(R.id.ll_cart);
ll_empty = (LinearLayout) findViewById(R.id.ll_empty);
iv_menu.setOnClickListener(this);
findViewById(R.id.btn_shopping_channel).setOnClickListener(this);
findViewById(R.id.btn_settle).setOnClickListener(this);
iv_menu.setVisibility(View.VISIBLE);
tv_title.setText("购物车");
mCount = Integer.parseInt(SharedUtil.getIntance(this).readShared("count", "0"));
showCount(mCount);
}

//显示购物车图标中的商品数量
private void showCount(int count) {
    mCount = count;
    tv_count.setText(""+mCount);
    if (mCount == 0) {
        ll_content.setVisibility(View.GONE);
        ll_cart.removeAllViews();
        ll_empty.setVisibility(View.VISIBLE);
    } else {
        ll_content.setVisibility(View.VISIBLE);
        ll_empty.setVisibility(View.GONE);
    }
}

@Override
public void onClick(View v) {
    if (v.getId() == R.id.iv_menu) {
        openOptionsMenu();
    } else if (v.getId() == R.id.btn_shopping_channel) {
        Intent intent = new Intent(this, ShoppingChannelActivity.class);
        startActivity(intent);
    } else if (v.getId() == R.id.btn_settle) {
        AlertDialog.Builder builder = new AlertDialog.Builder(this);
        builder.setTitle("结算商品");
        builder.setMessage("客官抱歉，支付功能尚未开通，请下次再来");
        builder.setPositiveButton("我知道了", null);
        builder.create().show();
    }
}
}

```

```

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.menu_cart, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    int id = item.getItemId();
    if (id == R.id.menu_shopping) {
        Intent intent = new Intent(this, ShoppingChannelActivity.class);
        startActivity(intent);
    } else if (id == R.id.menu_clear) {
        //清空购物车数据库
        mCartHelper.deleteAll();
        ll_cart.removeAllViews();
        SharedUtil.getIntance(this).writeShared("count", "0");
        showCount(0);
        mGoodsView.clear();
        mGoodsMap.clear();
        Toast.makeText(this, "购物车已清空", Toast.LENGTH_SHORT).show();
    } else if (id == R.id.menu_return) {
        finish();
    }
    return true;
}

private HashMap<Integer, Long> mGoodsView = new HashMap<Integer, Long>();
private View mContextView;
@Override
public void onCreateContextMenu(ContextMenu menu, View v, ContextMenuInfo menuInfo) {
    mContextView = v;
    getMenuInflater().inflate(R.menu.menu_goods, menu);
}

@Override
public boolean onContextItemSelected(MenuItem item) {
    int id = item.getItemId();
    if (id == R.id.menu_detail) {
        //跳转到查看商品详情页面
        goDetail(mGoodsView.get(mContextView.getId()));
    } else if (id == R.id.menu_delete) {
        //从购物车删除商品的数据库操作

```

```

        long goods_id = mGoodsView.get(mContextView.getId());
        mCartHelper.delete("goods_id="+goods_id);
        ll_cart.removeView(mContextView);
        //更新购物车中的商品数量
        int left_count = mCount-1;
        for (int i=0; i<mCartArray.size(); i++) {
            if (goods_id == mCartArray.get(i).goods_id) {
                left_count = mCount-mCartArray.get(i).count;
                mCartArray.remove(i);
                break;
            }
        }
        SharedUtil.getIntance(this).writeShared("count", ""+left_count);
        showCount(left_count);
        Toast.makeText(this, "已从购物车删除"+mGoodsMap.get(goods_id).name,
Toast.LENGTH_SHORT).show();
        mGoodsMap.remove(goods_id);
        refreshTotalPrice();
    }
    return true;
}

private void goDetail(long rowid) {
    Intent intent = new Intent(this, ShoppingDetailActivity.class);
    intent.putExtra("goods_id", rowid);
    startActivity(intent);
}

private GoodsDBHelper mGoodsHelper;
private CartDBHelper mCartHelper;
private String mFirst = "true";
@Override
protected void onResume() {
    super.onResume();
    mGoodsHelper = GoodsDBHelper.getInstance(this, 1);
    mGoodsHelper.openWriteLink();
    mCartHelper = CartDBHelper.getInstance(this, 1);
    mCartHelper.openWriteLink();
    mFirst = SharedUtil.getIntance(this).readShared("first", "true");
    downloadGoods();
    SharedUtil.getIntance(this).writeShared("first", "false");
    showCart();
}
}

```

```

@Override
protected void onPause() {
    super.onPause();
    mGoodsHelper.closeLink();
    mCartHelper.closeLink();
}

//模拟网络数据，初始化数据库中的商品信息
private void downloadGoods() {
    String path = Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_
DOWNLOADS) + "/";
    if (mFirst.equals("true")) {
        for (int i=0; i<mNameArray.length; i++) {
            GoodsInfo info = new GoodsInfo();
            info.name = mNameArray[i];
            info.desc = mDescArray[i];
            info.price = mPriceArray[i];
            long rowid = mGoodsHelper.insert(info);
            info.rowid = rowid;
            //往全局内存写入商品小图
            Bitmap thumb = BitmapFactory.decodeResource(getResources(), mThumbArray[i]);
            MainApplication.getInstance().mIconMap.put(rowid, thumb);
            String thumb_path = path + rowid + "_s.jpg";
            FileUtil.saveImage(thumb_path, thumb);
            info.thumb_path = thumb_path;
            //往 SD 卡保存商品大图
            Bitmap pic = BitmapFactory.decodeResource(getResources(), mPicArray[i]);
            String pic_path = path + rowid + ".jpg";
            FileUtil.saveImage(pic_path, pic);
            pic.recycle();
            info.pic_path = pic_path;
            mGoodsHelper.update(info);
        }
    } else {
        ArrayList<GoodsInfo> goodsArray = mGoodsHelper.query("1=1");
        for (int i=0; i<goodsArray.size(); i++) {
            GoodsInfo info = goodsArray.get(i);
            Bitmap thumb = BitmapFactory.decodeFile(info.thumb_path);
            MainApplication.getInstance().mIconMap.put(info.rowid, thumb);
        }
    }
}
}

```

4.6

AndroidSharedPreferencesSQLiteSD()AppApplication

“”4

3

(1)SharedPreferencesSQLiteSD4

(2)Application

(3)

第5章 高级控件



AppBroadcast“/”

5.1

AndroidDatePickerDialogTimePickerDialog

5.1.1 DatePicker EditTextinputType="date"EditText

“*****”AndroidDatePickerDatePicker

D



5-1

DatePickerDialogAlertDialogDatePickershow
OnDateSetListeneronDateSetonDateSet100111

```
public class DatePickerActivity extends AppCompatActivity implements OnClickListener,
OnDateSetListener {
    private TextView tv_date;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_date_picker);
        tv_date = (TextView) findViewById(R.id.tv_date);
        findViewById(R.id.btn_date).setOnClickListener(this);
    }

    @Override
    public void onClick(View v) {
        if (v.getId() == R.id.btn_date) {
            Calendar calendar = Calendar.getInstance();
            DatePickerDialog dialog = new DatePickerDialog(this, this,
                calendar.get(Calendar.YEAR), calendar.get(Calendar.MONTH),
                calendar.get(Calendar.DAY_OF_MONTH));
            dialog.show();
        }
    }

    @Override
    public void onDateSet(DatePicker view, int year, int monthOfYear, int dayOfMonth) {
        String desc = String.format("您选择的日期是%d年%d月%d日",
            year, monthOfYear+1, dayOfMonth);
        tv_date.setText(desc);
    }
}
```

5.1.2 TimePicker TimePickerTimePickerDialog DatePickerDialog



5-2

(1)true0~23

(2)OnTimeSetListeneronTimeSet

5-2

```

public class TimePickerActivity extends AppCompatActivity implements OnClickListener,
OnTimeSetListener {
    private TextView tv_time;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_time_picker);
        tv_time = (TextView) findViewById(R.id.tv_time);
        findViewById(R.id.btn_time).setOnClickListener(this);
    }

    @Override
    public void onClick(View v) {
        if (v.getId() == R.id.btn_time) {
            Calendar calendar = Calendar.getInstance();
            TimePickerDialog dialog = new TimePickerDialog(this, this,
                calendar.get(Calendar.HOUR_OF_DAY), calendar.get(Calendar.MINUTE),
true);
            dialog.show();
        }
    }

    @Override
    public void onTimeSet(TimePicker view, int hourOfDay, int minute) {
        String desc = String.format("您选择的时间是%d时%d分", hourOfDay, minute);
        tv_time.setText(desc);
    }
}

```

5.2

BaseAdapterListViewGridView

5.2.1BaseAdapter

3SpinnerArrayAdapterSimpleAdapterArrayAdapter
SimpleAdapterSimpleAdapterAndroidBaseAdapter

BaseAdapter3

getCount

getView

SpinnerBaseAdapter3

步骤 01

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/ll_item"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal" >

    <ImageView
        android:id="@+id/iv_icon"
        android:layout_width="0dp"
        android:layout_height="80dp"
        android:layout_weight="1"
        android:scaleType="fitCenter" />

    <LinearLayout
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="3"
        android:orientation="vertical" >

        <TextView
            android:id="@+id/tv_name"
            android:layout_width="match_parent"
            android:layout_height="0dp"
            android:layout_weight="1"
            android:gravity="left|center"
            android:textColor="@color/black"
            android:textSize="20sp" />

        <TextView
            android:id="@+id/tv_desc"
            android:layout_width="match_parent"
            android:layout_height="0dp"
            android:layout_weight="2"
            android:gravity="left|center"
            android:textColor="@color/black"
            android:textSize="13sp" />

    </LinearLayout>
</LinearLayout>
```

步驟 02 BaseAdapter

```

public class PlanetAdapter extends BaseAdapter {
    private LayoutInflater mInflater;
    private Context mContext;
    private int mLayoutId;
    private ArrayList<Planet> mPlanetList;
    private int mBackground;

    public PlanetAdapter(Context context, int layout_id, ArrayList<Planet> planet_list, int background) {
        mInflater = LayoutInflater.from(context);
        mContext = context;
        mLayoutId = layout_id;
        mPlanetList = planet_list;
        mBackground = background;
    }

    @Override
    public int getCount() {
        return mPlanetList.size();
    }

    @Override
    public Object getItem(int arg0) {
        return mPlanetList.get(arg0);
    }

    @Override
    public long getItemId(int arg0) {
        return arg0;
    }

    @Override
    public View getView(final int position, View convertView, ViewGroup parent) {
        ViewHolder holder = null;
        if (convertView == null) {
            holder = new ViewHolder();
            convertView = mInflater.inflate(mLayoutId, null);
            holder.ll_item = (LinearLayout) convertView.findViewById(R.id.ll_item);
            holder.iv_icon = (ImageView) convertView.findViewById(R.id.iv_icon);
            holder.tv_name = (TextView) convertView.findViewById(R.id.tv_name);
            holder.tv_desc = (TextView) convertView.findViewById(R.id.tv_desc);
            convertView.setTag(holder);
        } else {
            holder = (ViewHolder) convertView.getTag();
        }
    }
}

```

```

    }
    Planet planet = mPlanetList.get(position);
    holder.ll_item.setBackgroundColor(mBackground);
    holder.iv_icon.setImageResource(planet.image);
    holder.tv_name.setText(planet.name);
    holder.tv_desc.setText(planet.desc);
    return convertView;
}

public final class ViewHolder {
    private LinearLayout ll_item;
    public ImageView iv_icon;
    public TextView tv_name;
    public TextView tv_desc;
}
}

```

步骤 03 Spinner

```

private void initSpinner() {
    planetList = Planet.getDefaultList();
    PlanetAdapter adapter = new PlanetAdapter(this, R.layout.item_list, planetList, Color.WHITE);
    Spinner sp = (Spinner) findViewById(R.id.sp_planet);
    sp.setPrompt("请选择行星");
    sp.setAdapter(adapter);
    sp.setSelection(0);
    sp.setOnItemSelectedListener(new MySelectedListener());
}

private class MySelectedListener implements OnItemSelectedListener {
    public void onItemSelected(AdapterView<?> arg0, View arg1, int arg2, long arg3) {
        Toast.makeText(BaseAdapterActivity.this, "您选择的是"+planetList.get(arg2).name,
Toast.LENGTH_LONG).show();
    }

    public void onNothingSelected(AdapterView<?> arg0) {
    }
}
}

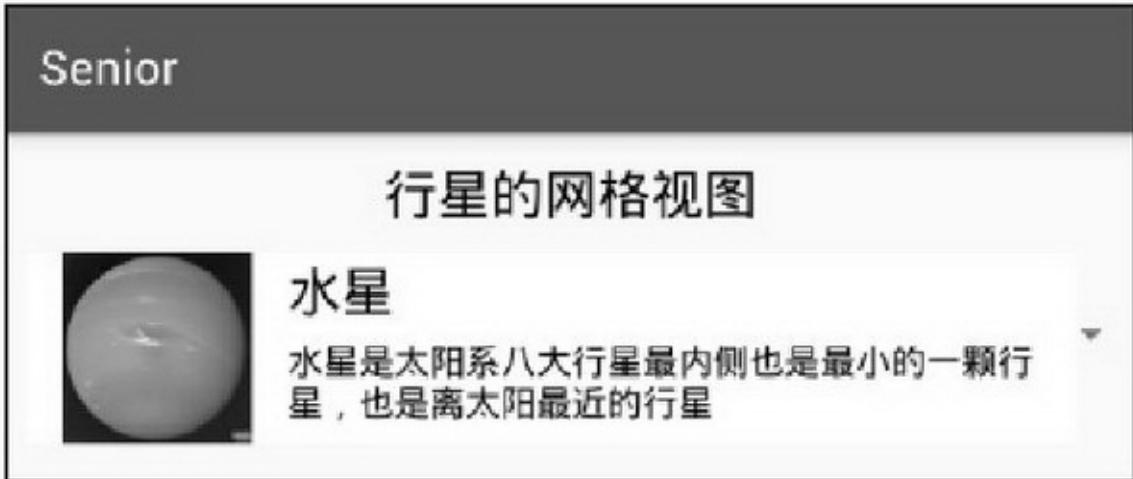
```

5-3item_list.xmlPlanetAdapter

请选择行星	
	<p>水星</p> <p>水星是太阳系八大行星最内侧也是最小的一颗行星，也是离太阳最近的行星</p>
	<p>金星</p> <p>金星是太阳系八大行星之一，排行第二，距离太阳0.725天文单位</p>
	<p>地球</p> <p>地球是太阳系八大行星之一，排行第三，也是太阳系中直径、质量和密度最大的类地行星，距离太阳1.5亿公里</p>
	<p>火星</p> <p>火星是太阳系八大行星之一，排行第四，属于类地行星，直径约为地球的53%</p>
	<p>木星</p> <p>木星是太阳系八大行星中体积最大、自转最快的行星，排行第五。它的质量为太阳的千分之一，但为太阳系中其它七大行星质量总和</p>
	<p>土星</p> <p>土星为太阳系八大行星之一，排行第六，体积仅次于木星</p>

5-3

5.2.2 ListView



5-4

Spinner5-4

ListViewListViewListView5-1

5-1ListView

XML 中的属性	ListView 类的设置方法	说明
divider	setDivider	指定分隔线的图形。如需取消分隔线，可设置该属性值为@null
dividerHeight	setDividerHeight	指定分隔线的高度
headerDividersEnabled	setHeaderDividersEnabled	指定是否显示列表开头的分隔线
footerDividersEnabled	setFooterDividersEnabled	指定是否显示列表末尾的分隔线

ListView3

setAdapterBaseAdapter

setOnItemClickListenerOnItemClickListener

setOnItemLongClickListenerOnItemLongClickListener


```
@Override
public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
    String desc = String.format("您点击了第%d个行星，它的名字是%s", position + 1,
        mPlanetList.get(position).name);
    Toast.makeText(mContext, desc, Toast.LENGTH_LONG).show();
}

@Override
public boolean onItemLongClick(AdapterView<?> parent, View view, int position, long id) {
    String desc = String.format("您长按了第%d个行星，它的名字是%s", position + 1,
        mPlanetList.get(position).name);
    Toast.makeText(mContext, desc, Toast.LENGTH_LONG).show();
    return true;
}
```

ListViewSpinnerListView

```

private class DividerSelectedListener implements OnItemSelectedListener {
    public void onItemSelected(AdapterView<?> arg0, View arg1, int arg2, long arg3) {
        LinearLayout.LayoutParams params = new LinearLayout.LayoutParams(
            LayoutParams.MATCH_PARENT, LayoutParams.WRAP_CONTENT);
        lv_planet.setDivider(drawable);
        lv_planet.setDividerHeight(dividerHeight);
        lv_planet.setPadding(0, 0, 0, 0);
        lv_planet.setBackgroundColor(Color.TRANSPARENT);
        if (arg2 == 0) { // 不显示分隔线(分隔线高度为 0)
            lv_planet.setDividerHeight(0);
        } else if (arg2 == 1) { // 不显示分隔线(分隔线为 null)
            lv_planet.setDivider(null);
            lv_planet.setDividerHeight(dividerHeight);
        } else if (arg2 == 2) { // 只显示内部分隔线(先设置分隔线高度)
            lv_planet.setDividerHeight(dividerHeight);
            lv_planet.setDivider(drawable);
        } else if (arg2 == 3) { // 只显示内部分隔线(后设置分隔线高度)
            lv_planet.setDivider(drawable);
            lv_planet.setDividerHeight(dividerHeight);
        } else if (arg2 == 4) { // 显示底部分隔线(高度是 wrap_content)
            lv_planet.setFooterDividersEnabled(true);
        } else if (arg2 == 5) { // 显示底部分隔线(高度是 match_parent)
            params = new LinearLayout.LayoutParams(LayoutParams.MATCH_PARENT, 0, 1);
            lv_planet.setFooterDividersEnabled(true);
        } else if (arg2 == 6) { // 显示顶部分隔线(别瞎折腾了, 显示不了)
            params = new LinearLayout.LayoutParams(LayoutParams.MATCH_PARENT, 0, 1);
            lv_planet.setFooterDividersEnabled(true);
            lv_planet.setHeaderDividersEnabled(true);
        } else if (arg2 == 7) { // 显示全部分隔线(看我用 padding 大法)
            lv_planet.setPadding(0, dividerHeight, 0, dividerHeight);
            lv_planet.setBackgroundDrawable(drawable);
        }
        lv_planet.setLayoutParams(params);
    }

    public void onNothingSelected(AdapterView<?> arg0) {
    }
}

```

5

(1)setDividerColorDrawabledivider

(2)divider@nulldividerHeight0wrap_content

dividerHeightn-15-5

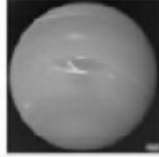
(3)setDividersetDividerHeightsetDividerHeightsetDivider

setDividerHeight5-6

Senior

分隔线显示

不显示分隔线(分隔线为null)



水星

水星是太阳系八大行星最内侧也是最小的一颗行星，也是离太阳最近的行星



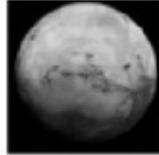
金星

金星是太阳系八大行星之一，排行第二，距离太阳0.725天文单位



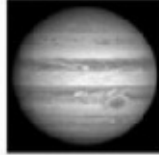
地球

地球是太阳系八大行星之一，排行第三，也是太阳系中直径、质量和密度最大的类地行星，距离太阳1.5亿公里



火星

火星是太阳系八大行星之一，排行第四，属于类地行星，直径约为地球的53%



木星

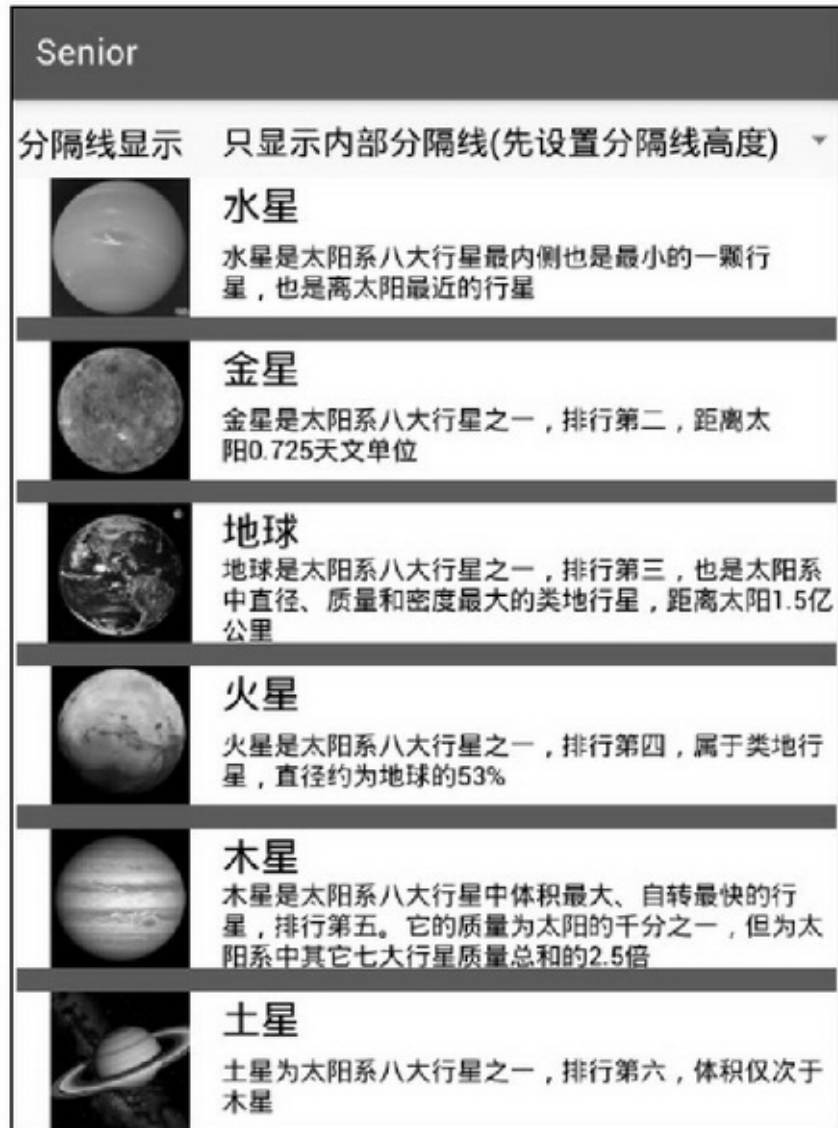
木星是太阳系八大行星中体积最大、自转最快的行星，排行第五。它的质量为太阳的千分之一，但为太阳系中其它七大行星质量总和的2.5倍



土星

土星为太阳系八大行星之一，排行第六，体积仅次于

5-5divider@null



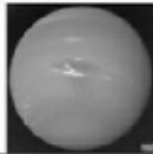
5-6setDividerHeight

(4)ListViewwrap_contentfooterDividersEnabledtrue
setFooterDividersEnabledtruematch_parent5-7

(5)ListViewheaderDividersEnabledsetHeaderDividersEnabled

Senior

分隔线显示 显示底部分隔线(高度是match_parent) ▾



水星

水星是太阳系八大行星最内側也是最小的一颗行星，也是离太阳最近的行星



金星

金星是太阳系八大行星之一，排行第二，距离太阳0.725天文单位



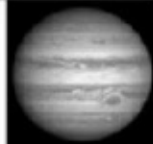
地球

地球是太阳系八大行星之一，排行第三，也是太阳系中直径、质量和密度最大的类地行星，距离太阳1.5亿公里



火星

火星是太阳系八大行星之一，排行第四，属于类地行星，直径约为地球的53%



木星

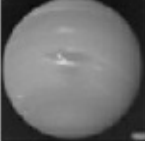



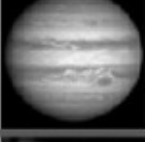

木星是太阳系八大行星中体积最大、自转最快的行星，排行第五。它的质量为太阳的千分之一，但为太阳系中其它七大行星质量总和的2.5倍



土星

土星为太阳系八大行星之一，排行第六，体积仅次于木星

5-7match_parent

Senior	
分隔线显示	显示全部分隔线(看我用padding大法) ▾
	<p>水星</p> <p>水星是太阳系八大行星最内側也是最小的一颗行星，也是离太阳最近的行星</p>
	<p>金星</p> <p>金星是太阳系八大行星之一，排行第二，距离太阳0.725天文单位</p>
	<p>地球</p> <p>地球是太阳系八大行星之一，排行第三，也是太阳系中直径、质量和密度最大的类地行星，距离太阳1.5亿公里</p>
	<p>火星</p> <p>火星是太阳系八大行星之一，排行第四，属于类地行星，直径约为地球的53%</p>
	<p>木星</p> <p>木星是太阳系八大行星中体积最大、自转最快的行星，排行第五。它的质量为太阳的千分之一，但为太阳系中其它七大行星质量总和的2.5倍</p>
	<p>土星</p> <p>土星为太阳系八大行星之一，排行第六，体积仅次于木星</p>

5-8padding

paddingListViewpaddingToppaddingBottompadding

5-8

35AndroidbugSDK

(1)setDividersetDividerHeightsetDividerif“divider = null”

“divider=null & mDividerHeight<=0”

```
public void setDivider(@Nullable Drawable divider) {
    if (divider != null) {
        mDividerHeight = divider.getIntrinsicHeight();
    } else {
        mDividerHeight = 0;
    }
    mDivider = divider;
    mDividerIsOpaque = divider == null || divider.getOpacity() == PixelFormat.OPAQUE;
    requestLayout();
    invalidate();
}
```

(2)ListViewdispatchDraw

```
bounds.top = bottom; bounds.bottom = bottom + dividerHeight;
drawDivider(canvas, bounds, i);
```



5-9

bounds.bottomchild.getTop()bounds.top
child.getTop()-dividerHeight

ListView——LinearLayoutListView
BaseAdapter5-9

ListView3

(1)ListViewListView0dp1ListViewwrap_content

5-9

(2)AppOnItemLongClickListener


```

private View mCurrentView;
@Override
public boolean onItemLongClick(AdapterView<?> parent, View view, int position, long id) {
    mCurrentGood = mCartArray.get(position);
    mCurrentView = view;
    mHandler.postDelayed(mPopupMenu, 100);
    return true;
}

private Handler mHandler = new Handler();
private Runnable mPopupMenu = new Runnable() {
    @Override
    public void run() {
        lv_cart.setOnItemLongClickListener(null);
        registerForContextMenu(mCurrentView);
        openContextMenu(mCurrentView);
        unregisterForContextMenu(mCurrentView);
        lv_cart.setOnItemLongClickListener(ShoppingCartActivity.this);
    }
};

```

(3) EditTextButton(ImageButtonCheckBox)

OnItemClickListener EditText EditText EditTextButton

EditTextButton(EditTextButton) descendantFocusability

android:descendantFocusability="blocksDescendants"

5.2.3 GridView

GridView ListView GridView5-2

5-2 GridView

XML 中的属性	GridView 类的设置方法	说明
horizontalSpacing	setHorizontalSpacing	指定网格项在水平方向的间距
verticalSpacing	setVerticalSpacing	指定网格项在垂直方向的间距
numColumns	setNumColumns	指定列的数目
stretchMode	setStretchMode	指定剩余空间的拉伸模式。拉伸模式的取值说明见表 5-3
columnWidth	setColumnWidth	指定每列的宽度。拉伸模式为 spacingWidth、spacingWidthUniform 时，必须指定列宽

XML 中的拉伸模式	GridView 类的拉伸模式	说明
none	NO_STRETCH	不拉伸
columnWidth	STRETCH_COLUMN_WIDTH	若有剩余空间，则拉伸列宽挤掉空隙
spacingWidth	STRETCH_SPACING	若有剩余空间，则列宽不变，把空间分配到每列间的空隙
spacingWidthUniform	STRETCH_SPACING_UNIFORM	若有剩余空间，则列宽不变，把空间分配到每列左右的空隙

5-3

GridView3

setAdapterBaseAdapter

setOnItemClickListenerListView

setOnItemLongClickListenerListView

GridViewGridViewGridView()

```

private class DividerSelectedListener implements OnItemSelectedListener {
    public void onItemSelected(AdapterView<?> arg0, View arg1, int arg2, long arg3) {
        gv_planet.setBackgroundColor(Color.RED);
        gv_planet.setHorizontalSpacing(dividerPad);
        gv_planet.setVerticalSpacing(dividerPad);
        gv_planet.setStretchMode(GridView.STRETCH_COLUMN_WIDTH);
        gv_planet.setColumnWidth(250);
        gv_planet.setPadding(0, 0, 0, 0);
        if (arg2 == 0) { // 不显示分隔线
            gv_planet.setBackgroundColor(Color.WHITE);
            gv_planet.setHorizontalSpacing(0);
            gv_planet.setVerticalSpacing(0);
        } else if (arg2 == 1) { // 只显示内部分隔线(NO_STRETCH)
            gv_planet.setStretchMode(GridView.NO_STRETCH);
        } else if (arg2 == 2) { // 只显示内部分隔线(COLUMN_WIDTH)
            gv_planet.setStretchMode(GridView.STRETCH_COLUMN_WIDTH);
        } else if (arg2 == 3) { // 只显示内部分隔线(STRETCH_SPACING)
            gv_planet.setStretchMode(GridView.STRETCH_SPACING);
        } else if (arg2 == 4) { // 只显示内部分隔线(SPACING_UNIFORM)
            gv_planet.setStretchMode(GridView.STRETCH_SPACING_UNIFORM);
        } else if (arg2 == 5) { // 显示全部分隔线 (使用 padding)
            gv_planet.setPadding(dividerPad, dividerPad, dividerPad, dividerPad);
        }
    }
    public void onNothingSelected(AdapterView<?> arg0) {
    }
}

```

5-105-11

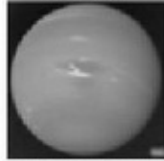
5-11columnWidth5-12nonespacingWidth

horizontalSpacing5-13

分隔线显示

不显示分隔线

水星



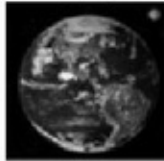
水星是太阳系八大行星最内侧也是最小的一颗行星，也是离太阳最近的行星

金星



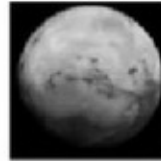
金星是太阳系八大行星之一，排行第二，距离太阳0.725天文单位的行星

地球



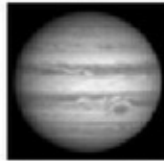
地球是太阳系八大行星之一，排行第三，也是太阳系中直径、质量和密度最大的类地行星，距离太阳1.5亿公里

火星



火星是太阳系八大行星之一，排行第四，属于类地行星，直径约为地球的53%

木星



木星是太阳系八大行星中体积最大、自转最快的行星，排行第五。它的质量为太阳的千分之一，但为太阳系中其它七大行星质

土星

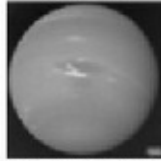


土星为太阳系八大行星之一，排行第六，体积仅次于木星

Senior

分隔线显示 只显示内部分隔线(COLUMN_WIDTH) ▾

水星



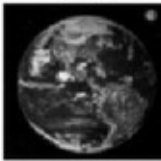
水星是太阳系八大行星最内侧也是最小的一颗行星，也是离太阳最近的行星

金星



金星是太阳系八大行星之一，排行第二，距离太阳0.725天文单位

地球



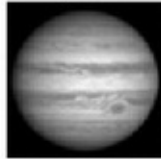
地球是太阳系八大行星之一，排行第三，也是太阳系中直径、质量和密度最大的类地行星，距离太阳1.5亿公里

火星



火星是太阳系八大行星之一，排行第四，属于类地行星，直径约为地球的53%

木星



木星是太阳系八大行星中体积最大、自转最快的行星，排行第五。它的质量为太阳的千分之一，但为太阳系中其它七大行星质

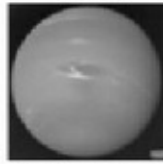
土星



土星为太阳系八大行星之一，排行第六，体积仅次于木星

分隔线显示 只显示内部分隔线(NO_STRETCH)

水星



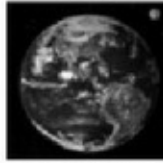
水星是太阳系八大行星最内侧也是最小的一颗行星，也是离太阳最近的行星

金星



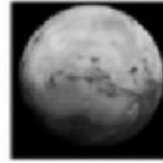
金星是太阳系八大行星之一，排行第二，距离太阳0.725天文单位

地球



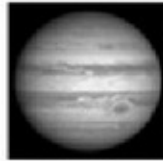
地球是太阳系八大行星之一，排行第三，也是太阳系中直径、质量和密度最大的类地行星，距离太阳1.5亿公里

火星



火星是太阳系八大行星之一，排行第四，属于类地行星，直径约为地球的53%

木星



木星是太阳系八大行星中体积最大、自转最快的行星，排行第五。它的质量为太阳的千分之一，但为太阳系中其它七大

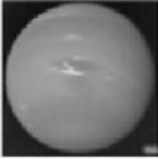



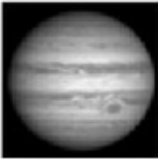

土星



土星为太阳系八大行星之一，排行第六，体积仅次于木星

Senior

分隔线显示 只显示内部分隔线(STRETCH_SPACING) ▾

<p>水星</p>  <p>水星是太阳系八大行星最内侧也是最小的一颗行星，也是离太阳最近的行星</p>	<p>金星</p>  <p>金星是太阳系八大行星之一，排行第二，距离太阳0.725天文单位</p>
<p>地球</p>  <p>地球是太阳系八大行星之一，排行第三，也是太阳系中直径、质量和密度最大的类地行星，距离太阳1.5亿公里</p>	<p>火星</p>  <p>火星是太阳系八大行星之一，排行第四，属于类地行星，直径约为地球的53%</p>
<p>木星</p>  <p>木星是太阳系八大行星中体积最大、自转最快的行星，排行第五。它的质量为太阳的千分之一，但为太阳系中其它七大</p>	<p>土星</p>  <p>土星为太阳系八大行星之一，排行第六，体积仅次于木星</p>

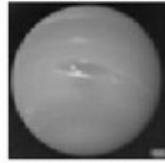
5-13spacingWidth

spacingWidthUniform5-14columnWidth

GridViewListViewhorizontalSpacingverticalSpacingpadding
padding5-15padding

分隔线显示 只显示内部分隔线(SPACING_UNIFORM) ▾

水星



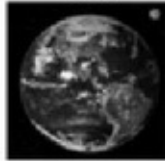
水星是太阳系八大行星最内侧也是最小的一颗行星，也是离太阳最近的行星

金星



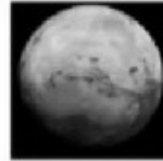
金星是太阳系八大行星之一，排行第二，距离太阳0.725天文单位

地球



地球是太阳系八大行星之一，排行第三，也是太阳系中直径、质量和密度最大的类地行星，距离太阳1.5亿公里

火星



火星是太阳系八大行星之一，排行第四，属于类地行星，直径约为地球的53%

木星



木星是太阳系八大行星中体积最大、自转最快的行星，排行第五。它的质量为太阳的千分之一，但为太阳系中其它七大

土星

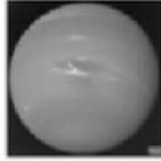


土星为太阳系八大行星之一，排行第六，体积仅次于木星

Senior

分隔线显示 显示全部分隔线(看我用padding大法) ▾

水星



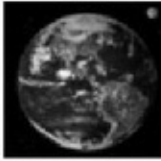
水星是太阳系八大行星最内侧也是最小的一颗行星，也是离太阳最近的行星

金星



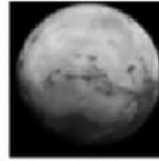
金星是太阳系八大行星之一，排行第二，距离太阳0.725天文单位

地球



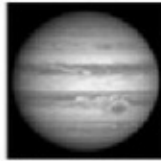
地球是太阳系八大行星之一，排行第三，也是太阳系中直径、质量和密度最大的类地行星，距离太阳1.5亿公里

火星



火星是太阳系八大行星之一，排行第四，属于类地行星，直径约为地球的53%

木星



木星是太阳系八大行星中体积最大、自转最快的行星，排行第五。它的质量为太阳的千分之一，但为太阳系中其它七大行星质

土星



土星为太阳系八大行星之一，排行第六，体积仅次于木星

5-15padding



5-16

GridView GridView5-16

(1)“”()ListView

android:descendantFocusability="blocksDescendants"

(2)“”Activity

步骤 01 addCartListener

```
public GoodsAdapter(Context context, ArrayList<GoodsInfo> goods_list, addCartListener listener) {
    mInflater = LayoutInflater.from(context);
    mContext = context;
    mGoodsArray = goods_list;
    mAddCartListener = listener;
}

private addCartListener mAddCartListener;
public static interface addCartListener {
    public void addToCart(long goods_id);
}
```

步骤 02 “”addToCart

mAddCartListener.addToCart(info.rowid);

步骤 03 ActivityaddCartListeneraddToCart

5.3

ViewPagerPagerAdapterPagerTitleStrip/PagerTabStrip
ViewPager

5.3.1 ViewPager ListViewGridView

AndroidViewPagerViewPager(ListView)ViewPager
ViewPagerListViewGridViewListViewGridViewBaseAdapter,
ViewPagerPagerAdapterListViewGridViewOnItemClickListener, ViewPager
OnPageChangeListenerViewPager
setAdapterPagerAdapter
setCurrentItem
▶ onPageScrollStateChanged
▶ onPageScrolled
▶ onPageSelected
addOnPageChangeListenerOnPageChangeListener3
PagerAdapterBaseAdaptergetItemInstantiateItem
destroyItemPagerAdapter

```

public class ImagePagerAdapater extends PagerAdapter {
    private Context mContext;
    private ArrayList<ImageView> mViewList = new ArrayList<ImageView>();
    private ArrayList<GoodsInfo> mGoodsList = new ArrayList<GoodsInfo>();

    public ImagePagerAdapater(Context context, ArrayList<GoodsInfo> goodsList) {
        mContext = context;
        mGoodsList = goodsList;
        for (int i=0; i<mGoodsList.size(); i++) {
            ImageView view = new ImageView(mContext);
            view.setLayoutParams(new LayoutParams(
                LayoutParams.MATCH_PARENT, LayoutParams.WRAP_CONTENT));
            view.setImageResource(mGoodsList.get(i).pic);
            view.setScaleType(ScaleType.FIT_CENTER);
            mViewList.add(view);
        }
    }

    @Override
    public int getCount() {
        return mViewList.size();
    }

    @Override
    public boolean isViewFromObject(View arg0, Object arg1) {
        return arg0 == arg1;
    }

    @Override
    public void destroyItem(ViewGroup container, int position, Object object) {
        container.removeView(mViewList.get(position));
    }

    @Override
    public Object instantiateItem(ViewGroup container, int position) {
        container.addView(mViewList.get(position));
        return mViewList.get(position);
    }
}

```

ImagePagerAdapater

```

public class ViewPagerActivity extends AppCompatActivity implements OnPageChangeListener {
    private ArrayList<GoodsInfo> goodsList;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_view_pager);
        ViewPager vp_content = (ViewPager) findViewById(R.id.vp_content);
        goodsList = GoodsInfo.getDefaultList();
        ImagePagerAdapater adapter = new ImagePagerAdapater(this, goodsList);
        vp_content.setAdapter(adapter);
        vp_content.setCurrentItem(0);
        vp_content.addOnPageChangeListener(this);
    }

    //翻页状态改变时调用，状态参数取值说明为：0 表示静止，1 表示正在滑动，2 表示滑动完毕
    //在翻页过程中，状态值变化依次为：正在滑动→滑动完毕→静止
    @Override
    public void onPageScrollStateChanged(int arg0) {
    }

    //在翻页过程中调用。该方法的三个参数取值说明为：第一个参数表示当前页面的序号
    //第二个参数表示当前页面偏移的百分比，取值为 0 到 1；第三个参数表示当前页面的偏移距离
    @Override
    public void onPageScrolled(int arg0, float arg1, int arg2) {
    }

    @Override
    public void onPageSelected(int arg0) {
        Toast.makeText(this, "您翻到的手机品牌是： "+goodsList.get(arg0).name,
        Toast.LENGTH_SHORT).show();
    }
}

```

ViewPagerv4android.support.v4.view.ViewPager

```

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="10dp" >
    <android.support.v4.view.ViewPager
        android:id="@+id/vp_content"
        android:layout_width="match_parent"
        android:layout_height="400dp" />
</LinearLayout>

```

5-17ViewPager



5-17

5.3.2 PagerTitleStrip/PagerTabStrip ViewPagerAndroid

PagerTitleStripPagerTabStripViewPagerPagerTitleStrip

```
P: @Override  
    public CharSequence getPageTitle(int position) {  
        return mGoodsList.get(position).name;  
    }  
}
```

PagerTitleStripPagerTitleStripv4

android.support.v4.view.PagerTitleStrip PagerTabStrip

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="10dp" >

    <android.support.v4.view.ViewPager
        android:id="@+id/vp_content"
        android:layout_width="match_parent"
        android:layout_height="400dp" >

        <android.support.v4.view.PagerTitleStrip
            android:id="@+id/pts_title"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />
    </android.support.v4.view.ViewPager>
</LinearLayout>
```

PagerTitleStrip5-18 PagerTabStrip5-19

Senior

iphone7

Mate8

小米5



5-18PagerTitleStrip



5-19PagerTabStrip

```
PagerTabStrip pts_tab = (PagerTabStrip) findViewById(R.id.pts_tab);  
pts_tab.setTextSize(TypedValue.COMPLEX_UNIT_SP, 20);  
pts_tab.setTextColor(Color.GREEN);
```

5.3.3

ViewPagerAppAppViewPager





5-21

3(ImageView)(RadioGroup)(Button)

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent" >

    <ImageView
        android:id="@+id/iv_launch"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:scaleType="fitXY" />

    <RadioGroup
        android:id="@+id/rg_indicate"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true"
        android:layout_gravity="bottom|center"
        android:orientation="horizontal"
        android:paddingBottom="20dp" />

    <Button
        android:id="@+id/btn_start"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginLeft="80dp"
        android:layout_marginRight="80dp"
        android:layout_centerInParent="true"
        android:gravity="center"
        android:text="立即开始美好生活"
        android:textColor="#ff3300"
        android:textSize="22sp"
        android:visibility="gone" />

</RelativeLayout>
```

5-225-235-22335-23



Senior

我的，就是我的



立即开始美好生活



```

public class LaunchSimpleAdapter extends PagerAdapter {
    private LayoutInflater mInflater;
    private Context mContext;
    private ArrayList<View> mViewList = new ArrayList<View>();

    public LaunchSimpleAdapter(Context context, int[] imageArray) {
        mInflater = LayoutInflater.from(context);
        mContext = context;
        for (int i=0; i<imageArray.length; i++) {
            View view = mInflater.inflate(R.layout.item_launch, null);
            ImageView iv_launch = (ImageView) view.findViewById(R.id.iv_launch);
            RadioGroup rg_indicate = (RadioGroup) view.findViewById(R.id.rg_indicate);
            Button btn_start = (Button) view.findViewById(R.id.btn_start);
            iv_launch.setImageResource(imageArray[i]);
            for (int j=0; j<imageArray.length; j++) {
                RadioButton radio = new RadioButton(mContext);
                radio.setLayoutParams(new LayoutParams(LayoutParams.WRAP_CONTENT,
LayoutParams.WRAP_CONTENT));
                radio.setButtonDrawable(R.drawable.launch_guide);
                radio.setPadding(10, 10, 10, 10);
                rg_indicate.addView(radio);
            }
            ((RadioButton)rg_indicate.getChildAt(i)).setChecked(true);
            if (i == imageArray.length-1) {
                btn_start.setVisibility(View.VISIBLE);
                btn_start.setOnClickListener(new OnClickListener() {
                    @Override
                    public void onClick(View v) {
                        Toast.makeText(mContext, "欢迎您开启美好生活",
Toast.LENGTH_SHORT).show();
                    }
                });
            }
            mViewList.add(view);
        }
    }

    @Override
    public int getCount() {
        return mViewList.size();
    }
}

```



```
}

@Override
public boolean isViewFromObject(View arg0, Object arg1) {
    return arg0 == arg1;
}

@Override
public void destroyItem(ViewGroup container, int position, Object object) {
    container.removeView(mViewList.get(position));
}

@Override
public Object instantiateItem(ViewGroup container, int position) {
    container.addView(mViewList.get(position));
    return mViewList.get(position);
}
}
```

5.4 Fragment Fragment Fragment State Pager Adapter Fragment Fragment Fragment

5.4.1

Fragment Fragment fragment

(Logo)

Fragment

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:background="#bbffbb" >

    <TextView
        android:id="@+id/tv_adv"
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:gravity="center"
        android:text="广告"
        android:textColor="#000000"
        android:textSize="17sp" />

    <ImageView
        android:id="@+id/iv_adv"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="5"
        android:src="@drawable/adv"
        android:scaleType="fitCenter" />

</LinearLayout>
```

Fragment Fragment

```

public class StaticFragment extends Fragment implements OnClickListener {
    protected View mView;
    protected Context mContext;
    private TextView tv_adv;
    private ImageView iv_adv;

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
        mContext = getActivity();
        mView = inflater.inflate(R.layout.fragment_static, container, false);
        tv_adv = (TextView) mView.findViewById(R.id.tv_adv);
        iv_adv = (ImageView) mView.findViewById(R.id.iv_adv);
        tv_adv.setOnClickListener(this);
        iv_adv.setOnClickListener(this);
        return mView;
    }

    @Override
    public void onClick(View v) {
        if (v.getId() == R.id.tv_adv) {
            Toast.makeText(mContext, "您点击了广告文本", Toast.LENGTH_LONG).show();
        } else if (v.getId() == R.id.iv_adv) {
            Toast.makeText(mContext, "您点击了广告图片", Toast.LENGTH_LONG).show();
        }
    }
}

```

FragmentfragmentfragmentnameFragment

```

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="5dp" >

    <fragment
        android:id="@+id/fragment_static"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:name="com.example.senior.fragment.StaticFragment" />

    <TextView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:gravity="center|top"
        android:text="这里是每个页面的具体内容"
        android:textColor="#000000"
        android:textSize="17sp" />

</LinearLayout>

```

5-24Fragment



5-24Fragment

(1)fragmentidAppMust specify unique android:id, android:tag, or have a parent with an id for ***

(2)Activity, Fragmentandroid.app.Fragment

android.support.v4.app.FragmentAppTrying to instantiate a class *** that is
not a Fragmentjava.lang.ClassCastException*** cannot be cast to
android.app.FragmentAppCompatActivityFragmentActivity
android.app.Fragmentandroid.support.v4.app.Fragment
 FragmentManagerCreateonStartonResumeonPause
onStoponDestroyFragmentManager5
 onAttachActivityActivityFragmentManagerActivityActivity
set***Listener
 onCreateView
 onActivityCreated
 onDestroyView
 onDetachActivity
 FragmentManagerCreateActivityonStartonResumeActivity
12:26:11.506D/StaticFragment(5809)onAttach
12:26:11.506D/StaticFragment(5809)onCreate
12:26:11.530D/StaticFragment(5809)onCreateView
12:26:11.530D/FragmentManagerStaticActivity(5809)onCreate 12:26:11.530
D/StaticFragment(5809)onActivityCreated 12:26:11.530
D/FragmentManagerStaticActivity(5809)onStart 12:26:11.530
D/StaticFragment(5809)onStart
 12:26:11.530D/FragmentManagerStaticActivity(5809)onResume 12:26:11.530
D/StaticFragment(5809)onResume
 FragmentManagerPauseonStoponDestroyActivity
12:26:36.586D/StaticFragment(5809)onPause
12:26:36.586D/FragmentManagerStaticActivity(5809)onPause 12:26:36.990
D/StaticFragment(5809)onStop
 12:26:36.990D/FragmentManagerStaticActivity(5809)onStop 12:26:36.990
D/StaticFragment(5809)onDestroyView 12:26:36.990
D/StaticFragment(5809)onDestroy
 12:26:36.990D/StaticFragment(5809)onDetach

12:26:36.990D/FragmentStaticActivity(5809)onDestroy

5.4.2/FragmentStatePagerAdapter Fragment

FragmentFragmentViewPagerFragment

ViewPagerFragmentViewPagerPagerAdapterFragment

FragmentStatePagerAdapterFragmentStatePagerAdaptergetItem

```
public class MobilePagerAdapter extends FragmentStatePagerAdapter {
    private ArrayList<GoodsInfo> mGoodsList = new ArrayList<GoodsInfo>();
    public MobilePagerAdapter(FragmentManager fm, ArrayList<GoodsInfo> goodsList) {
        super(fm);
        mGoodsList = goodsList;
    }

    public int getCount() {
        return mGoodsList.size();
    }

    public Fragment getItem(int position) {
        return DynamicFragment.newInstance(position,
            mGoodsList.get(position).pic, mGoodsList.get(position).desc);
    }

    @Override
    public CharSequence getPageTitle(int position) {
        return mGoodsList.get(position).name;
    }
}
```

newInstanceFragmentsetArgumentsFragmentonCreateView
getArguments

```

public class DynamicFragment extends Fragment {
    protected View mView;
    protected Context mContext;
    private int mPosition;
    private int mImageId;
    private String mDesc;

    public static DynamicFragment newInstance(int position, int image_id, String desc) {
        DynamicFragment fragment = new DynamicFragment();
        Bundle bundle = new Bundle();
        bundle.putInt("position", position);
        bundle.putInt("image_id", image_id);
        bundle.putString("desc", desc);
        fragment.setArguments(bundle);
        return fragment;
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
        mContext = getActivity();
        if (getArguments() != null) {
            mPosition = getArguments().getInt("position", 0);
            mImageId = getArguments().getInt("image_id", 0);
            mDesc = getArguments().getString("desc");
        }
        mView = inflater.inflate(R.layout.fragment_dynamic, container, false);
        ImageView iv_pic = (ImageView) mView.findViewById(R.id.iv_pic);
        TextView tv_desc = (TextView) mView.findViewById(R.id.tv_desc);
        iv_pic.setImageResource(mImageId);
        tv_desc.setText(mDesc);
        return mView;
    }
}

```

ActivityAppCompatActivityFragmentActivity

```

public class FragmentDynamicActivity extends FragmentActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_fragment_dynamic);
        PagerTabStrip pts_tab = (PagerTabStrip) findViewById(R.id.pts_tab);
        pts_tab.setTextSize(TypedValue.COMPLEX_UNIT_SP, 20);
        ViewPager vp_content = (ViewPager) findViewById(R.id.vp_content);
        ArrayList<GoodsInfo> goodsList = GoodsInfo.getDefaultList();
        MobilePagerAdapter adapter = new MobilePagerAdapter(getSupportFragmentManager(),
goodsList);
        vp_content.setAdapter(adapter);
        vp_content.setCurrentItem(0);
    }
}

```

5-25FragmentViewPager



5-25Fragment

Fragment

12:28:28.074D/FragmentDynamicActivity(5809)onCreate

12:28:28.074D/FragmentDynamicActivity(5809)onStart 12:28:28.074

D/FragmentDynamicActivity(5809)onResume 12:28:28.086

D/DynamicFragment(5809)onAttach position=0

12:28:28.086D/DynamicFragment(5809)onCreate position=0

12:28:28.114D/DynamicFragment(5809)onCreateView position=0

12:28:28.114D/DynamicFragment(5809)onActivityCreated position=0

12:28:28.114D/DynamicFragment(5809)onStart position=0
12:28:28.114D/DynamicFragment(5809)onResume position=0
12:28:28.114D/DynamicFragment(5809)onAttach position=0
12:28:28.114D/DynamicFragment(5809)onCreate position=0
12:28:28.146D/DynamicFragment(5809)onCreateView position=1
12:28:28.146D/DynamicFragment(5809)onStart position=1
12:28:28.146D/DynamicFragment(5809)onResume position=1

12:28:57.994D/DynamicFragment(5809)onPause position=0
12:28:57.994D/DynamicFragment(5809)onPause position=1
12:28:57.994D/FragmentDynamicActivity(5809)onPause

12:28:58.402D/DynamicFragment(5809)onStop position=0
12:28:58.402D/DynamicFragment(5809)onStop position=1
12:28:58.402D/FragmentDynamicActivity(5809)onStop 12:28:58.402
D/DynamicFragment(5809)onDestroyView position=0
12:28:58.402D/DynamicFragment(5809)onDestroy position=0
12:28:58.402D/DynamicFragment(5809)onDetach position=0
12:28:58.402D/DynamicFragment(5809)onDestroyView position=1
12:28:58.402D/DynamicFragment(5809)onDestroy position=1
12:28:58.402D/DynamicFragment(5809)onDetach position=1
12:28:58.402D/FragmentDynamicActivity(5809)onDestroy

(1)FragmentonCreateActivity

(2)onActivityCreatedActivityonCreate

(3)FragmentFragmentFragment

FragmentFragmentViewPager

5.4.3

Fragment“5.3.3”

```
public class LaunchImproveAdapter extends FragmentStatePagerAdapter {
    private ArrayList<Integer> mImageList = new ArrayList<Integer>();
    public LaunchImproveAdapter(FragmentManager fm, int[] imageArray) {
        super(fm);
        for (int i=0; i<imageArray.length; i++) {
            mImageList.add(imageArray[i]);
        }
    }

    public int getCount() {
        return mImageList.size();
    }

    public Fragment getItem(int position) {
        return LaunchFragment.newInstance(position, mImageList.get(position));
    }
}
```

Fragment

```

public class LaunchFragment extends Fragment {
    protected View mView;
    protected Context mContext;
    private int mPosition;
    private int mImageId;
    private int mCount = 4;

    public static LaunchFragment newInstance(int position, int image_id) {
        LaunchFragment fragment = new LaunchFragment();
        Bundle bundle = new Bundle();
        bundle.putInt("position", position);
        bundle.putInt("image_id", image_id);
        fragment.setArguments(bundle);
        return fragment;
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
        mContext = getActivity();
        if (getArguments() != null) {
            mPosition = getArguments().getInt("position", 0);
            mImageId = getArguments().getInt("image_id", 0);
        }
        mView = inflater.inflate(R.layout.item_launch, container, false);
        ImageView iv_launch = (ImageView) mView.findViewById(R.id.iv_launch);
        RadioGroup rg_indicate = (RadioGroup) mView.findViewById(R.id.rg_indicate);
        Button btn_start = (Button) mView.findViewById(R.id.btn_start);
        iv_launch.setImageResource(mImageId);
        for (int j=0; j<mCount; j++) {
            RadioButton radio = new RadioButton(mContext);
            radio.setLayoutParams(new LayoutParams(LayoutParams.WRAP_CONTENT,
LayoutParams.WRAP_CONTENT));
            radio.setButtonDrawable(R.drawable.launch_guide);
            radio.setPadding(10, 10, 10, 10);
            rg_indicate.addView(radio);
        }
        ((RadioButton)rg_indicate.getChildAt(mPosition)).setChecked(true);
        if (mPosition == mCount-1) {
            btn_start.setVisibility(View.VISIBLE);
            btn_start.setOnClickListener(new OnClickListener() {
                @Override
                public void onClick(View v) {
                    Toast.makeText(mContext, "欢迎开启美好生活",

```

```

Toast.LENGTH_SHORT).show();
                }
            });
        }
        return mView;
    }
}

```



5-26Fragment

5.5 Broadcast

BroadcastReceiver AndroidManifest.xml

5.5.1/

Intent "5.2.3 GridView" (setArguments)

(1) Fragmenton Attachon AttachActivity Activity

(2) Android "5.2" ListView GridView padding

BroadcastReceiver

(Broadcast) AndroidActivity

(1) ActivityBroadcast

(2)

(3)

3

sendBroadcast

registerReceiveronStartonResume

unregisterReceiveronStoponPause

LocalBroadcastManager registerReceiver unregisterReceiver(

)()

Fragment Spinner Fragment

```

public final static String EVENT = "com.example.senior.fragment.BroadcastFragment";
private String[] mColorNameArray = {"红色", "黄色", "绿色", "青色", "蓝色"};
private int[] mColorIdArray= {Color.RED, Color.YELLOW, Color.GREEN, Color.CYAN,
Color.BLUE};
class ColorSelectedListener implements OnItemSelectedListener {
    public void onItemSelected(AdapterView<?> arg0, View arg1, int arg2, long arg3) {
        Intent intent = new Intent(BroadcastFragment.EVENT);
        intent.putExtra("color", mColorIdArray[arg2]);
        LocalBroadcastManager.getInstance(mContext).sendBroadcast(intent);
    }

    public void onNothingSelected(AdapterView<?> arg0) {
    }
}
}

```

Activity

```

@Override
public void onStart() {
    super.onStart();
    bgChangeReceiver = new BgChangeReceiver();
    IntentFilter filter = new IntentFilter(BroadcastFragment.EVENT);
    LocalBroadcastManager.getInstance(this).registerReceiver(bgChangeReceiver, filter);
}

@Override
public void onStop() {
    LocalBroadcastManager.getInstance(this).unregisterReceiver(bgChangeReceiver);
    super.onStop();
}

private BgChangeReceiver bgChangeReceiver;
private class BgChangeReceiver extends BroadcastReceiver {
    @Override
    public void onReceive(Context context, Intent intent) {
        if (intent != null) {
            int color = intent.getIntExtra("color", Color.WHITE);
            ll_brd_temp.setBackgroundColor(color);
        }
    }
}
}

```

5-27Fragment



5-27FragmentActivity

5.5.2AlarmManager AlarmManagerAndroidApp

AppAppAppBroadcast

AlarmManager


```

@Override
public void onClick(View v) {
    if (v.getId() == R.id.btn_alarm) {
        Intent intent = new Intent(ALARM_EVENT);
        PendingIntent pIntent = PendingIntent.getBroadcast(this, 0, intent,
            PendingIntent.FLAG_UPDATE_CURRENT);
        AlarmManager alarmMgr = (AlarmManager) getSystemService(ALARM_SERVICE);
        Calendar calendar = Calendar.getInstance();
        calendar.setTimeInMillis(System.currentTimeMillis());
        calendar.add(Calendar.SECOND, mDelay);
        alarmMgr.set(AlarmManager.RTC_WAKEUP, calendar.getTimeInMillis(), pIntent);
        mDesc = DateUtil.getNowTime() + " 设置闹钟";
        tv_alarm.setText(mDesc);
    }
}

```

AlarmReceiver

```

private String ALARM_EVENT = "com.example.senior.AlarmActivity.AlarmReceiver";
private static String mDesc = "";
private static boolean bArrived = false;
public static class AlarmReceiver extends BroadcastReceiver {
    @Override
    public void onReceive(Context context, Intent intent) {
        if (intent != null) {
            Log.d(TAG, "AlarmReceiver onReceive");
            if (tv_alarm != null && bArrived == false) {
                bArrived = true;
                mDesc = String.format("%s\n%s 闹钟时间到达", mDesc,
DateUtil.getNowTime());
                tv_alarm.setText(mDesc);
            }
        }
    }
}

```

AndroidManifest.xmlapplication(AndroidManifest.xml)

```

<receiver android:name=".AlarmActivity$AlarmReceiver" >
    <intent-filter>
        <action android:name="com.example.senior.AlarmActivity.AlarmReceiver" />
    </intent-filter>
</receiver>

```

5-285-295-285-29



5-28



5-29

AlarmManagerPendingIntent
PendingIntentPendingIntentActivityBroadcastIntent
PendingIntent, PendingIntentgetBroadcast
AlarmManagerset(AlarmManager.RTC_WAKEUP)
(getBroadcastPendingIntent)

5.6/

“/”

5.6.1

12125-302016105-31201612

2016年 日历						
九月	十月					十一月
周一	周二	周三	周四	周五	周六	周日
26 廿六	27 廿七	28 廿八	29 廿九	30 卅十	1 国庆节	2 初二
3 初三	4 初四	5 初五	6 初六	7 初七	8 寒露	9 重阳节
10 双十节	11 十一	12 十二	13 十三	14 十四	15 十五	16 十六
17 十七	18 十八	19 十九	20 廿十	21 廿一	22 廿二	23 廿三
24 霜降	25 廿五	26 廿六	27 廿七	28 廿八	29 廿九	30 卅十
31 万圣节	1 初二	2 初三	3 初四	4 初五	5 初六	6 初七

2016年 日历						
十一月		十二月				
周一	周二	周三	周四	周五	周六	周日
28 廿九	29 十一月	30 初二	1 初三	2 初四	3 初五	4 初六
5 初七	6 初八	7 大雪	8 初十	9 十一	10 十二	11 十三
12 十四	13 十五	14 十六	15 十七	16 十八	17 十九	18 廿十
19 廿一	20 廿二	21 廿三	22 冬至	23 廿五	24 平安夜	25 圣诞节
26 廿八	27 廿九	28 卅十	29 十二月	30 初二	31 初三	1 元旦
2 初五	3 初六	4 初七	5 腊八节	6 小寒	7 初十	8 十一

5-31201612

6

(1)GridViewGridView7

(2)BaseAdapter

(3)ViewPager12ViewPager

(4)Fragment1212Fragment

(5)FragmentManager12FragmentViewPager

(6)PagerTabStripPagerTabStrip

5-32()

2016年 日程安排		
第三十九周	第四十周	第四十一周
星期一	9月26日 农历八月廿六 今日暂无日程安排	
星期二	9月27日 农历八月廿七 今日暂无日程安排	
星期三	9月28日 农历八月廿八 今日暂无日程安排	
星期四	9月29日 农历八月廿九 今日暂无日程安排	
星期五	9月30日 农历八月卅十 今日暂无日程安排	
星期六	10月1日 农历九月初一 国庆节 今日暂无日程安排	
星期日	10月2日 农历九月初二 今日暂无日程安排	



Senior

后退 日程安排 保存

当前日期： 2016年10月2日 农历九月初二
星期日

日程时间： 09:30

提醒间隔： 提前半小时

日程标题： 去海边玩

日程内容：

照相
捡贝壳
露营
烧烤
爱干啥就干啥

5-33

ListView7()ListView

BaseAdapter

ViewPager52ViewPager

Fragment5252Fragment

FragmentManagerAdapter52FragmentViewPager

PagerTabStripPagerTabStrip

BroadcastFragmentActivityFragmentActivity

TimePickerDialog
AlarmManagerAlarmManager
TextViewButtonEditTextSpinnerSQLite

5.6.2 Vibrator VibratorVIBRATOR_SERVICE

```
hasVibrator  
vibrate  
cancel  
AndroidManifest.xml  
<!-- -->  
<uses-permission android:name="android.permission.VIBRATE" />
```

```
Vibrator vibrator = (Vibrator)  
getSystemService(Context.VIBRATOR_SERVICE); vibrator.vibrate(3000); //3
```

5.6.3

```
packageActivity  
5  
 package7  
com.example.schedule.activityAcitivity  
com.example.schedule.adapter  
com.example.schedule.bean  
com.example.schedule.databaseSQLite  
com.example.schedule.fragment  
com.example.schedule.receiver  
com.example.schedule.util
```



packageApppackage

步骤 02

CalendarActivity.javaactivity_calendar.xml
ScheduleActivity.javaactivity_schedule.xmlScheduleDetailActivity.java
activity_schedule_detail.xmlMainApplication.javaAlarmReceiver.java

步骤 03

AndroidManifest.xmlAndroidManifest.xml3
(1)3activity
<activity android:name=".activity.CalendarActivity" /> <activity
android:name=".activity.ScheduleActivity" /> <activity
android:name=".activity.ScheduleDetailActivity" /> (2)receiver

```
<receiver android:name=".receiver.AlarmReceiver" >  
  <intent-filter>  
    <action android:name="com.example.senior.ScheduleDetailActivity.AlarmReceiver" />  
  </intent-filter>  
</receiver>
```

(3)

步骤 04

res/drawableres/layout

步骤 05

java

2016年 日程安排		
第三十九周	第四十周	第四十一周
星期一	9月26日 农历八月廿六 14时30分：培训	
星期二	9月27日 农历八月廿七 今日暂无日程安排	
星期三	9月28日 农历八月廿八 今日暂无日程安排	
星期四	9月29日 农历八月廿九 今日暂无日程安排	
星期五	9月30日 农历八月卅十 17时00分：坐高铁	
星期六	10月1日 农历九月初一 国庆节 今日暂无日程安排	
星期日	10月2日 农历九月初二 09时30分：去海边玩	

5-34

ScheduleDetailActivity.java

```

private String[] alarmArray = {"不提醒", "提前 5 分钟", "提前 10 分钟",
    "提前 15 分钟", "提前半小时", "提前 1 小时", "当前时间后 10 秒"};
private int[] advanceArray = {0, 5, 10, 15, 30, 60, 10};
private int alarmType = 0;
class AlarmSelectedListener implements OnItemSelectedListener {
    public void onItemSelected(AdapterView<?> arg0, View arg1, int arg2, long arg3) {
        alarmType = arg2;
    }

    public void onNothingSelected(AdapterView<?> arg0) {
    }
}

@Override
protected void onResume() {
    super.onResume();
    arrange = new ScheduleArrange();
    arrangeHelper = new ScheduleArrangeHelper(this, DbHelper.db_name, null, 1);
    List<ScheduleArrange> arrangeList = (List<ScheduleArrange>)
arrangeHelper.queryInfoByDay(day);
    if (arrangeList.size() >= 1) {
        enableEdit(false);
        arrange = arrangeList.get(0);
        schedule_time.setText(arrange.hour+":"+arrange.minute);
        schedule_alarm.setSelection(arrange.alarm_type);
        schedule_title.setText(arrange.title);
        schedule_content.setText(arrange.content);
    }
}

```

```

    } else {
        enableEdit(true);
    }
}

@Override
protected void onPause() {
    super.onPause();
    arrangeHelper.close();
}

@Override
public void onClick(View v) {
    int resid = v.getId();
    if (resid == R.id.schedule_time) {
        Calendar calendar = Calendar.getInstance();
        TimePickerDialog dialog = new TimePickerDialog(this, this,
            calendar.get(Calendar.HOUR_OF_DAY), calendar.get(Calendar.MINUTE),
true);
        dialog.show();
    } else if (resid == R.id.btn_back) {
        finish();
    } else if (resid == R.id.btn_edit) {
        enableEdit(true);
    } else if (resid == R.id.btn_save) {
        if (schedule_title.getText().toString().equals("")) == true) {
            Toast.makeText(this, "请输入日程标题", Toast.LENGTH_SHORT).show();
            return;
        }
        enableEdit(false);
        String[] time_split = schedule_time.getText().toString().split(":");
        arrange.hour = time_split[0];
        arrange.minute = time_split[1];
        arrange.alarm_type = alarmType;
        arrange.title = schedule_title.getText().toString();
        arrange.content = schedule_content.getText().toString();
        if (arrange.xuhao <= 0) {
            arrange.month = month;
            arrange.day = day;
            arrangeHelper.add(arrange);
        } else {
            arrangeHelper.update(arrange);
        }
    }
}

```

```

Toast.makeText(this, "保存日程成功", Toast.LENGTH_SHORT).show();
//设置提醒闹钟
if (alarmType > 0) {
    Intent intent = new Intent(ALARM_EVENT);
    PendingIntent pIntent = PendingIntent.getBroadcast(this, 0, intent,
        PendingIntent.FLAG_UPDATE_CURRENT);
    AlarmManager alarmMgr = (AlarmManager) getSystemService(ALARM_SERVICE);
    Calendar calendar = Calendar.getInstance();
    if (alarmType == 6) {
        calendar.setTimeInMillis(System.currentTimeMillis());
        calendar.add(Calendar.SECOND, advanceArray[alarmType]);
    } else {
        int day_int = Integer.parseInt(day);
        calendar.set(day_int/10000, day_int%10000/100-1, day_int%100,
            Integer.parseInt(advance.hour), Integer.parseInt(advance.minute), 0);
        calendar.add(Calendar.SECOND, -advanceArray[alarmType]*60);
    }
    alarmMgr.set(AlarmManager.RTC_WAKEUP, calendar.getTimeInMillis(), pIntent);
}
}

@Override
public void onTimeSet(TimePicker view, int hourOfDay, int minute) {
    String time = String.format("%02d:%02d", hourOfDay, minute);
    schedule_time.setText(time);
}

private String ALARM_EVENT = "com.example.senior.ScheduleDetailActivity.AlarmReceiver";
public static class AlarmReceiver extends BroadcastReceiver {
    @Override
    public void onReceive(Context context, Intent intent) {
        if (intent != null) {
            Log.d(TAG, "AlarmReceiver onReceive");
            Vibrator vibrator = (Vibrator)
context.getSystemService(Context.VIBRATOR_SERVICE);
            vibrator.vibrate(3000); // 默认震动 3 秒
        }
    }
}
}

```

5.7

App()()()Broadcast

()“/”App

3

(1)

(2)

(3)Broadcast

第6章 自定义控件



AppService“”

6.1

33onLayoutonDrawdispatchDraw

6.1.1

Android

5PagerTitleStripPagerTabStripTextColor textSize

步骤 01

res\values\attrs.xml declare-styleable name PagerTab
attr textColor textSize

```
<resources>  
  <declare-styleable name="PagerTab">  
    <attr name="textColor" format="color" />  
    <attr name="textSize" format="dimension" />  
  </declare-styleable>  
</resources>
```

步骤 02

PagerTab.java

```

public class PagerTab extends PagerTabStrip {
    private int textColor = Color.BLACK;
    private int textSize = 15;

    public PagerTab(Context context) {
        super(context);
    }

    public PagerTab(Context context, AttributeSet attrs) {
        super(context, attrs);
        if (attrs != null) {
            TypedArray attrArray=getContext().obtainStyledAttributes(attrs, R.styleable.PagerTab);
            textColor = attrArray.getColor(R.styleable.PagerTab_textColor, textColor);
            textSize = attrArray.getDimensionPixelSize(R.styleable.PagerTab_textSize, textSize);
            attrArray.recycle();
        }
    }

    @Override
    protected void onDraw(Canvas canvas) {
        setTextColor(textColor);
        setTextSize(TypedValue.COMPLEX_UNIT_SP, textSize);
        super.onDraw(canvas);
    }
}

```

步骤 03

xmlns:app="http://schemas.android.com/apk/res-auto"
 android.support.v4.view.PagerTabStrip(
 com.example.custom.widget.PagerTab)——app:textColorapp:textSize


```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="10dp" >

    <android.support.v4.view.ViewPager
        android:id="@+id/vp_content"
        android:layout_width="match_parent"
        android:layout_height="400dp" >

        <com.example.custom.widget.PagerTab
            android:id="@+id/pts_tab"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            app:textColor="@color/red"
            app:textSize="17sp" />

    </android.support.v4.view.ViewPager>
</LinearLayout>
```

App6-1

1attrnameformat()2getColorgetColorDimensionPixelSize

6-1

Custom

iphone7

Mate8

小米5



6-1

6-1

属性类型	获取方法	说明
boolean	getBoolean	布尔值。取值为 true 或 false
integer	getInt	整型值
float	getFloat	浮点值
string	getString	字符串
color	getColor	颜色值。取值为开头带#的六位或八位十六进制数
dimension	getDimension	尺寸值。取值为末尾带 dp 的尺寸数值
dimension	getDimensionPixelSize	字体大小。取值为末尾带 sp 的尺寸数值
fraction	getFraction	百分数。取值为末尾带%的百分数
reference	getResourceId	参考某一资源。取值如@drawable/ic_launcher
enum	getInt	枚举值
flag	getInt	标志位

6-1enumflagLinearLayoutorientationImageViewscaleTypeTextView gravityEditTextinputType

```
<declare-styleable name="PagerTab">
  <attr name="customOrientation">
    <enum name="horizontal" value="0" />
    <enum name="vertical" value="1" />
  </attr>
</declare-styleable>
```

```
<declare-styleable name="PagerTab">
  <attr name="customGravity">
    <flag name = "center" value = "0" />
    <flag name = "left" value = "1" />
    <flag name = "top" value = "2" />
    <flag name = "right" value = "4" />
    <flag name = "bttom" value = "8" />
  </attr>
</declare-styleable>
```

6.1.2

3

(1)

(2)onMeasure()

(3)onLayoutonDrawdispatchDraw3
 3PagerTab3()
 (1)public PagerTab(Context context)
 (2)public PagerTab(Context context,AttributeSet attrs)
 (3)3public PagerTab(Context context, AttributeSet attrs, int defStyleAttr)
 3defStyleAttr(referencestyles.xml)

步骤 01 styles.xml

步骤 02 attrs.xml

<attr name="CustomizeStyle" format="reference" /> **步骤 03**

```

public PagerTab(Context context, AttributeSet attrs) {
    this(context, attrs, R.attr.CustomizeStyle);
}

public PagerTab(Context context, AttributeSet attrs, int defStyleAttr) {
    super(context, attrs, defStyleAttr);
    if (attrs != null) {
        TypedArray attrArray = getContext().obtainStyledAttributes( attrs, R.styleable.PagerTab,
defStyleAttr, R.style.DefaultCustomizeStyle);
        //此处省略各个属性值的读取
        attrArray.recycle();
    }
}

```

attrs.xmlR.attr.CustomizeStylestyles.xml
 R.style.DefaultCustomizeStyle

6.1.3

36-2

XML 中的尺寸类型	LayoutParams 类的尺寸类型	说明
match_parent	MATCH_PARENT	与上级视图大小一样
wrap_content	WRAP_CONTENT	按照自身尺寸进行适配
**dp	整型数	具体的尺寸数值

132Android3

1.

PaintmeasureText

```
public static float getTextWidth(String text, float textSize) {
    if (text==null || text.length()<=0) {
        return 0;
    }
    Paint paint = new Paint();
    paint.setTextSize(textSize);
    return paint.measureText(text);
}
```

FontMetrics56-3

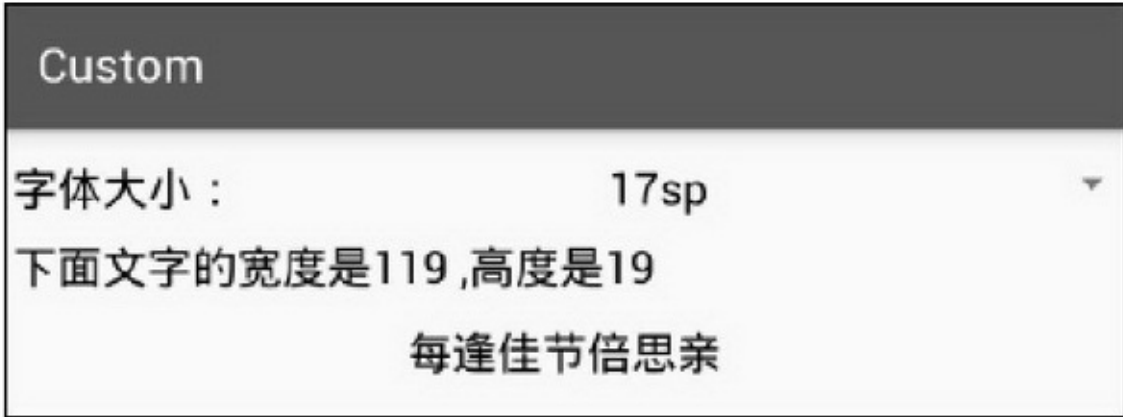
6-3FontMetrics

FontMetrics 类的距离属性	说明
top	行的顶部与基线的距离
ascent	字符的顶部与基线的距离
descent	字符的底部与基线的距离
bottom	行的底部与基线的距离
leading	行间距

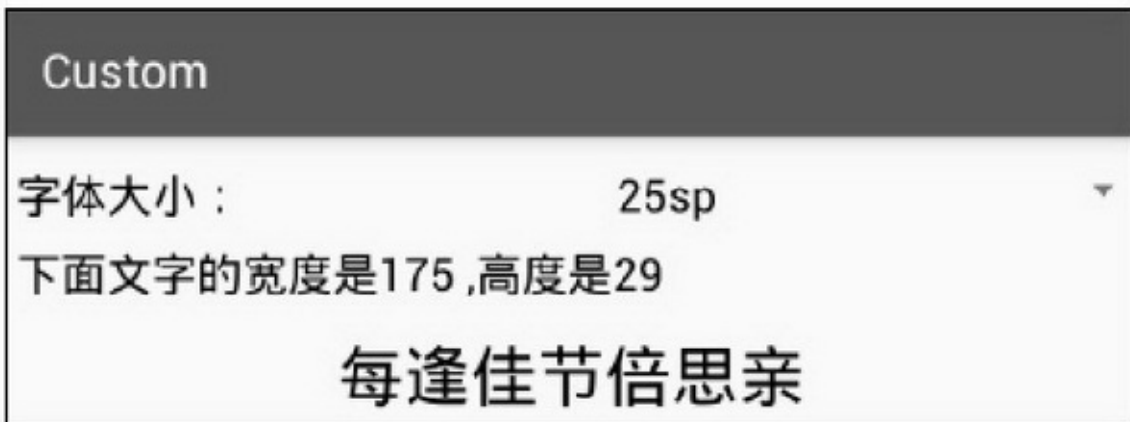
descentascentbottomtopleading

```
public static float getTextHeight(String text, float textSize) {
    Paint paint = new Paint();
    paint.setTextSize(textSize);
    FontMetrics fm = paint.getFontMetrics();
    return fm.descent - fm.ascent; //文本自身的高度
    //return fm.bottom - fm.top + fm.leading; //文本所在行的行高
}
```

17sp 119 196-225sp 175 296-3



6-217sp



6-325sp

2.

AndroidBitmap.getWidth().getHeight().getDrawable()
getWidth().getHeight()

3.

TextView.Button.Image.View.Image.Button
padding.margin.View.layout_width.layout_height.3.Android.3

6-4

6-4

MeasureSpec 类的测量模式	视图宽、高的赋值方式	说明
AT_MOST	MATCH_PARENT	达到最大
UNSPECIFIED	WRAP_CONTENT	未指定（实际就是自适应）
EXACTLY	具体 dp 值	精确尺寸

3 ViewGroup.getChildMeasureSpec MeasureSpec.makeMeasureSpec MeasureSpec View.measure

```

public static float getRealHeight(View child) {
    LinearLayout llayout = (LinearLayout) child;
    ViewGroup.LayoutParams params = llayout.getLayoutParams();
    if (params == null) {
        params = new ViewGroup.LayoutParams(
            LayoutParams.MATCH_PARENT, LayoutParams.WRAP_CONTENT);
    }
    int widthSpec = ViewGroup.getChildMeasureSpec(0, 0, params.width);
    int heightSpec;
    if (params.height > 0) {
        heightSpec = View.MeasureSpec.makeMeasureSpec(params.height,
MeasureSpec.EXACTLY);
    } else {
        heightSpec = View.MeasureSpec.makeMeasureSpec(0, MeasureSpec.UNSPECIFIED);
    }
    llayout.measure(widthSpec, heightSpec);
    return llayout.getMeasuredHeight(); // 获得布局高度，获得布局宽度可调用
getMeasuredWidth
}

```

App6-4getRealHeight170



6-4

6.1.4

3onLayoutonDrawdispatchDraw3
onLayout → onDraw → dispatchDrawonLayoutdispatchDraw3

1. onLayout

onLayoutlayout
onLayout


```

public class OffsetLayout extends AbsoluteLayout {
    private int mOffsetHorizontal = 0, mOffsetVertical = 0;
    public OffsetLayout(Context context) {
        super(context);
    }

    public OffsetLayout(Context context, AttributeSet attrs) {
        super(context, attrs);
    }

    @Override
    protected void onLayout(boolean changed, int l, int t, int r, int b) {
        int count = getChildCount();
        for (int i = 0; i < count; i++) {
            View child = getChildAt(i);
            if (child.getVisibility() != GONE) {
                int new_left = (r-l)/2-child.getMeasuredWidth()/2+mOffsetHorizontal;
                int new_top = (b-t)/2-child.getMeasuredHeight()/2+mOffsetVertical;
                child.layout(new_left, new_top,
                    new_left+child.getMeasuredWidth(), new_top+child.getMeasuredHeight());
            }
        }
    }

    public void setOffsetHorizontal(int offset) {
        mOffsetHorizontal = offset;
        mOffsetVertical = 0;
        requestLayout();
    }

    public void setOffsetVertical(int offset) {
        mOffsetHorizontal = 0;
        mOffsetVertical = offset;
        requestLayout();
    }
}

```

OffsetLayoutsetOffsetHorizontalsetOffsetVertical6-5~6-84
6-56-61006-71006-8100



6-5

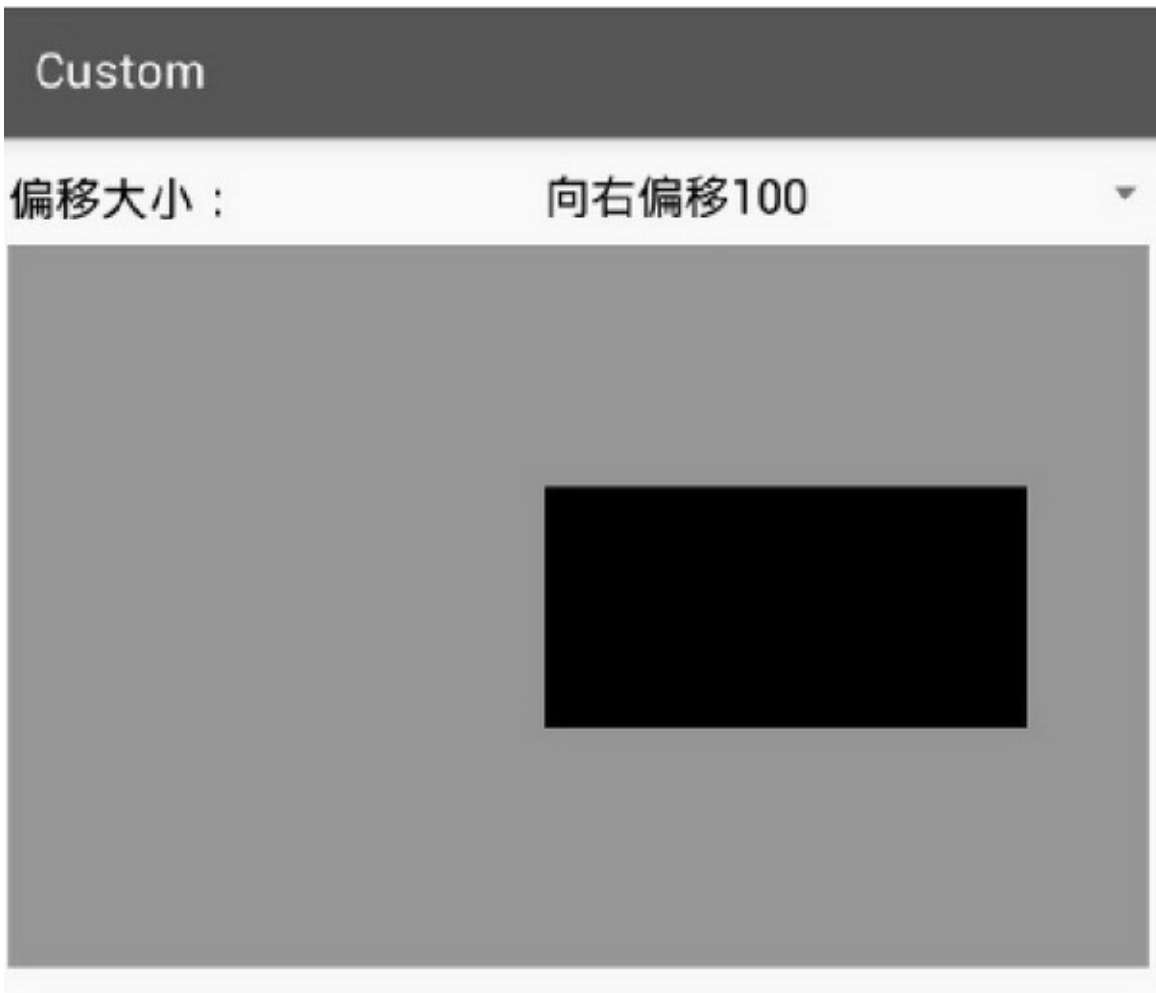
Custom

偏移大小：

向左偏移100



6-6100



6-7100



6-8100

2. onDraw

onDrawCanvasCanvas3

(1)

clipPath

clipRect

clipRegion

(2)

drawArc/4truefalse

drawBitmap

drawCircle

drawLine

drawOval

drawPath

drawPoint

drawRect

drawRoundRect

drawText

(3)

rotate

scale

translate

save

restore

draw***PaintPaint

setAntiAlias

setDither

setColor

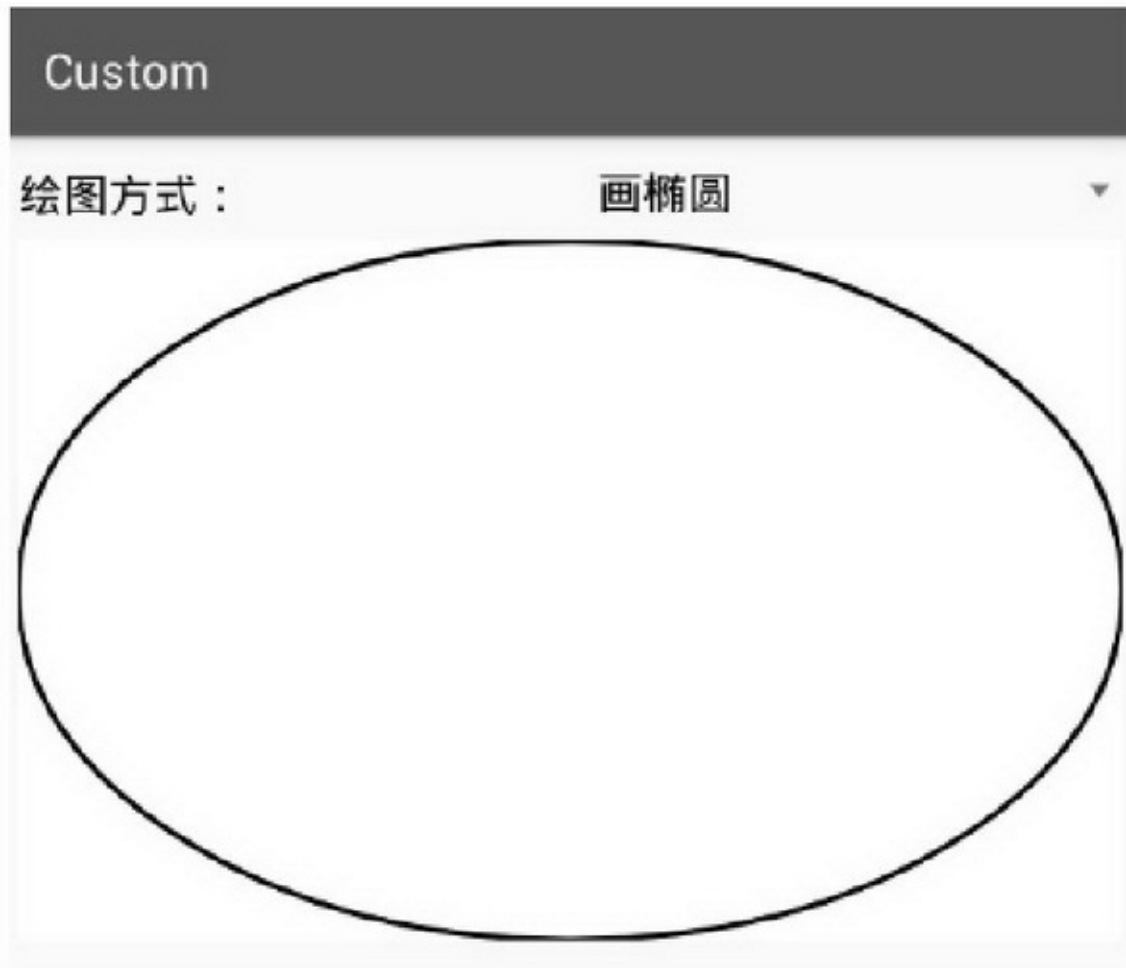
setShadowLayer

setStyleStyle.STROKEStyle.FILL

setStrokeWidth

drawRoundRect6-9drawOval6-10





6-10

3. dispatchDraw

dispatchDrawonDrawonDrawdispatchDrawdispatchDraw

dispatchDraw


```
public class AfterRelativeLayout extends RelativeLayout {
    private int mDrawType = 0;
    private Paint mPaint = new Paint();

    public AfterRelativeLayout(Context context) {
        this(context, null);
    }

    public AfterRelativeLayout(Context context, AttributeSet attrs) {
        super(context, attrs);
        mPaint.setAntiAlias(true);           //设置画笔为无锯齿
        mPaint.setDither(true);             //防抖动
        mPaint.setColor(Color.BLACK);       //设置画笔颜色
        mPaint.setStrokeWidth(3);           //线宽
        mPaint.setStyle(Style.STROKE);      //画笔类型。STROKE: 空心, FILL: 实心
    }

    @Override
    protected void dispatchDraw(Canvas canvas) {
        super.dispatchDraw(canvas);
        int width = getMeasuredWidth();
        int height = getMeasuredHeight();
        if (width>0 && height>0) {
            if (mDrawType == 1) { // 画矩形
```

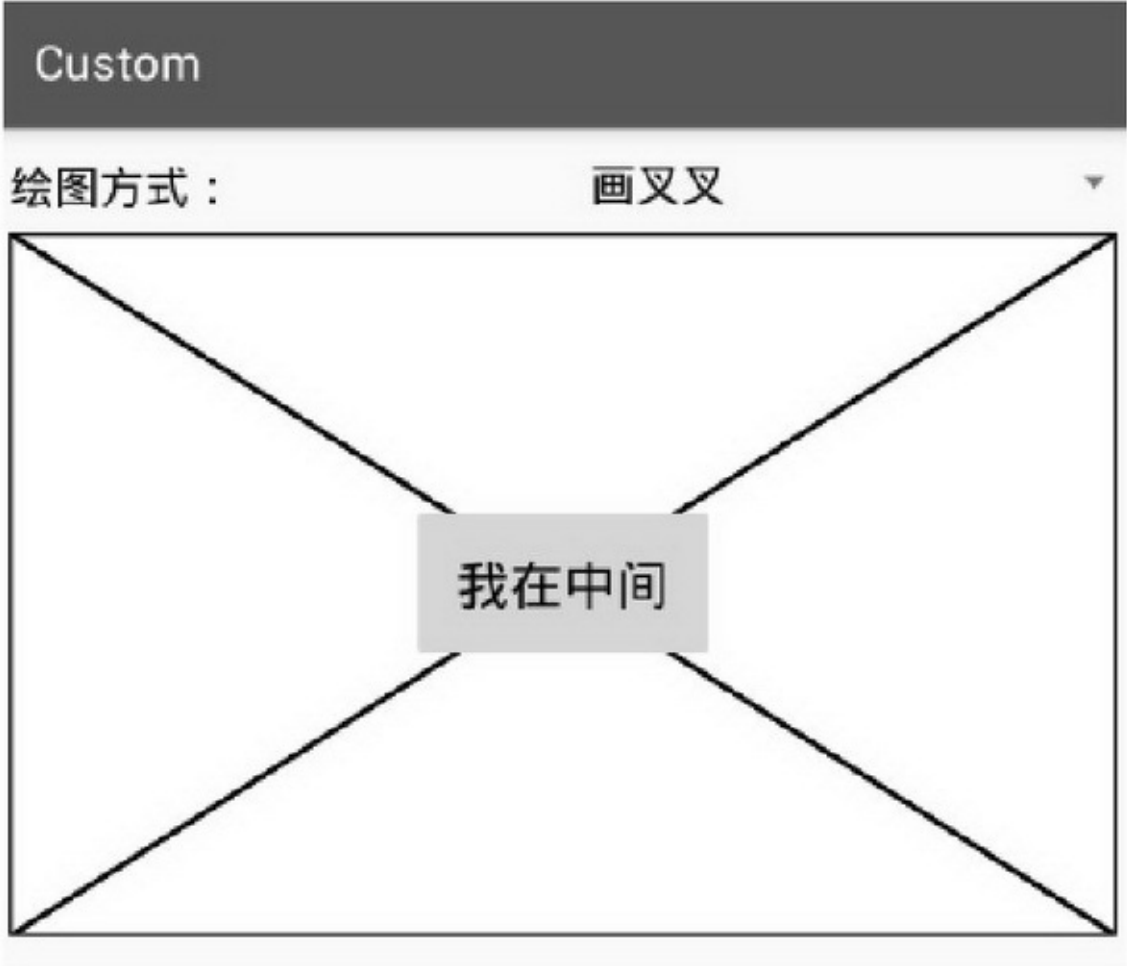
```

        Rect rect = new Rect(0, 0, width, height);
        canvas.drawRect(rect, mPaint);
    } else if (mDrawType == 2) { // 画圆角矩形
        RectF rectF = new RectF(0, 0, width, height);
        canvas.drawRoundRect(rectF, 30, 30, mPaint);
    } else if (mDrawType == 3) { // 画圆圈
        int radius = Math.min(width, height)/2;
        canvas.drawCircle(width/2, height/2, radius, mPaint);
    } else if (mDrawType == 4) { // 画椭圆
        RectF oval = new RectF(0, 0, width, height);
        canvas.drawOval(oval, mPaint);
    } else if (mDrawType == 5) { // 画线段
        Rect rect = new Rect(0, 0, width, height);
        canvas.drawRect(rect, mPaint);
        canvas.drawLine(0, 0, width, height, mPaint);
        canvas.drawLine(0, height, width, 0, mPaint);
    }
}

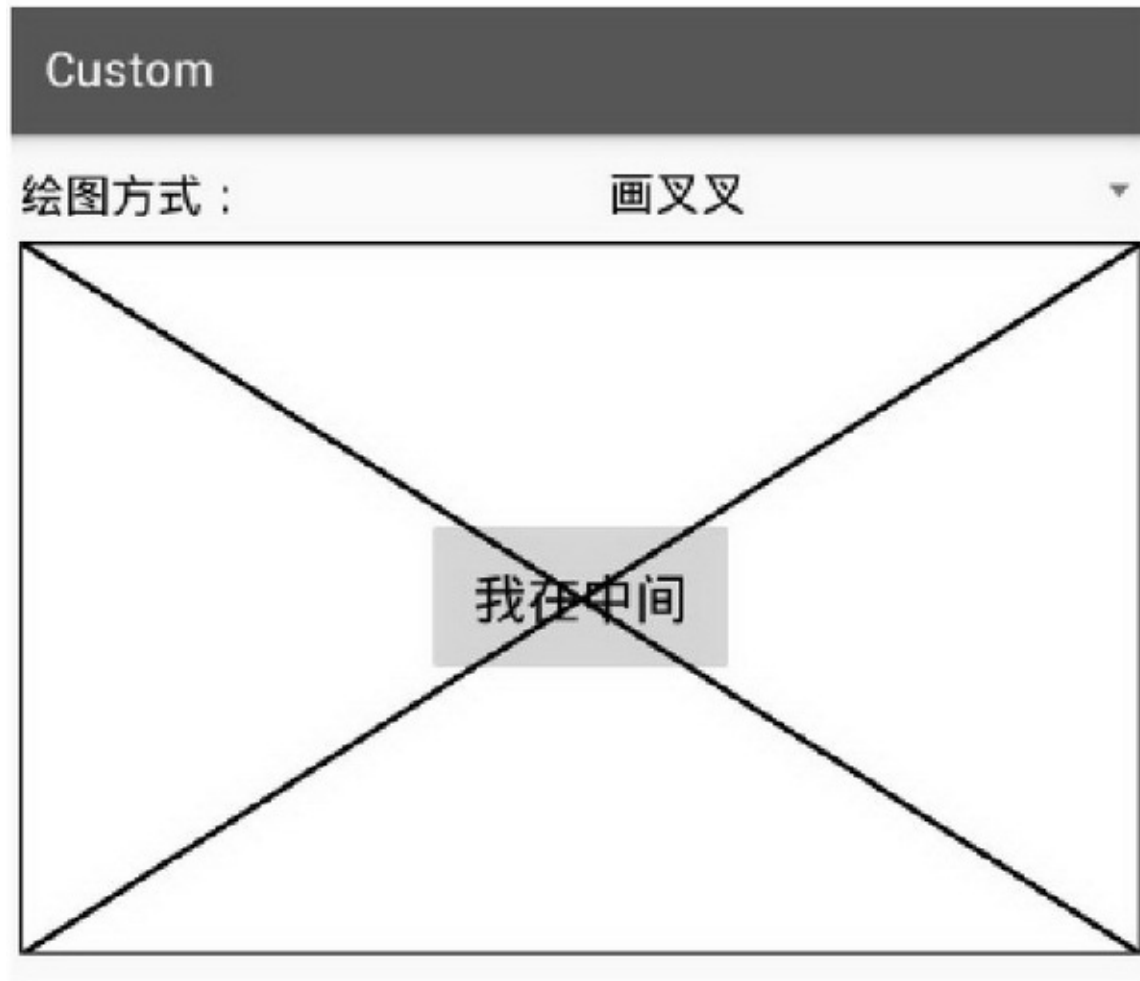
public void setDrawType(int type) {
    setBackgroundColor(Color.WHITE);
    mDrawType = type;
    invalidate();
}
}

```

onDraw
 dispatchDraw
 onDraw
 dispatchDraw
 onDraw
 dispatchDraw



6-11onDraw



6-12dispatchDraw

onLayoutonDrawdispatchDraw

(1)onLayoutonDrawdispatchDraw

(2)onDrawdispatchDraw

(3)onLayoutrequestLayoutonDrawdispatchDrawinvalidate

6.2

6.2.1 Runnable Handler+RunnableHandlerpostDelayed

Runnable

RunnableRunnableRunnablerunrunRunnablerun

ViewpostpostDelayedRunnableHandlerHandlerRunnable

postRunnable

postDelayedRunnable

postAtTimeRunnable

removeCallbacksRunnable

Runnable20

```

@Override
public void onClick(View v) {
    if (v.getId() == R.id.btn_runnable) {
        if (bStart == false) {
            btn_runnable.setText("停止计数");
            mHandler.post(mCounter);
        } else {
            btn_runnable.setText("开始计数");
            mHandler.removeCallbacks(mCounter);
        }
        bStart = !bStart;
    }
}

private boolean bStart = false;
private Handler mHandler = new Handler();
private int mCount = 0;
private Runnable mCounter = new Runnable() {
    @Override
    public void run() {
        mCount++;
        tv_result.setText("当前计数值为 : "+mCount);
        mHandler.postDelayed(this, 1000);
    }
};

```

6-136-146-136-1421





6-14

6.2.2

6.1paddingpaddingpadding

Runnable

```

@Override
public void onClick(View v) {
    mHeight = (int) MeasureUtil.getRealHeight(ll_header);
    if (v.getId() == R.id.btn_pull) {
        if (bStart == false) {
            mOffset = -mHeight;
            btn_pull.setEnabled(false);
            mHandler.post(mRefresh);
        } else {
            btn_pull.setText("开始刷新");
            ll_header.setVisibility(View.GONE);
        }
        bStart = !bStart;
    }
}

private int mHeight;
private boolean bStart = false;
private Handler mHandler = new Handler();
private int mOffset = 0;
private Runnable mRefresh = new Runnable() {
    @Override
    public void run() {
        if (mOffset <= 0) {
            ll_header.setPadding(0, mOffset, 0, 0);
            ll_header.setVisibility(View.VISIBLE);
            mOffset += 8;
            mHandler.postDelayed(this, 80);
        } else {
            btn_pull.setText("恢复页面");
            btn_pull.setEnabled(true);
        }
    }
};

```

6-156-166-156-16



6-15



6-16

6.2.3

Windows6.1

00

```

public class CircleAnimation extends RelativeLayout {
    private final static String TAG = "CircleAnimation";
    private RectF mRect = new RectF(100, 10, 400, 310);
    private int mBeginAngle = 0, mEndAngle = 270;
    private int mFrontColor = 0xff0000, mShadeColor = 0xffe0e0;
    private float mFrontLine = 10, mShadeLine = 10;
    private Style mFrontStyle = Style.STROKE, mShadeStyle = Style.STROKE;
    private ShadeView mShadeView;
    private FrontView mFrontView;
    private int mRate = 2, mDrawTimes = 0, mInterval = 70, mFactor, mSeq = 0, mDrawingAngle = 0;
    private Context mContext;

    public CircleAnimation(Context context) {
        super(context);
        mContext = context;
    }

    public void render() {
        removeAllViews();
        mShadeView = new ShadeView(mContext);
        addView(mShadeView);
        mFrontView = new FrontView(mContext);
        addView(mFrontView);
        play();
    }

    public void play() {
        mSeq = 0;
        mDrawingAngle = 0;
        mDrawTimes = mEndAngle/mRate;
        mFactor = mDrawTimes/mInterval + 1;
        Log.d(TAG, "mDrawTimes="+mDrawTimes+",mInterval="+mInterval+",mFactor="+mFactor);
        mFrontView.invalidateView();
    }

    public void setRect(int left, int top, int right, int bottom) {
        mRect = new RectF(left, top, right, bottom);
    }

    public void setAngle(int begin_angle, int end_angle) {
        mBeginAngle = begin_angle;
        mEndAngle = end_angle;
    }
}

```

```

}

//speed:每次移动几个度数  frames:每秒移动几帧
public void setmRate(int speed, int frames) {
    mRate = speed;
    mInterval = 1000/frames;
}

public void setFront(int color, float line, Style style) {
    mFrontColor = color;
    mFrontLine = line;
    mFrontStyle = style;
}

public void setShade(int color, float line, Style style) {
    mShadeColor = color;
    mShadeLine = line;
    mShadeStyle = style;
}

private class ShadeView extends View {
    private Paint paint;
    public ShadeView(Context context) {
        super(context);
        paint = new Paint();
        paint.setAntiAlias(true);
        paint.setDither(true);
        paint.setColor(mShadeColor);
        paint.setStrokeWidth(mShadeLine);
        paint.setStyle(mShadeStyle);
    }

    @Override
    protected void onDraw(Canvas canvas) {
        super.onDraw(canvas);
        canvas.drawArc(mRect, mBeginAngle, 360, false, paint);
    }
}

private class FrontView extends View {
    private Paint paint;
    private Handler handler = new Handler();
    public FrontView(Context context) {
        super(context);
        paint = new Paint();
        paint.setAntiAlias(true); //设置画笔为无锯齿
    }
}

```






6.3

5——

6.3.1 Dialog AppWindowToastDialog

Window5

setContentviewActivitysetContentviewgetWindow().

setContentview

setLayout

setGravity

setBackgroundDrawable

findViewByIdIDActivityActivityWindowgetWindow().

findViewById

WindowDialog3

步骤 01

步骤 02

步骤 03

Dialog(styles.xml)

getWindow

show

isShowing

hide

dismiss

setCancelable

setCanceledOnTouchOutside

setOnShowListenerOnShowListeneronShow
setOnDismissListenerOnDismissListeneronDismiss

6.3.2

AlertDialog5DatePickerDialogTimePickerDialog
DatePickerDialog“yyyy-mm-dd”“”6-19





6-20

6-20“”“”3

步骤

01

TextViewDatePickerButton

```

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/transparent"
    android:gravity="center"
    android:orientation="vertical"
    android:paddingLeft="40dp"
    android:paddingRight="40dp" >

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:orientation="vertical"
        android:background="@color/white" >

        <TextView
            android:id="@+id/tv_title"
            android:layout_width="match_parent"
            android:layout_height="60dp"
            android:paddingLeft="10dp"
            android:gravity="left|center"
            android:text="请选择日期"
            android:textColor="@color/blue"
            android:textSize="22sp" />

        <View
            android:layout_width="match_parent"
            android:layout_height="2dp"
            android:background="@color/blue" />

        <DatePicker
            android:id="@+id/dp_date"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:calendarViewShown="false"
            android:gravity="center"
            android:spinnersShown="true" />

        <View
            android:layout_width="match_parent"

```

```

    android:layout_height="1dp"
    android:background="@color/blue" />

```

```

<Button
    android:id="@+id/btn_ok"
    android:layout_width="match_parent"
    android:layout_height="60dp"
    android:background="@null"
    android:gravity="center"
    android:text="确定"
    android:textColor="@color/black"
    android:textSize="17sp" />

```

```

</LinearLayout>

```

```

</LinearLayout>

```

步骤 02

```
public class CustomDateDialog implements OnClickListener, OnDateChangeListener {
    private Dialog dialog;
    private View view;
    private TextView tv_title;
    private DatePicker dp_date;

    public CustomDateDialog(Context context) {
        view = LayoutInflater.from(context).inflate(R.layout.dialog_date, null);
        dialog = new Dialog(context, R.style.CustomDateDialog);
        tv_title = (TextView) view.findViewById(R.id.tv_title);
        dp_date = (DatePicker) view.findViewById(R.id.dp_date);
        view.findViewById(R.id.btn_ok).setOnClickListener(this);
    }

    public void setTitle(String title) {
        tv_title.setText(title);
    }

    public void setDate(int year, int month, int day, OnDateSetListener listener) {
        dp_date.init(year, month, day, this);
        mDateSetListener = listener;
    }

    public void show() {
        dialog.getWindow().setContentView(view);
        dialog.getWindow().setLayout(LayoutParams.MATCH_PARENT, LayoutParams.WRAP_CONTENT);
    }
}
```

```

        dialog.show();
    }

    public void dismiss() {
        if (dialog != null && dialog.isShowing()) {
            dialog.dismiss();
        }
    }

    public boolean isShowing() {
        if (dialog != null) {
            return dialog.isShowing();
        } else {
            return false;
        }
    }

    @Override
    public void onClick(View v) {
        dismiss();
        if (mDateSetListener != null) {
            dp_date.clearFocus();
            //这里给系统月份加一，调用时就不必每次都加一了
            mDateSetListener.onDateSet(dp_date.getYear(),
                dp_date.getMonth()+1, dp_date.getDayOfMonth());
        }
    }

    @Override
    public void onChanged(DatePicker view, int year, int monthOfYear, int dayOfMonth) {
        dp_date.init(year, monthOfYear, dayOfMonth, this);
    }

    private OnDateSetListener mDateSetListener;
    public interface OnDateSetListener {
        void onDateSet(int year, int monthOfYear, int dayOfMonth);
    }
}

```

步骤

03

Activity

```

Calendar calendar = Calendar.getInstance();
CustomDateDialog dialog = new CustomDateDialog(this);
dialog.setDate(calendar.get(Calendar.YEAR), calendar.get(Calendar.MONTH),
    calendar.get(Calendar.DAY_OF_MONTH), this);
dialog.show();

```

6.3.3

6-216-226-21ListView6-22GridView

您将添加以下号码为好友			
关系	手机号码	允许访问朋友圈	操作
叔叔	▼ 15960238696	<input checked="" type="radio"/> 允许 <input type="radio"/> 禁止	删除
表哥	▼ 15805910591	<input checked="" type="radio"/> 允许 <input type="radio"/> 禁止	删除
朋友	▼ 18905710571	<input checked="" type="radio"/> 允许 <input type="radio"/> 禁止	删除
确定			

6-21

其他				
其他	本人	妻子	哥哥	弟弟
姐姐	妹妹	表哥	表弟	表姐
表妹	叔叔	婶婶	姑妈	姑父
舅舅	舅妈	阿姨	姨夫	朋友
其他	∨ 15805910591	<input checked="" type="radio"/> 允许	<input type="radio"/> 禁止	删除
其他	∨ 18905710571	<input checked="" type="radio"/> 允许	<input type="radio"/> 禁止	删除
确定				

6-22

6.4

NotificationProcessBarRemoteViews

6.4.1 Notification AppNotification

Notification.Builder

setWhen“”

setShowWhen

setUsesChronometertrue“”

setSmallIcon()

setTicker

setLargeIcon()

setContentTitle

setContentText

setSubTextsetProgress

setProgress

setNumbersetProgress

setContentInfosetContentNumber

setContentIntentPendingIntentPendingIntentgetActivity

getActivity

setDeleteIntentPendingIntent

setAutoCanceltruefalse

setContentRemoteViewsBuilder

buildNotification

4

(1)setSmallIcon

(2)setWhensetUsesChronometer

(3)setSubTextsetProgress

(4)setNumbersetContentInfo

NotificationNotificationManager3

notify

cancel

cancelAll

```
private void sendSimpleNotify(String title, String message) {
    Intent clickIntent = new Intent(this, MainActivity.class);
    PendingIntent contentIntent = PendingIntent.getActivity(this,
        R.string.app_name, clickIntent, PendingIntent.FLAG_UPDATE_CURRENT);
    Notification.Builder builder = new Notification.Builder(this);
    builder.setContentIntent(contentIntent)
        .setAutoCancel(true).setSmallIcon(R.drawable.ic_app)
        .setTicker("提示消息来啦").setWhen(System.currentTimeMillis())
        .setLargeIcon(BitmapFactory.decodeResource(getResources(), R.drawable.ic_app))
        .setContentTitle(title).setContentText(message);
    Notification notify = builder.build();
    NotificationManager notifyMgr = (NotificationManager)
getSystemService(Context.NOTIFICATION_SERVICE);
    notifyMgr.notify(R.string.app_name, notify);
}
```

6-23



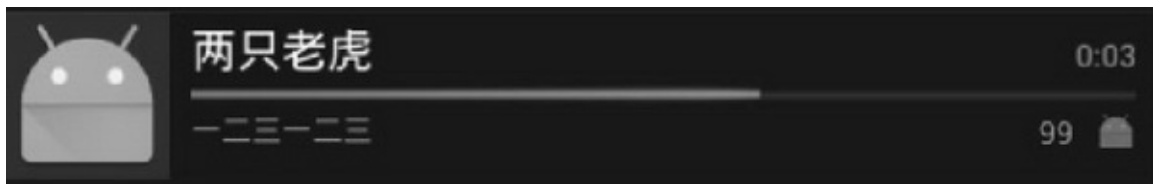
6-23


```

private void sendCounterNotify(String title, String message) {
    Intent cancelIntent = new Intent(this, MainActivity.class);
    PendingIntent deleteIntent = PendingIntent.getActivity(this,
        R.string.app_name, cancelIntent, PendingIntent.FLAG_UPDATE_CURRENT);
    Notification.Builder builder = new Notification.Builder(this);
    builder.setDeleteIntent(deleteIntent)
        .setAutoCancel(true).setUsesChronometer(true)
        .setProgress(100, 60, false).setNumber(99)
        .setSmallIcon(R.drawable.ic_app).setTicker("提示消息来啦")
        .setLargeIcon(BitmapFactory.decodeResource(getResources(), R.drawable.ic_app))
        .setContentTitle(title).setContentText(message);
    Notification notify = builder.build();
    NotificationManager notifyMgr = (NotificationManager)
getSystemService(Context.NOTIFICATION_SERVICE);
    notifyMgr.notify(R.string.app_name, notify);
}

```

6-24



6-24

6.4.2 ProgressBar NotificationsetProgressProgressBar

```

styleandroid:attr/progressBarStyleHorizontal
android:attr/progressBarStyle
max
progress
secondaryProgressprogresssecondaryProgress
progressDrawable
9
setProgress

```

getProgress
setSecondaryProgress
getSecondaryProgress
setMax
getMax
incrementProgressBy
incrementSecondaryProgressBy
setProgressDrawable

(1)maxprogressprogressBarStyleHorizontalprogressBarStyle

6.2

(2)progressDrawableLayerDrawableXML

XMLlayer-list@android:id/background(shape)id
@android:id/progressClipDrawable(clip)

```
<layer-list xmlns:android="http://schemas.android.com/apk/res/android" >
  <item android:id="@android:id/background">
    <shape>
      <solid android:color="#333333" />
    </shape>
  </item>
  <item android:id="@android:id/progress">
    <clip>
      <nine-patch android:src="@drawable/notify_green" />
    </clip>
  </item>
</layer-list>
```

XML

```
<ProgressBar
    android:id="@+id/pb_progress"
    style="?android:attr/progressBarStyleHorizontal"
    android:layout_width="match_parent"
    android:layout_height="30dp"
    android:background="@color/black"
    android:max="100"
    android:progress="0"
    android:progressDrawable="@drawable/notify_progress_green" />
```

6-256-266-2506-264040%



6-250



6-2640

6.4.3 RemoteViews NotificationsetContentRemoteViewsBuilder

RemoteViewsActivityRemoteViews

(1)RemoteViewsActivity

(2)RemoteViewsTextViewImageViewImageButton

ProgressBarChronometer()AnalogClock()

(3)RemoteViewsset

RemoteViewsid

setViewVisibility

setViewPadding

setTextViewTextTextViewButton

setTextViewTextSizeTextViewButton

setTextColorTextViewButton

setTextViewCompoundDrawablesTextViewButton

setImageResourceImageViewImageButton

setImageBitmapImageViewImageButton

setChronometer

setProgressBar

setOnClickPendingIntent

RemoteViewsNotificationsetContent

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:minHeight="64dp"
    android:orientation="horizontal" >

    <ImageView
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:scaleType="fitCenter"
        android:src="@drawable/tt" />

    <LinearLayout
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_marginLeft="5dp"
        android:layout_marginRight="5dp"
        android:layout_weight="5"
```

```
        android:orientation="vertical" >
        <ProgressBar
            android:id="@+id/pb_play"
            style="?android:attr/progressBarStyleHorizontal"
            android:layout_width="match_parent"
            android:layout_height="0dp"
            android:layout_weight="1"
            android:max="100"
            android:progress="10" />
        <TextView
            android:id="@+id/tv_play"
            android:layout_width="match_parent"
            android:layout_height="0dp"
            android:layout_weight="1"
            android:textColor="#ffffff"
            android:textSize="17sp" />
    </LinearLayout>
    <LinearLayout
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:orientation="vertical" >
        <Chronometer
            android:id="@+id/chr_play"
            android:layout_width="match_parent"
            android:layout_height="0dp"
            android:layout_weight="1" />
        <Button
            android:id="@+id/btn_play"
            android:layout_width="match_parent"
            android:layout_height="0dp"
            android:layout_weight="2"
            android:text="暂停"
            android:textColor="#ffffff"
            android:textSize="17sp" />
    </LinearLayout>
</LinearLayout>
```

```

private void sendCustomNotify(Context ctx, String event, String song, boolean isPlay, int progress, long
time) {
    Intent pIntent = new Intent(event);
    PendingIntent nIntent = PendingIntent.getBroadcast(
        ctx, R.string.app_name, pIntent, PendingIntent.FLAG_UPDATE_CURRENT);
    RemoteViews notify_music = new RemoteViews(ctx.getPackageName(), R.layout.notify_music);
    if (isPlay == true) {
        notify_music.setTextViewText(R.id.btn_play, "暂停");
        notify_music.setTextViewText(R.id.tv_play, song+"正在播放");
        notify_music.setChronometer(R.id.chr_play, time, "%s", true);
    } else {
        notify_music.setTextViewText(R.id.btn_play, "继续");
        notify_music.setTextViewText(R.id.tv_play, song+"暂停播放");
        notify_music.setChronometer(R.id.chr_play, time, "%s", false);
    }
    notify_music.setProgressbar(R.id.pb_play, 100, progress, false);
    notify_music.setOnClickPendingIntent(R.id.btn_play, nIntent);
    Intent intent = new Intent(ctx, MainActivity.class);
    PendingIntent contentIntent = PendingIntent.getActivity(ctx,
        R.string.app_name, intent, PendingIntent.FLAG_UPDATE_CURRENT);
    Notification.Builder builder = new Notification.Builder(ctx);
    builder.setContentIntent(contentIntent)
        .setContent(notify_music).setTicker(song).setSmallIcon(R.drawable.tt_s);
    Notification notify = builder.build();
    NotificationManager notifyMgr = (NotificationManager) getSystemService
(Context.NOTIFICATION_SERVICE);
    notifyMgr.notify(R.string.app_name, notify);
}

```

6-27



6-27

6.5 Service

Service3Notification

6.5.1 Service

ServiceAndroid56.4AndroidAppServiceActivity

Service

onCreate

onStartAndroid2.0

onStartCommandAndroid2.06-5

6-5

返回值类型	返回值说明
START_STICKY	粘性的服务。如果服务进程被杀掉，就保留服务的状态为开始状态，但不保留传送的 Intent 对象。随后系统尝试重新创建服务，由于服务状态为开始状态，因此创建服务后一定会调用 onStartCommand 方法。如果在此期间没有任何启动命令传送给服务，参数 Intent 就为空值
START_NOT_STICKY	非粘性的服务。使用这个返回值时，如果服务被异常杀掉，系统就不会自动重启该服务
START_REDELIVER_INTENT	重传 Intent 的服务。使用这个返回值时，如果服务被异常杀掉，系统就会自动重启该服务，并传入 Intent 的原值
START_STICKY_COMPATIBILITY	START_STICKY 的兼容版本，不保证服务被杀掉后一定能重启

onDestroy

onBind

onRebindonUnbindtrue

onUnbindtrueonRebindfalsefalse

Service3

1.


```

public class SimpleService extends Service {
    private static final String TAG = "SimpleService";

    @Override
    public int onStartCommand(Intent intent, int flags, int startid) {
        Log.d(TAG, "测试服务到此一游!");
        return START_STICKY;
    }

    @Override
    public IBinder onBind(Intent intent) {
        return null;
    }
}

```

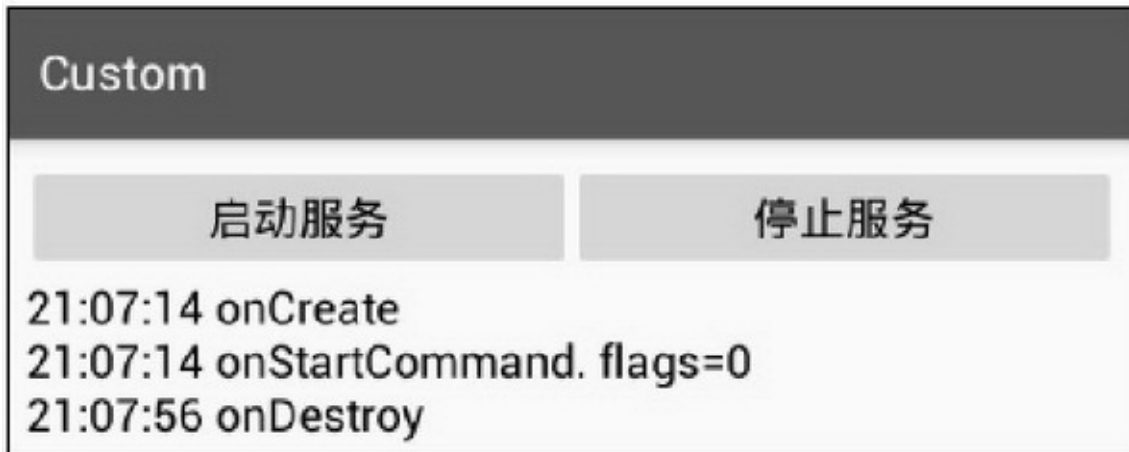
“”AcitivitystartServicestopServiceIntent

```

Intent intent = new Intent(this, SimpleService.class); startService(intent);
onCreateonStartCommand6-28onDestroy6-29

```





6-29

2.

Binder

```
public class BindImmediateService extends Service {
    private static final String TAG = "BindImmediateService";
    private final IBinder mBinder = new LocalBinder();
    public class LocalBinder extends Binder {
        public BindImmediateService getService() {
            return BindImmediateService.this;
        }
    }

    @Override
    public IBinder onBind(Intent intent) {
        Log.d(TAG, "绑定服务开始旅程!");
        return mBinder;
    }

    @Override
    public boolean onUnbind(Intent intent) {
        Log.d(TAG, "绑定服务结束旅程!");
        return false;
    }
}
```

“””Activity/ServiceConnection

bindServiceunbindService

```

public class BindImmediateActivity extends AppCompatActivity implements OnClickListener {
    private static TextView tv_immediate;
    private Intent mIntent;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_bind_immediate);
        tv_immediate = (TextView) findViewById(R.id.tv_immediate);
        findViewById(R.id.btn_start_bind).setOnClickListener(this);
        findViewById(R.id.btn_unbind).setOnClickListener(this);
        mIntent = new Intent(this, BindImmediateService.class);
    }

    @Override
    public void onClick(View v) {
        if (v.getId() == R.id.btn_start_bind) {
            boolean bindFlag = bindService(mIntent, mFirstConn, Context.BIND_AUTO_CREATE);
        } else if (v.getId() == R.id.btn_unbind) {
            if (mBindService != null) {
                unbindService(mFirstConn);
                mBindService = null;
            }
        }
    }

    private BindImmediateService mBindService;
    private ServiceConnection mFirstConn = new ServiceConnection() {
        //获取服务对象时的操作
        public void onServiceConnected(ComponentName name, IBinder service) {
            //如果服务运行于另一个进程，就不能直接强制转换类型
            //否则会报错 java.lang.ClassCastException: android.os.BinderProxy cannot be cast to...
            mBindService = ((BindImmediateService.LocalBinder) service).getService();
        }

        //无法获取服务对象时的操作
        public void onServiceDisconnected(ComponentName name) {
            mBindService = null;
        }
    };
}

```

6-306-316-30onCreateonBind6-31onUnbind
onDestroy



6-30



6-31

3.

startServicebindService

“ → → ”6-32“ → → ”6-33



6-32



6-33

(1)onBindonRebind(onUnbind)

(2)onUnbind

6.5.2

startForegroundNotification

stopForegroundtruefalse

```
@TargetApi(Build.VERSION_CODES.JELLY_BEAN)
public class MusicService extends Service {
    private final IBinder mBinder = new LocalBinder();
    public class LocalBinder extends Binder {
        public MusicService getService() {
            return MusicService.this;
        }
    }

    @Override
    public IBinder onBind(Intent intent) {
        Log.d(TAG, "onBind");
        return mBinder;
    }

    private String mSong;
    private String PAUSE_EVENT = "";
    private boolean bPlay = true;
    private long mBaseTime;
    private long mPauseTime = 0;
    private int mProcess = 0;
    private Handler mHandler = new Handler();
```

```

private Runnable mPlay = new Runnable() {
    @Override
    public void run() {
        if (bPlay == true) {
            mProcess = (mProcess < 100) ? mProcess+2 : 0;
            mHandler.postDelayed(this, 1000);
        }
        Notification notify = getNotify(MusicService.this, PAUSE_EVENT, mSong, bPlay,
mProcess, mBaseTime);
        startForeground(2, notify);
    }
};

private Notification getNotify(Context ctx, String event, String song, boolean isPlay, int progress, long
time) {
    Intent pIntent = new Intent(event);
    PendingIntent nIntent = PendingIntent.getBroadcast(
        ctx, R.string.app_name, pIntent, PendingIntent.FLAG_UPDATE_CURRENT);
    RemoteViews notify_music = new RemoteViews(ctx.getPackageName(), R.layout.notify_music);
    if (isPlay == true) {
        notify_music.setTextViewText(R.id.btn_play, "暂停");
        notify_music.setTextViewText(R.id.tv_play, song+"正在播放");
        notify_music.setChronometer(R.id.chr_play, time, "%s", true);
    } else {
        notify_music.setTextViewText(R.id.btn_play, "继续");
        notify_music.setTextViewText(R.id.tv_play, song+"暂停播放");
        notify_music.setChronometer(R.id.chr_play, time, "%s", false);
    }
    notify_music.setProgressBar(R.id.pb_play, 100, progress, false);
    notify_music.setOnClickPendingIntent(R.id.btn_play, nIntent);
    Intent intent = new Intent(ctx, MainActivity.class);
    PendingIntent contentIntent = PendingIntent.getActivity(ctx,
        R.string.app_name, intent, PendingIntent.FLAG_UPDATE_CURRENT);
    Notification.Builder builder = new Notification.Builder(ctx);
    builder.setContentIntent(contentIntent).setContent(notify_music)
        .setTicker(song).setSmallIcon(R.drawable.tt_s);
    Notification notify = builder.build();
    return notify;
}

@Override
public int onStartCommand(Intent intent, int flags, int startId) {
    mBaseTime = SystemClock.elapsedRealtime();
}

```

```

        bPlay = intent.getBooleanExtra("is_play", true);
        mSong = intent.getStringExtra("song");
        mHandler.postDelayed(mPlay, 200);
        return START_STICKY;
    }

    @Override
    public void onCreate() {
        PAUSE_EVENT = getResources().getString(R.string.pause_event);
        pauseReceiver = new PauseReceiver();
        IntentFilter filter = new IntentFilter(PAUSE_EVENT);
        registerReceiver(pauseReceiver, filter);
        super.onCreate();
    }

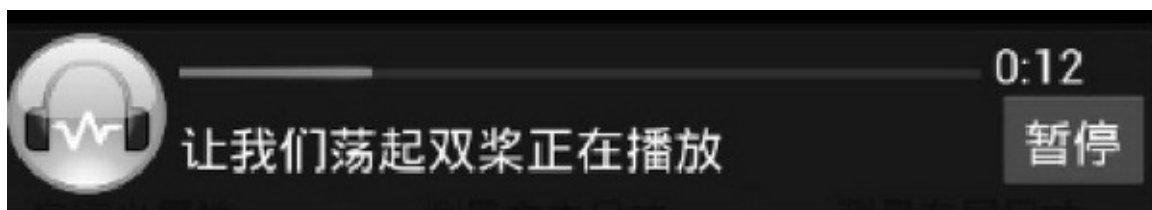
    @Override
    public void onDestroy() {
        unregisterReceiver(pauseReceiver);
        super.onDestroy();
    }

    private PauseReceiver pauseReceiver;
    public class PauseReceiver extends BroadcastReceiver {
        @Override
        public void onReceive(Context context, Intent intent) {
            if (intent != null) {
                bPlay = !bPlay;
                if (bPlay == true) {
                    mHandler.postDelayed(mPlay, 200);
                    if (mPauseTime > 0) {
                        long gap = SystemClock.elapsedRealtime() - mPauseTime;
                        mBaseTime += gap;
                    }
                } else {
                    mPauseTime = SystemClock.elapsedRealtime();
                }
            }
        }
    }
}

```

/getBroadcastgetActivity

6-346-356-346-35





让我们荡起双桨暂停播放

0:20

继续

6-35

6.6

6.6.1

App*****App——100M101G100App
App“”
6-366-37

统计日期 2016年10月14日



本月已用流量待统计

今日已用流量34.3M
超出限额4.3M



浏览器

com.lenovo.browser

19.2M



CSDN

net.csdn.csdnplus

9.6M



微信

com.tencent.mm

2.2M



苏宁易购

com.suning.mobile.e
buy

1.8M



支付宝

com.eg.android.Alip
ayGphone

320.5K



6-3750M

Custom

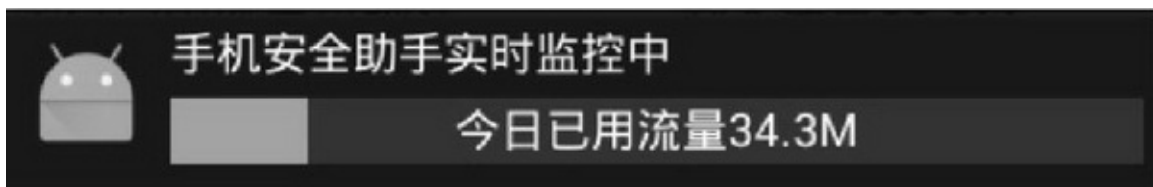
每月限额： M

每日限额： M

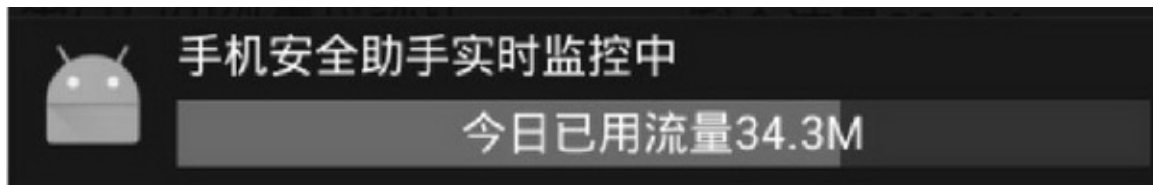
保存

6-38

App6-396-40



6-3930M



6-4050M

4

ServiceAcitivityService

6.6.2 PackageManager PackageManager ApplicationInfo

```
public static ArrayList<AppInfo> getAppInfo(Context ctx, int type) {
    ArrayList<AppInfo> appList = new ArrayList<AppInfo>();
    SparseIntArray siArray = new SparseIntArray();
    PackageManager pm = ctx.getPackageManager();
    List<ApplicationInfo> installList = pm.getInstalledApplications(PackageManager.
PERMISSION_GRANTED);
    for (int i=0; i<installList.size(); i++) {
        ApplicationInfo item = installList.get(i);
        if (siArray.indexOfKey(item.uid) >= 0) { //去掉重复的应用信息
            continue;
        }
        siArray.put(item.uid, 1);
        try {
            String[] permissions = pm.getPackageInfo(item.packageName, PackageManager.
GET_PERMISSIONS).requestedPermissions;
            if (permissions == null) {
                continue;
            }
            boolean bNet = false;
            for (String permission : permissions) {
                if (permission.equals("android.permission.INTERNET")) {
                    bNet = true;
                    break;
                }
            }
            if (type==0 || (type==1 && bNet)) {
                AppInfo app = new AppInfo();
                app.uid = item.uid;
                app.label = item.loadLabel(pm).toString();
                app.package_name = item.packageName;
                app.icon = item.loadIcon(pm);
                appList.add(app);
            }
        } catch (Exception e) {
            e.printStackTrace();
            continue;
        }
    }
    return appList;
}
```

TrafficStats6

getTotalRxBytes

getTotalTxBytes

getMobileRxByteswifi

getMobileTxBytes
getUidRxBytes
getUidTxBytes

6.6.3

55

步骤 01 package7

com.example.assistant.activityAcitivity
com.example.assistant.adapter
com.example.assistant.bean
com.example.assistant.databaseSQLite
com.example.assistant.serviceService
com.example.assistant.util
com.example.assistant.widget

步骤 02 MobileAssistantActivity.java

activity_mobile_assistant.xml MobileConfigActivity.java
activity_mobile_config.xml TrafficService.java

步骤 03 AndroidManifest.xml3

(1)activity

<activity android:name=".MobileAssistantActivity" /> <activity
android:name=".MobileConfigActivity" /> (2)service

<service android:name=".service.TrafficService" android:enabled="true" />

(3)applicationnameMainApplication

android:name=".MainApplication"

步骤 04 XML

(1)res/drawable

(2)res/layout

(3)res/values/styles.xml

步骤 05 java

TrafficService.java

```
@TargetApi(Build.VERSION_CODES.JELLY_BEAN)
public class TrafficService extends Service {
    private static final String TAG = "TrafficService";
    private MainApplication app;
    private int limit_day;
    private int mNowDay;

    @Override
    public int onStartCommand(Intent intent, int flags, int startid) {
        app = MainApplication.getInstance();
        limit_day = Integer.parseInt(SharedUtil.getIntance(this).readShared("limit_day", "30"));
        mHandler.post(mRefresh);
        return START_STICKY;
    }

    private Handler mHandler = new Handler();
    private Runnable mRefresh = new Runnable() {
        @Override
        public void run() {
            refreshData();
            refreshNotify();
            mHandler.postDelayed(this, 10000);
        }
    };

    private void refreshData() {
        mNowDay = Integer.parseInt(DateUtil.getNowDateTime("yyyyMMdd"));
        //获取最新的流量信息
        ArrayList<AppInfo> appinfoList = AppUtil.getAppInfo(this, 1);
        for (int i=0; i<appinfoList.size(); i++) {
            AppInfo item = appinfoList.get(i);
            item.traffic = TrafficStats.getUidRxBytes(item.uid);
            item.month = mNowDay/100;
            item.day = mNowDay;
            appinfoList.set(i, item);
        }
        app.mTrafficHelper.insert(appinfoList);
    }

    private void refreshNotify() {
        String last_date = DateUtil.getAddDate(""+mNowDay, -1);
        ArrayList<AppInfo> lastArray = app.mTrafficHelper.query("day="+last_date);
    }
}
```



```

ArrayList<AppInfo> thisArray = app.mTrafficHelper.query("day="+mNowDay);
long traffic_day = 0;
for (int i=0; i<thisArray.size(); i++) {
    AppInfo item = thisArray.get(i);
    for (int j=0; j<lastArray.size(); j++) {
        if (item.uid == lastArray.get(j).uid) {
            item.traffic -= lastArray.get(j).traffic;
            break;
        }
    }
    traffic_day += item.traffic;
}
String desc = "今日已用流量" + StringUtil.formatTraffic(traffic_day);
int progress = 0;
int layoutId = R.layout.notify_traffic_green;
float trafficM = traffic_day/1024.0f/1024.0f;
if (trafficM > limit_day*2) {
    progress = (int) ((trafficM>limit_day*3)?100:(trafficM-limit_day*2)*100/limit_day);
    layoutId = R.layout.notify_traffic_red;
} else if (trafficM > limit_day) {
    progress = (int) ((trafficM>limit_day*2)?100:(trafficM-limit_day)*100/limit_day);
    layoutId = R.layout.notify_traffic_yellow;
} else {
    progress = (int) (trafficM*100/limit_day);
}
Log.d(TAG, "progress="+progress);
RemoteViews notify_traffic = new RemoteViews(this.getPackageName(), layoutId);
notify_traffic.setTextViewText(R.id.tv_flow, desc);
notify_traffic.setProgress(R.id.pb_flow, 100, progress, false);
Intent intent = new Intent(this, MainActivity.class);
PendingIntent contentIntent = PendingIntent.getActivity(this,
    R.string.app_name, intent, PendingIntent.FLAG_UPDATE_CURRENT);
Notification.Builder builder = new Notification.Builder(this);
builder.setContentIntent(contentIntent).setContent(notify_traffic)
    .setTicker("手机安全助手运行中").setSmallIcon(R.drawable.ic_app);
mNotify = builder.build();
startForeground(9, mNotify);
}

private Notification mNotify;
@Override
public void onDestroy() {
    super.onDestroy();
    if (mNotify != null) {

```

```

        stopForeground(true);
    }
}

@Override
public IBinder onBind(Intent intent) {
    return null;
}
}

```

6.7

App0000

Service(3)“”App

4

(1)

(2)

(3)

(4)Service/

第7章 组合控件



App3“”

7.1

3TabActivityActivityGroupFragmentActivity

7.1.1

ButtonImageButtonRadioButtonCheckBoxSwitch

+7-14



7-1

AppQQAndroid

2SwitchiOS,backgrounddrawableTop

textColorStateDrawable

步骤 01 XML

```
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:state_selected="true" android:color="@color/tab_text_selected" />
    <item android:color="@color/tab_text_normal" />
</selector>
```

步骤

02

TextViewbackgroundtextColordrawableTop

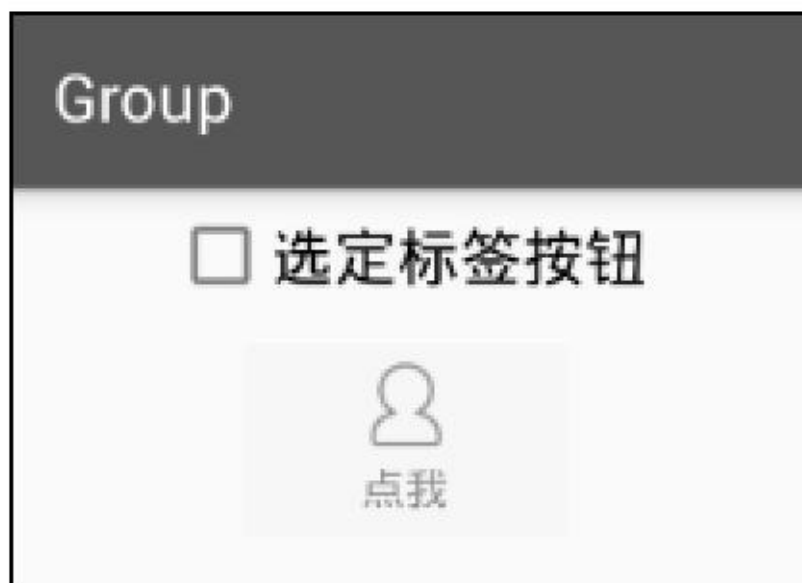
```
<TextView
    android:id="@+id/tv_tab_button"
    android:layout_width="100dp"
    android:layout_height="60dp"
    android:padding="5dp"
    android:layout_gravity="center"
    android:gravity="center"
    android:background="@drawable/tab_bg_selector"
    android:text="点我"
    android:textSize="12sp"
    android:textColor="@drawable/tab_text_selector"
    android:drawableTop="@drawable/tab_first_selector" />
```

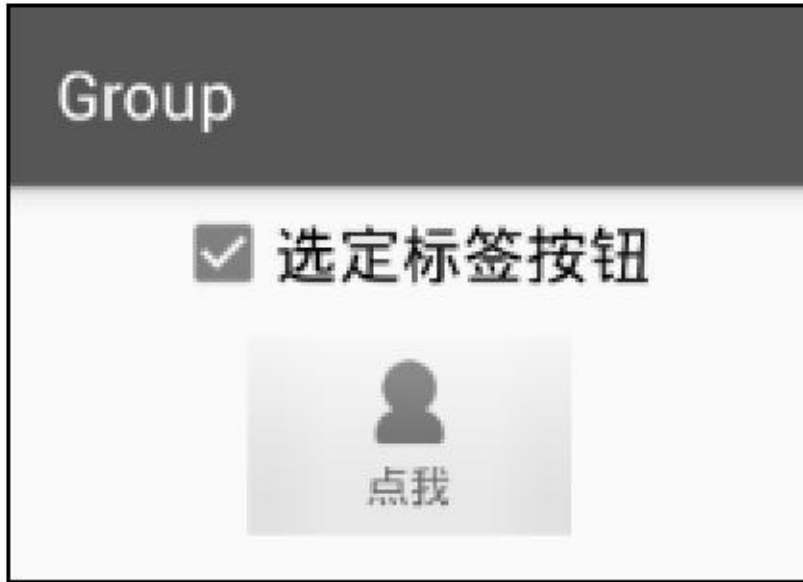
步骤

03

TextViewsetSelected(true)

setSelected(false)7-27-37-27-3





7-3

45values/styles.xmlTabButton

```
<style name="TabButton">
    <item name="android:layout_width">match_parent</item>
    <item name="android:layout_height">match_parent</item>
    <item name="android:padding">5dp</item>
    <item name="android:layout_gravity">center</item>
    <item name="android:gravity">center</item>
    <item name="android:background">@drawable/tab_bg_selector</item>
    <item name="android:textSize">12sp</item>
    <item name="android:textStyle">normal</item>
    <item name="android:textColor">@drawable/tab_text_selector</item>
</style>
```

TextViewstyle="@style/TabButton"styles.xml

7.1.2

3TabActivityActivityGroupFragmentActivity3

1.TabActivity

TabActivityTabHostTabWidget

TabActivity

```
<TabHost xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@android:id/tabhost"
    android:layout_width="match_parent"
    android:layout_height="match_parent" >

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent" >

        <FrameLayout
            android:id="@android:id/tabcontent"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_marginBottom="@dimen/tabbar_height" />

        <TabWidget
            android:id="@android:id/tabs"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:visibility="gone" />

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="@dimen/tabbar_height"
            android:layout_alignParentBottom="true"
            android:gravity="bottom"
            android:orientation="horizontal" >

            <LinearLayout
                android:id="@+id/ll_first"
                android:layout_width="0dp"
                android:layout_height="match_parent"
                android:layout_weight="1"
                android:orientation="vertical" >
```

```

        <TextView
            style="@style/TabButton"
            android:drawableTop="@drawable/tab_first_selector"
            android:text="@string/menu_first" />
    </LinearLayout>

    <LinearLayout
        android:id="@+id/ll_second"
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:orientation="vertical" >

        <TextView
            style="@style/TabButton"
            android:drawableTop="@drawable/tab_second_selector"
            android:text="@string/menu_second" />
    </LinearLayout>

    <LinearLayout
        android:id="@+id/ll_third"
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:orientation="vertical" >

        <TextView
            style="@style/TabButton"
            android:drawableTop="@drawable/tab_third_selector"
            android:text="@string/menu_third" />
    </LinearLayout>
</LinearLayout>
</RelativeLayout>
</TabHost>

```

ActivityTabActivity

```

public class TabHostActivity extends TabActivity implements OnClickListener {
    private static final String TAG = "TabHostActivity";
    private Bundle mBundle = new Bundle();
    private TabHost tab_host;
    private LinearLayout ll_first, ll_second, ll_third;
    private String FIRST_TAG = "first";

```



```

private String SECOND_TAG = "second";
private String THIRD_TAG = "third";

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_tab_host);
    mBundle.putString("tag", TAG);
    ll_first = (LinearLayout) findViewById(R.id.ll_first);
    ll_second = (LinearLayout) findViewById(R.id.ll_second);
    ll_third = (LinearLayout) findViewById(R.id.ll_third);
    ll_first.setOnClickListener(this);
    ll_second.setOnClickListener(this);
    ll_third.setOnClickListener(this);
    tab_host = getTabHost();
    tab_host.addTab(getNewTab(FIRST_TAG, R.string.menu_first,
        R.drawable.tab_first_selector, TabFirstActivity.class));
    tab_host.addTab(getNewTab(SECOND_TAG, R.string.menu_second,
        R.drawable.tab_second_selector, TabSecondActivity.class));
    tab_host.addTab(getNewTab(THIRD_TAG, R.string.menu_third,
        R.drawable.tab_third_selector, TabThirdActivity.class));
    tab_host.setCurrentTabByTag(FIRST_TAG);
    changeContainerView(ll_first);
}

private TabHost.TabSpec getNewTab(String spec, int label, int icon, Class<?> cls) {
    Intent intent = new Intent(this, cls).putExtras(mBundle);
    return tab_host.newTabSpec(spec).setContent(intent)
        .setIndicator(getString(label), getResources().getDrawable(icon));
}

@Override
public void onClick(View v) {
    if (v.getId()==R.id.ll_first || v.getId()==R.id.ll_second || v.getId()==R.id.ll_third) {
        changeContainerView(v);
    }
}

private void changeContainerView(View v) {
    ll_first.setSelected(false);
    ll_second.setSelected(false);
    ll_third.setSelected(false);
    v.setSelected(true);
}

```

```

    if (v == ll_first) {
        tab_host.setCurrentTabByTag(FIRST_TAG);
    } else if (v == ll_second) {
        tab_host.setCurrentTabByTag(SECOND_TAG);
    } else if (v == ll_third) {
        tab_host.setCurrentTabByTag(THIRD_TAG);
    }
}
}

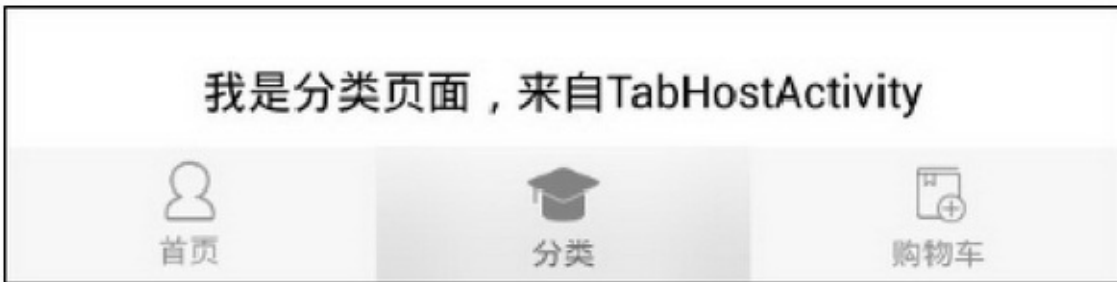
```

getNewTabTabHostsetCurrentTabByTag

7-47-57-4“”7-5“”



7-4“”



7-5“”

2.ActivityGroup

ActivityGroupActivityActivityGroupActivityActivity

FragmentActivityGroupActivityGroup

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <LinearLayout
        android:id="@+id/ll_container"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:gravity="bottom|center"
        android:orientation="horizontal" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="@dimen/tabbar_height"
        android:orientation="horizontal" >
```

```

<LinearLayout
    android:id="@+id/ll_first"
    android:layout_width="0dp"
    android:layout_height="match_parent"
    android:layout_weight="1"
    android:orientation="vertical" >

    <TextView
        style="@style/TabButton"
        android:drawableTop="@drawable/tab_first_selector"
        android:text="@string/menu_first" />
</LinearLayout>

<LinearLayout
    android:id="@+id/ll_second"
    android:layout_width="0dp"
    android:layout_height="match_parent"
    android:layout_weight="1"
    android:orientation="vertical" >

    <TextView
        style="@style/TabButton"
        android:drawableTop="@drawable/tab_second_selector"
        android:text="@string/menu_second" />
</LinearLayout>

<LinearLayout
    android:id="@+id/ll_third"
    android:layout_width="0dp"
    android:layout_height="match_parent"
    android:layout_weight="1"
    android:orientation="vertical" >

    <TextView
        style="@style/TabButton"
        android:drawableTop="@drawable/tab_third_selector"
        android:text="@string/menu_third" />
</LinearLayout>
</LinearLayout>
</LinearLayout>

```

ActivityGroup

```

public class TabGroupActivity extends ActivityGroup implements OnClickListener {
    private static final String TAG = "TabGroupActivity";
    private Bundle mBundle = new Bundle();
    private LinearLayout ll_container, ll_first, ll_second, ll_third;

```

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_tab_group);
    ll_container = (LinearLayout) findViewById(R.id.ll_container);
    ll_first = (LinearLayout) findViewById(R.id.ll_first);
    ll_second = (LinearLayout) findViewById(R.id.ll_second);
    ll_third = (LinearLayout) findViewById(R.id.ll_third);
    ll_first.setOnClickListener(this);
    ll_second.setOnClickListener(this);
    ll_third.setOnClickListener(this);
    mBundle.putString("tag", TAG);
    changeContainerView(ll_first);
}

@Override
public void onClick(View v) {
    if (v.getId()==R.id.ll_first || v.getId()==R.id.ll_second || v.getId()==R.id.ll_third) {
        changeContainerView(v);
    }
}

private void changeContainerView(View v) {
    ll_first.setSelected(false);
    ll_second.setSelected(false);
    ll_third.setSelected(false);
    v.setSelected(true);
    if (v == ll_first) {
        toActivity("first", TabFirstActivity.class);
    } else if (v == ll_second) {
        toActivity("second", TabSecondActivity.class);
    } else if (v == ll_third) {
        toActivity("third", TabThirdActivity.class);
    }
}

private void toActivity(String label, Class<?> cls) {
    Intent intent = new Intent(this, cls).putExtras(mBundle);
    ll_container.removeAllViews();
    View v = getLocalActivityManager().startActivity(label, intent).getDecorView();
    v.setLayoutParams(new LayoutParams(
        LayoutParams.MATCH_PARENT, LayoutParams.MATCH_PARENT));
}

```

```

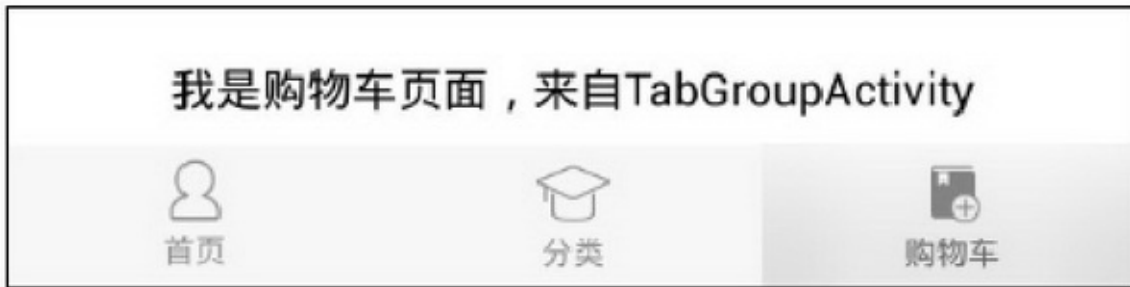
        ll_container.addView(v);
    }
}

```

toActivitystartActivityWindowWindow(
getDecorView)DecorViewDecorViewActivityGroupstartActivity
getLocalActivityManagergetLocalActivityManagerActivityGroup
TabActivity7-67-77-6“”7-7“”



7-6“”



7-7“”

3.FragmentActivity
ActivityGroupActivityGroupActivityActivityFragment
FragmentActivity3FragmentActivity

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <!-- 把 FragmentLayout 放在 FragmentTabHost 上面，标签页就在页面底部；
        反之 FragmentLayout 在 FragmentTabHost 下面，标签页就在页面顶部。 -->

    <FrameLayout
        android:id="@+id/realtabcontent"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1" />

    <android.support.v4.app.FragmentTabHost
        android:id="@android:id/tabhost"
        android:layout_width="match_parent"
        android:layout_height="@dimen/tabbar_height" >

        <FrameLayout
            android:id="@android:id/tabcontent"
            android:layout_width="0dp"
            android:layout_height="0dp"
            android:layout_weight="0" />

    </android.support.v4.app.FragmentTabHost>
</LinearLayout>
```

```

public class TabFragmentActivity extends FragmentActivity {
    private static final String TAG = "TabFragmentActivity";
    private FragmentTabHost mTabHost;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_tab_fragment);
        Bundle bundle = new Bundle();
        bundle.putString("tag", TAG);
        mTabHost = (FragmentTabHost) findViewById(android.R.id.tabhost);
        mTabHost.setup(this, getSupportFragmentManager(), R.id.realtabcontent);
        //addTab(标题, 跳转的 Fragment, 传递参数的 Bundle)
        mTabHost.addTab(getTabView(R.string.menu_first, R.drawable.tab_first_selector),
            TabFirstFragment.class, bundle);
        mTabHost.addTab(getTabView(R.string.menu_second, R.drawable.tab_second_selector),
            TabSecondFragment.class, bundle);
        mTabHost.addTab(getTabView(R.string.menu_third, R.drawable.tab_third_selector),
            TabThirdFragment.class, bundle);
        //设置 tabs 之间的分隔线不显示
        mTabHost.getTabWidget().setShowDividers(LinearLayout.SHOW_DIVIDER_NONE);
    }

    private TabSpec getTabView(int textId, int imgId) {
        String text = getResources().getString(textId);
        Drawable drawable = getResources().getDrawable(imgId);
        //必须设置图片大小, 否则不显示
        drawable.setBounds(0, 0, drawable.getMinimumWidth(), drawable.getMinimumHeight());
        //R.layout.item_tabbar 是单个标签按钮的布局文件
        View item_tabbar = getLayoutInflater().inflate(R.layout.item_tabbar, null);
        TextView tv_item = (TextView) item_tabbar.findViewById(R.id.tv_item_tabbar);
        tv_item.setText(text);
        tv_item.setCompoundDrawables(null, drawable, null, null);
        TabSpec spec = mTabHost.newTabSpec(text).setIndicator(item_tabbar);
        return spec;
    }
}

```

FragmentActivityaddTabFragmentFragmentTabHostsetSelected
 FragmentActivity
 FragmentActivity7-87-97-8“”7-9“”



7-8“”



7-9“”

7.2

ToolBarOverflowMenuSearchViewTabLayoutToolBar
TabLayout

7.2.1 Toolbar AppAndroid5.0ActionBarActionBar Android5.0ToolBarActionBar

ActionBarToolBarActionBarToolBarActionBar

步骤 01

styles.xmlActionBar

```
<style name="AppCompatActivity"
```

```
parent="Theme.AppCompat.Light.NoActionBar" />
```

AndroidManifest.xmlactivityandroid:theme

```
android:theme="@style/AppCompatActivity"
```

步骤 03

LinearLayoutverticalToolBarToolBar

ViewGroup

```
<android.support.v7.widget.Toolbar  
    android:id="@+id/tl_head"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content" />
```

步骤 04

ActivityAppCompatActivityAndroid Studio

AppCompatActivityonCreateToolBarsetSupportActionBarToolBar

ToolBarActionBarToolBarToolBar7-1(6“6.1.1”)

XML 中的属性	Toolbar 类的设置方法	说明
logo	setLogo	设置工具栏图标
title	setTitle	设置标题文字
titleTextColor	setTitleTextColor	设置标题的文字颜色
titleTextAppearance	setTitleTextAppearance	设置标题的文字风格。风格定义在 styles.xml 中
subtitle	setSubtitle	设置副标题文字。副标题在标题下方
subtitleTextColor	setSubtitleTextColor	设置副标题的文字颜色
subtitleTextAppearance	setSubtitleTextAppearance	设置副标题的文字风格
navigationIcon	setNavigationIcon	设置左侧导航图标
无	setNavigationOnClickListener	设置导航图标的点击监听器

Toolbar

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_toolbar);
    Toolbar tl_head = (Toolbar) findViewById(R.id.tl_head);
    tl_head.setNavigationIcon(R.drawable.ic_back);
    tl_head.setTitle("工具栏页面");
    tl_head.setTitleTextColor(Color.RED);
    tl_head.setLogo(R.drawable.ic_app);
    tl_head.setSubtitle("Toolbar");
    tl_head.setSubtitleTextColor(Color.YELLOW);
    tl_head.setBackgroundResource(R.color.blue_light);
    setSupportActionBar(tl_head);
    //setNavigationOnClickListener 必须放到 setSupportActionBar 之后，不然不起作用
    tl_head.setNavigationOnClickListener(new OnClickListener() {
        @Override
        public void onClick(View view) {
            finish();
        }
    });
}

```



7-10

7.2.2 OverflowMenu OverflowMenu

OptionsMenu showAsAction 7-2

7-2

展示位置类型	说明
always	总是在导航栏上显示菜单图标
ifRoom	如果导航栏右侧有空间，该项就直接显示在导航栏上，不再放入溢出菜单
never	从不在导航栏上直接显示，一直放在溢出菜单列表里面
withText	如果能在导航栏上显示，除了显示图标，还要显示该项的文字说明
collapseActionView	操作视图要折叠为一个按钮，点击该按钮再展开操作视图，主要用于 SearchView

iconMenuBuilders setOptionalIconsVisible

```

public static void setOverflowIconVisible(int featureId, Menu menu) {
    // ActionBar 的 featureId 是 8，Toolbar 的 featureId 是 108
    if (featureId % 100 == Window.FEATURE_ACTION_BAR && menu != null) {
        if (menu.getClass().getSimpleName().equals("MenuBuilder")) {
            try {
                Method m = menu.getClass().getDeclaredMethod(
                    "setOptionalIconsVisible", Boolean.TYPE);
                m.setAccessible(true);
                m.invoke(menu, true);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    }
}

```

showAsAction ifRoom always ActionBar Toolbar

menuxmlns:app="http://schemas.android.com/apk/res-auto"
android:showAsAction="ifRoom"app:showAsAction="ifRoom"

```
<menu xmlns:android="http://schemas.android.com/apk/res/android"
      xmlns:app="http://schemas.android.com/apk/res-auto" >

  <item
    android:id="@+id/menu_refresh"
    android:orderInCategory="1"
    android:icon="@drawable/ic_refresh"
    app:showAsAction="ifRoom"
    android:title="刷新"/>

  <item
    android:id="@+id/menu_about"
    android:orderInCategory="8"
    android:icon="@drawable/ic_about"
    app:showAsAction="never"
    android:title="关于"/>

  <item
    android:id="@+id/menu_quit"
    android:orderInCategory="9"
    android:icon="@drawable/ic_quit"
    app:showAsAction="never"
    android:title="退出"/>

</menu>
```

```

@Override
public boolean onOptionsItemSelected(int featureId, Menu menu) {
    Utils.setOverflowIconVisible(featureId, menu); // 显示菜单项左侧的图标
    return super.onOptionsItemSelected(featureId, menu);
}

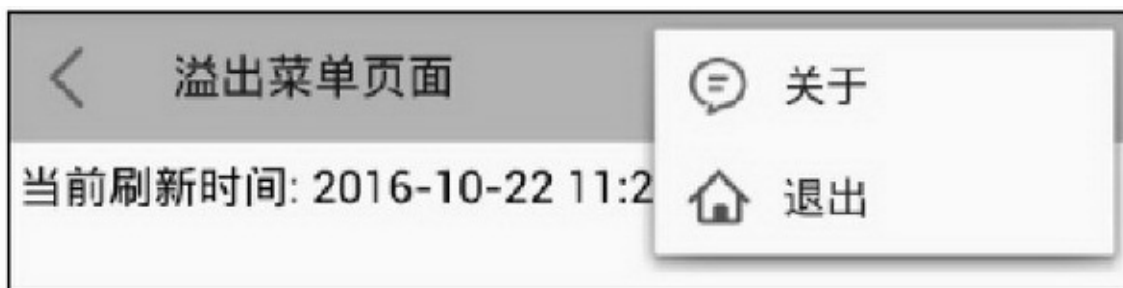
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.menu_overflow, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    int id = item.getItemId();
    if (id == android.R.id.home) { //该分支响应导航图标的点击动作
        finish();
    } else if (id == R.id.menu_refresh) {
        tv_desc.setText("当前刷新时间: "+Utils.getNowDateTime("yyyy-MM-dd HH:mm:ss"));
        return true;
    } else if (id == R.id.menu_about) {
        Toast.makeText(this, "这个是工具栏的演示 demo", Toast.LENGTH_LONG).show();
        return true;
    } else if (id == R.id.menu_quit) {
        finish();
    }
    return super.onOptionsItemSelected(item);
}
}

```

7-117-127-117-12





7-12

7.2.3 SearchView App

步骤 01

```
<item
    android:id="@+id/menu_search"
    android:orderInCategory="1"
    android:icon="@drawable/ic_search"
    app:showAsAction="ifRoom"
    android:title="搜索"
    app:actionViewClass="android.support.v7.widget.SearchView" />
```

步骤 02

res/xml/searchable.xml

```
<searchable xmlns:android="http://schemas.android.com/apk/res/android"
    android:label="@string/app_name"
    android:hint="@string/please_input"
    android:inputType="text"
    android:searchButtonText="@string/search" />
```

步骤 03

AndroidManifest.xml activityactionmeta-data

```
<activity android:name=".SearchResultActivity" android:theme="@style/AppCompatTheme" >
    <intent-filter>
        <action android:name="android.intent.action.SEARCH"/>
    </intent-filter>
    <meta-data android:name="android.app.searchable" android:resource="@xml/searchable"/>
</activity>
```

步骤

04

ActivityActivitySearchResultActivity

```
private void initView(Menu menu) {
    MenuItem menuItem = menu.findItem(R.id.menu_search);
    SearchView searchView = (SearchView) MenuItemCompat.getActionView(menuItem);
    if (searchView == null) {
        Log.d(TAG, "Fail to get SearchView.");
    } else {
        if (getIntent() != null) {
            //设置是否将搜索视图默认折叠为图标
            searchView.setIconifiedByDefault(getIntent().getBooleanExtra("collapse", true));
        } else {
            searchView.setIconifiedByDefault(true);
        }
        //设置是否启用完成图标
        searchView.setSubmitButtonEnabled(true);
        SearchManager searchManager = (SearchManager) getSystemService(Context.SEARCH_
SERVICE);

        ComponentName cn = new ComponentName(this, SearchResultActivity.class);
        SearchableInfo info = searchManager.getSearchableInfo(cn);
        if (info == null) {
            Log.d(TAG, "Fail to get SearchResultActivity.");
        }
        //设置搜索动作的定义
        searchView.setSearchableInfo(info);
        sac_key = (SearchView.SearchAutoComplete) searchView.findViewById(R.id.search_src_text);
        sac_key.setTextColor(Color.WHITE);
        sac_key.setHintTextColor(Color.WHITE);
        //设置搜索关键字的监听器，如是否展示关键字的匹配列表
        searchView.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
```

```

        @Override
        public boolean onQueryTextSubmit(String query) {
            return false;
        }

        @Override
        public boolean onQueryTextChange(String newText) {
            doSearch(newText);
            return true;
        }
    });
    Bundle bundle = new Bundle();
    bundle.putString("hi", "hello");
    searchView.setAppSearchData(bundle);
}

private SearchView.SearchAutoComplete sac_key;
private String[] hintArray = { "iphone", "iphone7s", "iphone7", "iphone7 plus", "iphone6s", "iphone6",
"iphone6 plus"};
private void doSearch(String text) {
    if (text.indexOf("i") == 0) {
        ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,
            R.layout.search_list_auto, hintArray);
        sac_key.setAdapter(adapter);
        sac_key.setOnItemClickListener(new OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
                TextView tv_item = (TextView) view;
                sac_key.setText(tv_item.getText());
            }
        });
    }
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.menu_search, menu);
    //对搜索框做初始化
    initSearchView(menu);
    return true;
}
}

```

步骤 05 Activity


```
private void doSearchQuery(Intent intent) {
    if (intent == null) {
        return;
    } else {
        // 如果通过 ACTION_SEARCH 调用，即通过搜索调用
        if (Intent.ACTION_SEARCH.equals(intent.getAction())) {
            // 获取额外信息
            Bundle bundle = intent.getBundleExtra(SearchManager.APP_DATA);
            String value = bundle.getString("hi");
            // 获取搜索内容
            String queryString = intent.getStringExtra(SearchManager.QUERY);
            tv_search_result.setText("您输入的搜索文字是: "+queryString+", 额外信息: "+value);
        }
    }
}
```

7-13~7-167-137-147-157-16



7-13



7-14



7-15



7-16

7.2.4 TabLayout Toolbar ActionBar6“”
LinearLayout LinearLayout Toolbar
Toolbar LinearLayout

```

<android.support.v7.widget.Toolbar
    android:id="@+id/tl_head"
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:background="@color/blue_light"
    app:navigationIcon="@drawable/ic_back" >

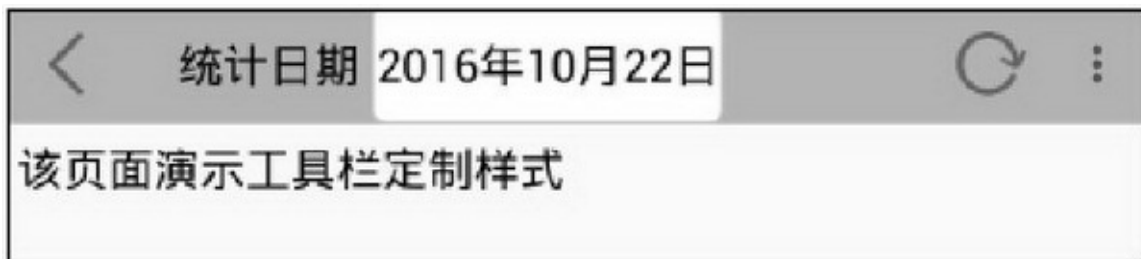
    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content" >

        <TextView
            android:id="@+id/tv_day"
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:layout_centerInParent="true"
            android:background="@drawable/edittext_selector"
            android:gravity="center"
            android:textColor="@color/black"
            android:textSize="17sp" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:layout_toLeftOf="@+id/tv_day"
            android:gravity="center"
            android:text="统计日期 "
            android:textColor="@color/black"
            android:textSize="17sp" />

    </RelativeLayout>
</android.support.v7.widget.Toolbar>

```



7-17

7-17

ToolbarApp7-187-19PagerTabStrip

```
designTabLayoutbuild.gradledependenciesdesign  
compile 'com.android.support:design:25.1.0'
```





7-19

TabLayoutPagerTabStripTabLayout6

tabBackground

tabIndicatorColor

tabIndicatorHeight

tabTextColor

tabTextAppearance

tabSelectedTextColor

XMLTabLayout

```

<android.support.v7.widget.Toolbar
    android:id="@+id/tl_head"
    android:layout_width="match_parent"
    android:layout_height="50dp"
    app:navigationIcon="@drawable/ic_back" >

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content" >

        <android.support.design.widget.TabLayout
            android:id="@+id/tab_title"
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:layout_centerInParent="true"
            app:tabIndicatorColor="@color/red"
            app:tabIndicatorHeight="2dp"
            app:tabSelectedTextColor="@color/red"
            app:tabTextColor="@color/grey"
            app:tabTextAppearance="@style/TabText" />

    </RelativeLayout>
</android.support.v7.widget.Toolbar>

```

TabLayout4

newTab

addTab

getTabAt



onTabSelected



onTabUnselected



onTabReselected

setOnTabSelectedListenerOnTabSelectedListener3

TabLayoutViewPager

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_tab_layout);
    tl_head = (ToolBar) findViewById(R.id.tl_head);
    tab_title = (TabLayout) findViewById(R.id.tab_title);
    vp_content = (ViewPager) findViewById(R.id.vp_content);
    setSupportActionBar(tl_head);
    mTitleArray.add("商品");
    mTitleArray.add("详情");
    initTabLayout();
    initTabViewPager();
}

private void initTabLayout() {
    tab_title.addTab(tab_title.newTab().setText(mTitleArray.get(0)));
    tab_title.addTab(tab_title.newTab().setText(mTitleArray.get(1)));
    tab_title.setOnTabSelectedListener(this);
}

private void initTabViewPager() {
    GoodsPagerAdapter adapter = new GoodsPagerAdapter(getSupportFragmentManager(),
mTitleArray);
    vp_content.setAdapter(adapter);
    vp_content.addOnPageChangeListener(new SimpleOnPageChangeListener() {
        @Override
        public void onPageSelected(int position) {
            tab_title.getTabAt(position).select();
        }
    });
}

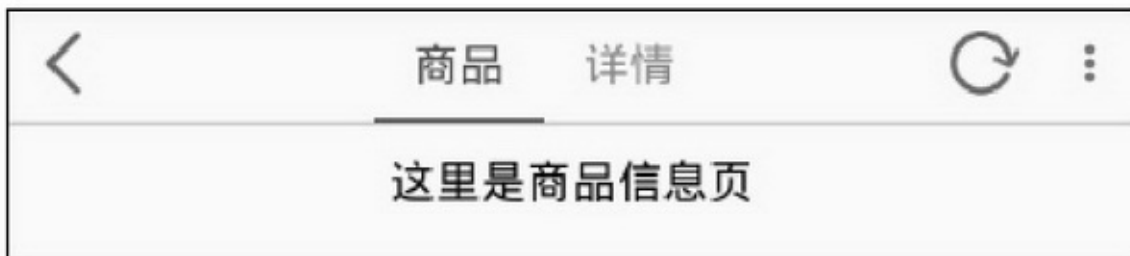
@Override
public void onTabReselected(Tab tab) {
}

@Override
public void onTabSelected(Tab tab) {
    vp_content.setCurrentItem(tab.getPosition());
}

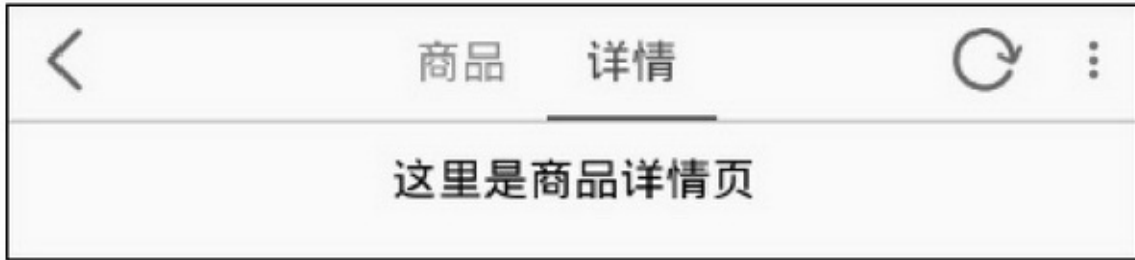
@Override
public void onTabUnselected(Tab tab) {
}

```

“7-20”7-21App



7-20“



7-21“”

TabLayoutTextViewtextColorImageViewsrc

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent" >

    <TextView
        android:id="@+id/tv_toolbar1"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_centerInParent="true"
        android:gravity="center"
        android:textColor="@drawable/toolbar_text_selector"
        android:textSize="17sp" />

    <ImageView
        android:id="@+id/iv_point1"
        android:layout_width="25dp"
        android:layout_height="25dp"
        android:layout_toRightOf="@+id/tv_toolbar1"
        android:paddingTop="10dp"
        android:paddingLeft="3dp"
        android:scaleType="fitCenter"
        android:src="@drawable/toolbar_image_selector" />

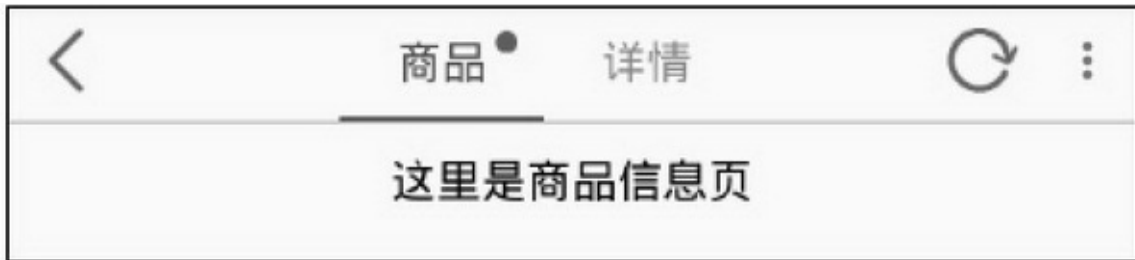
</RelativeLayout>
```

initTabLayoutsetCustomView

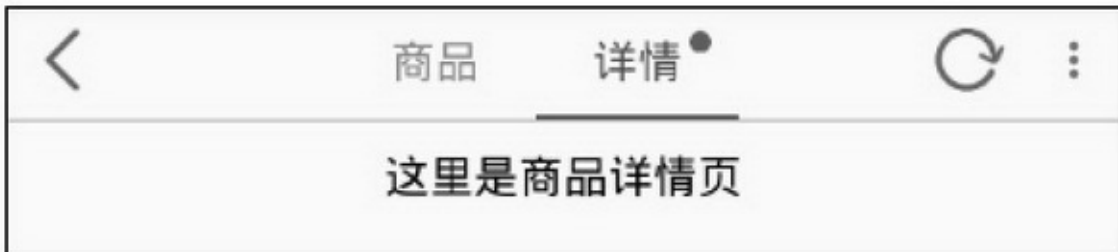

```
private void initTabLayout() {
    tab_title.addTab(tab_title.newTab().setCustomView(R.layout.item_toolbar1));
    tv_toolbar1 = (TextView) findViewById(R.id.tv_toolbar1);
    tv_toolbar1.setText(mTitleArray.get(0));
    tab_title.addTab(tab_title.newTab().setCustomView(R.layout.item_toolbar2));
    tv_toolbar2 = (TextView) findViewById(R.id.tv_toolbar2);
    tv_toolbar2.setText(mTitleArray.get(1));
    tab_title.setOnTabSelectedListener(new ViewPagerOnTabSelectedListener(vp_content));
}
```

App7-227-237-22“”“”7-23“”

“”



7-22“”



7-23“”

7.3

BannerBanner

7.3.1

ViewPagerFragment

7-24



7-24

ViewPagerOnPageChangeListener5onPageScrollStateChanged
onPageScrolledonPageSelected

1.onPageSelected

onPageScrollStateChangedonPageScrolled

OnPageChangeListenerSimpleOnPageChangeListenerViewPager
onPageSelected

```

mPager.addOnPageChangeListener(new SimpleOnPageChangeListener() {
    @Override
    public void onPageSelected(int position) {
        setButton(position);
    }
});

```

2.onPageSelectedonPageScrollStateChangedonPageScrolled onPageScrolled6

```

public class PagerIndicator extends LinearLayout {
    private Context mContext;
    private int mCount = 5;
    private int mPad = 30;
    private int mSeq = 0;
    private float mRatio = 0.0f;
    private Paint mPaint;
    private Bitmap mBackImage;
    private Bitmap mForeImage;

    public PagerIndicator(Context context) {
        this(context, null);
    }

    public PagerIndicator(Context context, AttributeSet attrs) {
        super(context, attrs);
        mContext = context;
        init();
    }

    private void init() {
        mPaint = new Paint();
        mBackImage = BitmapFactory.decodeResource(getResources(), R.drawable.icon_point_n);
        mForeImage = BitmapFactory.decodeResource(getResources(), R.drawable.icon_point_c);
    }

    @Override
    protected void dispatchDraw(Canvas canvas) {
        super.dispatchDraw(canvas);
        int left = (getMeasuredWidth() - mCount*mPad) / 2;
        for (int i=0; i<mCount; i++) {
            canvas.drawBitmap(mBackImage, left+i*mPad, 0, mPaint);
        }
        canvas.drawBitmap(mForeImage, left+(mSeq+mRatio)*mPad, 0, mPaint);
    }

    public void setCount(int count, int pad) {
        mCount = count;
        mPad = pad;
        invalidate();
    }
}

```

```

    }

    public void setCurrent(int seq, float ratio) {
        mSeq = seq;
        mRatio = ratio;
        invalidate();
    }
}

```

OnPageChangeListener onPageScrolled setCurrent 7-24

```

private class BannerChangeListener implements ViewPager.OnPageChangeListener {
    @Override
    public void onPageScrollStateChanged(int arg0) {
    }

    @Override
    public void onPageScrolled(int seq, float ratio, int offset) {
        mIndicator.setCurrent(seq, ratio);
    }

    @Override
    public void onPageSelected(int seq) {
        mIndicator.setCurrent(seq, 0);
    }
}

```

7.3.2 Banner ViewPager App Banner

6 Handler + Runnable Banner Handler Runnable

```

public class BannerPager extends RelativeLayout implements View.OnClickListener {
    private static final String TAG = "BannerPager";
    private Context mContext;
    private ViewPager mPager;
    private List<ImageView> mViewList = new ArrayList<ImageView>();
    private RadioGroup mGroup;
    private int mCount;
    private LayoutInflater mInflater;
}

```

```

private int dip_15;
private static int mInterval = 2000;

public BannerPager(Context context) {
    this(context, null);
}

public BannerPager(Context context, AttributeSet attrs) {
    super(context, attrs);
    mContext = context;
    init();
}

public void start() {
    mHandler.postDelayed(mScroll, mInterval);
}

public void stop() {
    mHandler.removeCallbacks(mScroll);
}

public void setInterval(int interval) {
    mInterval = interval;
}

public void setImage(ArrayList<Integer> imageList) {
    for (int i = 0; i < imageList.size(); i++) {
        Integer imageID = ((Integer) imageList.get(i)).intValue();
        ImageView iv = new ImageView(mContext);
        iv.setLayoutParams(new LayoutParams(
            LayoutParams.MATCH_PARENT, LayoutParams.MATCH_PARENT));
        iv.setScaleType(ImageView.ScaleType.FIT_XY);
        iv.setImageResource(imageID);
        iv.setOnClickListener(this);
        mViewList.add(iv);
    }
    mPager.setAdapter(new ImageAdapater());
    mPager.addOnPageChangeListener(new SimpleOnPageChangeListener() {
        @Override
        public void onPageSelected(int position) {
            setButton(position);
        }
    });
}

```

```

mCount = imageList.size();
for (int i = 0; i < mCount; i++) {
    RadioButton radio = new RadioButton(mContext);
    radio.setLayoutParams(new RadioGroup.LayoutParams(dip_15, dip_15));
    radio.setGravity(Gravity.CENTER);
    radio.setButtonDrawable(R.drawable.indicator_selector);
    mGroup.addView(radio);
}
mPager.setCurrentItem(0);
setButton(0);
}

private void setButton(int position) {
    ((RadioButton) mGroup.getChildAt(position)).setChecked(true);
}

private void init() {
    mInflater = ((Activity) mContext).getLayoutInflater();
    View view = mInflater.inflate(R.layout.banner_pager, null);
    mPager = (ViewPager) view.findViewById(R.id.vp_banner);
    mGroup = (RadioGroup) view.findViewById(R.id.rg_indicator);
    addView(view);
    dip_15 = Utils.dip2px(mContext, 15);
}

private Handler mHandler = new Handler();
private Runnable mScroll = new Runnable() {
    @Override
    public void run() {
        scrollToNext();
        mHandler.postDelayed(this, mInterval);
    }
};

public void scrollToNext() {
    int index = mPager.getCurrentItem() + 1;
    if (mViewList.size() <= index) {
        index = 0;
    }
    mPager.setCurrentItem(index);
}
}

```

```

private class ImageAdapater extends PagerAdapter {
    @Override
    public int getCount() {
        return mViewList.size();
    }

    @Override
    public boolean isViewFromObject(View arg0, Object arg1) {
        return arg0 == arg1;
    }

    @Override
    public void destroyItem(ViewGroup container, int position, Object object) {
        container.removeView(mViewList.get(position));
    }

    @Override
    public Object instantiateItem(ViewGroup container, int position) {
        container.addView(mViewList.get(position));
        return mViewList.get(position);
    }
}

@Override
public void onClick(View v) {
    int position = mPager.getCurrentItem();
    mListener.onBannerClick(position);
}

public void setOnBannerListener(BannerClickListener listener) {
    mListener = listener;
}

private BannerClickListener mListener;
public interface BannerClickListener {
    public void onBannerClick(int position);
}
}

```

ActivityBannersetImagestart

```

mBanner = (BannerPager) findViewById(R.id.banner_pager);
LayoutParams params = (LayoutParams) mBanner.getLayoutParams();
params.height = (int) (DisplayUtil.getSreenWidth(this) * 250f/ 640f);
mBanner.setLayoutParams(params);

ArrayList<Integer> bannerArray = new ArrayList<Integer>();
bannerArray.add(Integer.valueOf(R.drawable.banner_1));
bannerArray.add(Integer.valueOf(R.drawable.banner_2));
bannerArray.add(Integer.valueOf(R.drawable.banner_3));

```

```
bannerArray.add(Integer.valueOf(R.drawable.banner_4));  
bannerArray.add(Integer.valueOf(R.drawable.banner_5));  
mBanner.setImage(bannerArray);  
    mBanner.setOnBannerListener(this);  
mBanner.start();  
Banner47-2557-26
```



7-254



7.4

RecyclerView

7.4.1 RecyclerView TabLayout RecyclerView

List View GridView Staggered GridView

RecyclerView 5.0 Android build.gradle dependencies recyclerview
Pinterest Like Adapter View RecyclerView App

```
compile 'com.android.support:recyclerview-v7:25.1.0'
```

```
setAdapter RecyclerView.Adapter
```

```
setLayoutManager LinearLayoutManager GridLayoutManager
```

```
StaggeredGridLayoutManager
```

```
addItemDecoration
```

```
removeItemDecoration
```

```
setItemAnimator DefaultItemAnimator
```

```
addOnItemTouchListener RecyclerView
```

```
removeOnItemTouchListener
```

```
RecyclerView — RecyclerView.Adapter RecyclerView.setAdapter
```

```
RecyclerView.Adapter RecyclerView.Adapter
```

1.

```
getItemCount
```

```
onCreateViewHolder
```

```
onBindViewHolder
```

2.

```
getItemViewType onCreateViewHolder
```

```
getItemId
```

3.

```
scrollToPosition
```

notifyItemInserted
notifyItemRemoved
notifyItemChanged
notifyDataSetChanged
RecyclerView.Adapter

```
public class LinearAdapter extends RecyclerView.Adapter<ViewHolder> implements  
    OnItemClickListener, OnItemLongClickListener {  
    private final static String TAG = "LinearAdapter";  
    private Context mContext;  
    private LayoutInflater mInflater;  
    private ArrayList<GoodsInfo> mPublicArray;
```

```

public LinearAdapter(Context context, ArrayList<GoodsInfo> publicArray) {
    mContext = context;
    mInflater = LayoutInflater.from(context);
    mPublicArray = publicArray;
}

@Override
public int getItemCount() {
    return mPublicArray.size();
}

@Override
public ViewHolder onCreateViewHolder(ViewGroup vg, int viewType) {
    View v = null;
    ViewHolder holder = null;
    v = mInflater.inflate(R.layout.item_linear, vg, false);
    holder = new ItemHolder(v);
    return holder;
}

@Override
public void onBindViewHolder(ViewHolder vh, final int position) {
    ItemHolder holder = (ItemHolder) vh;
    holder.iv_pic.setImageResource(mPublicArray.get(position).pic_id);
    holder.tv_title.setText(mPublicArray.get(position).title);
    holder.tv_desc.setText(mPublicArray.get(position).desc);
    // 列表项的点击事件需要自己实现
    holder.ll_item.setOnClickListener(new OnClickListener() {
        @Override
        public void onClick(View v) {
            if (mOnItemClickListener != null) {
                mOnItemClickListener.onItemClick(v, position);
            }
        }
    });
    holder.ll_item.setOnLongClickListener(new OnLongClickListener() {
        @Override
        public boolean onLongClick(View v) {
            if (mOnItemLongClickListener != null) {
                mOnItemLongClickListener.onItemLongClick(v, position);
            }
            return true;
        }
    });
}

```

```

    }
    });
}

@Override
public int getItemViewType(int position) {
    // 这里返回每项的类型，开发者可自定义头部类型与一般类型
    // 在 onCreateViewHolder 方法中根据类型加载的不同布局，从而实现带头部的网格布局
    return 0;
}

@Override
public long getItemId(int position) {
    return position;
}

public class ItemHolder extends RecyclerView.ViewHolder {
    public LinearLayout ll_item;
    public ImageView iv_pic;
    public TextView tv_title;
    public TextView tv_desc;

    public ItemHolder(View v) {
        super(v);
        ll_item = (LinearLayout) v.findViewById(R.id.ll_item);
        iv_pic = (ImageView) v.findViewById(R.id.iv_pic);
        tv_title = (TextView) v.findViewById(R.id.tv_title);
        tv_desc = (TextView) v.findViewById(R.id.tv_desc);
    }
}

private OnItemClickListener mOnItemClickListener;
public void setOnItemClickListener(OnItemClickListener listener) {
    this.mOnItemClickListener = listener;
}

private OnItemLongClickListener mOnItemLongClickListener;
public void setOnItemLongClickListener(OnItemLongClickListener listener) {
    this.mOnItemLongClickListener = listener;
}

@Override
public void onItemClick(View view, int position) {
    String desc = String.format("您点击了第%d项，标题是%s", position + 1,
        mPublicArray.get(position).title);
}

```

```

        Toast.makeText(mContext, desc, Toast.LENGTH_SHORT).show();
    }

    @Override
    public void onItemLongClick(View view, int position) {
        String desc = String.format("您长按了第%d项，标题是%s", position + 1,
            mPublicArray.get(position).title);
        Toast.makeText(mContext, desc, Toast.LENGTH_SHORT).show();
    }
}

```

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_recycler_linear);
    rv_linear = (RecyclerView) findViewById(R.id.rv_linear);
    LinearLayoutManager manager = new LinearLayoutManager(this);
    manager.setOrientation(LinearLayout.VERTICAL);
    rv_linear.setLayoutManager(manager);
    LinearAdapter adapter = new LinearAdapter(this, GoodsInfo.getDefaultList());
    adapter.setOnItemClickListener(adapter);
    adapter.setOnItemLongClickListener(adapter);
    rv_linear.setAdapter(adapter);
    rv_linear.setItemAnimator(new DefaultItemAnimator());
    rv_linear.addItemDecoration(new SpacesItemDecoration(1));
}

```

7-27 ListView RecyclerView



7-27

7.4.2 LayoutManager RecyclerView RecyclerViewLayoutManager3.setLayoutManager

1. LinearLayoutManager

LinearLayoutManager(LinearLayoutManager)

LinearLayoutManager

setOrientation :LinearLayout.HORIZONTAL

VERTICAL

setReverseLayout :false

RecyclerView

```
public class SpacesItemDecoration extends RecyclerView.ItemDecoration {
    private int space;
    public SpacesItemDecoration(int space) {
        this.space = space;
    }

    @Override
    public void getItemOffsets(Rect outRect, View view, RecyclerView parent, RecyclerView.State state) {
        outRect.left = space;
        outRect.right = space;
        outRect.bottom = space;
        outRect.top = space;
    }
}
```

2. GridLayoutManager

GridLayoutManager(GridLayoutManager)

GridView(GridView)

GridLayoutManager

setSpanCount

setSpanSizeLookup(GridLayoutManager.SpanSizeLookup)

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_recycler_grid);
    rv_grid = (RecyclerView) findViewById(R.id.rv_grid);
    GridLayoutManager manager = new GridLayoutManager(this, 5);
    rv_grid.setLayoutManager(manager);
    GridAdapter adapter = new GridAdapter(this, GoodsInfo.getDefaultGrid());
    adapter.setOnItemClickListener(adapter);
    adapter.setOnItemLongClickListener(adapter);
    rv_grid.setAdapter(adapter);
    rv_grid.setItemAnimator(new DefaultItemAnimator());
    rv_grid.addItemDecoration(new SpacesItemDecoration(1));
}
```

7-28 GridView



7-28

GridViewsetSpanSizeLookupActivity


```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_recycler_combine);
    rv_combine = (RecyclerView) findViewById(R.id.rv_combine);
    GridLayoutManager manager = new GridLayoutManager(this, 4);
    //以下占位规则的意思是：第一项和第二项占两列，其他项占一列
    //如果网格的列数为4，那么第一项和第二项平分第一行，从第二行开始每行有4项
    manager.setSpanSizeLookup(new GridLayoutManager.SpanSizeLookup() {
        @Override
        public int getSpanSize(int position) {
            if (position == 0 || position == 1) {
                return 2;
            } else {
                return 1;
            }
        }
    });
    rv_combine.setLayoutManager(manager);
    CombineAdapter adapter = new CombineAdapter(this, GoodsInfo.getDefaultCombine());
    adapter.setOnItemClickListener(adapter);
    adapter.setOnItemLongClickListener(adapter);
    rv_combine.setAdapter(adapter);
    rv_combine.setItemAnimator(new DefaultItemAnimator());
    rv_combine.addItemDecoration(new SpacesItemDecoration(1));
}

```



7-29

3. StaggeredGridLayoutManager App
StaggeredGridLayoutManager
StaggeredGridLayoutManager

setSpanCount :
setOrientation :LinearLayoutManager
setReverseLayoutfalsetrue

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_recycler_staggered);
    rv_staggered = (RecyclerView) findViewById(R.id.rv_staggered);
    StaggeredGridLayoutManager manager = new StaggeredGridLayoutManager(3, LinearLayoutManager.
VERTICAL);
    rv_staggered.setLayoutManager(manager);
    StaggeredAdapter adapter = new StaggeredAdapter(this, GoodsInfo.getDefaultStag());
    adapter.setOnItemClickListener(adapter);
    adapter.setOnItemLongClickListener(adapter);
    rv_staggered.setAdapter(adapter);
    rv_staggered.setItemAnimator(new DefaultItemAnimator());
    rv_staggered.addItemDecoration(new SpacesItemDecoration(3));
}
```

7-307-31App“”

Group



促销价



惊爆价



跳楼价



清仓价



白菜价



割肉价



7-312

7.4.3

Activity

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_recycler_dynamic);
    findViewById(R.id.btn_recycler_add).setOnClickListener(this);
    rv_dynamic = (RecyclerView) findViewById(R.id.rv_dynamic);
    LinearLayoutManager manager = new LinearLayoutManager(this);
    manager.setOrientation(LinearLayout.VERTICAL);
    rv_dynamic.setLayoutManager(manager);
    mAllArray = GoodsInfo.getDefaultList();
    mPublicArray = GoodsInfo.getDefaultList();
    mAdapter = new LinearDynamicAdapter(this, mPublicArray);
    mAdapter.setOnItemClickListener(this);
    mAdapter.setOnItemLongClickListener(this);
    mAdapter.setOnItemDeleteClickListener(this);
    rv_dynamic.setAdapter(mAdapter);
    rv_dynamic.setItemAnimator(new DefaultItemAnimator());
    rv_dynamic.addItemDecoration(new SpacesItemDecoration(1));
}

@Override
public void onClick(View v) {
    if (v.getId() == R.id.btn_recycler_add) {
        int position = (int) (Math.random()*100%mAllArray.size());
        GoodsInfo old_item = mAllArray.get(position);
        GoodsInfo new_item = new GoodsInfo(old_item.pic_id, old_item.title, old_item.desc);
        mPublicArray.add(0, new_item);
        mAdapter.notifyItemInserted(0);
        rv_dynamic.scrollToPosition(0);
    }
}

@Override
public void onItemLongClick(View view, int position) {
    GoodsInfo item = mPublicArray.get(position);
    item.bPressed = !item.bPressed;
    mPublicArray.set(position, item);

```

```

        mAdapter.notifyItemChanged(position);
    }

@Override
public void onItemClick(View view, int position) {
    String desc = String.format("您点击了第%d项，标题是%s", position + 1,
        mPublicArray.get(position).title);
    Toast.makeText(this, desc, Toast.LENGTH_SHORT).show();
}

@Override
public void onItemDeleteClick(View view, int position) {
    mPublicArray.remove(position);
    mAdapter.notifyItemRemoved(position);
}

```

7-327-337-347-357-327-337-34iphone

“”“”7-35



Group

增加新聊天



北方周末

俄罗斯老人在东北，生活滋润乐不思蜀



首都日报

金秋时节香山染红，市民纷纷登山赏叶



海峡时报

台风接踵而来，出门小心暴雨



北方周末

俄罗斯老人在东北，生活滋润乐不思蜀



参照消息

蒙内铁路建成通车，中国标准再下一城



挨踢杂志

米6大战Mate10，千元智能机谁领风骚

Group

增加新聊天



北方周末

俄罗斯老人在东北，生活滋润乐不思蜀



首都日报

金秋时节香山染红，市民纷纷登山赏叶

删除该聊天



海峡时报

台风接踵而来，出门小心暴雨



北方周末

俄罗斯老人在东北，生活滋润乐不思蜀



参照消息

蒙内铁路建成通车，中国标准再下一城



挨踢杂志

米6大战Mate10，千元智能机谁领风骚

Group

增加新聊天



北方周末

俄罗斯老人在东北，生活滋润乐不思蜀



海峡时报

台风接踵而来，出门小心暴雨



北方周末

俄罗斯老人在东北，生活滋润乐不思蜀



参照消息

蒙内铁路建成通车，中国标准再下一城



挨踢杂志

米6大战Mate10，千元智能机谁领风骚



7-36

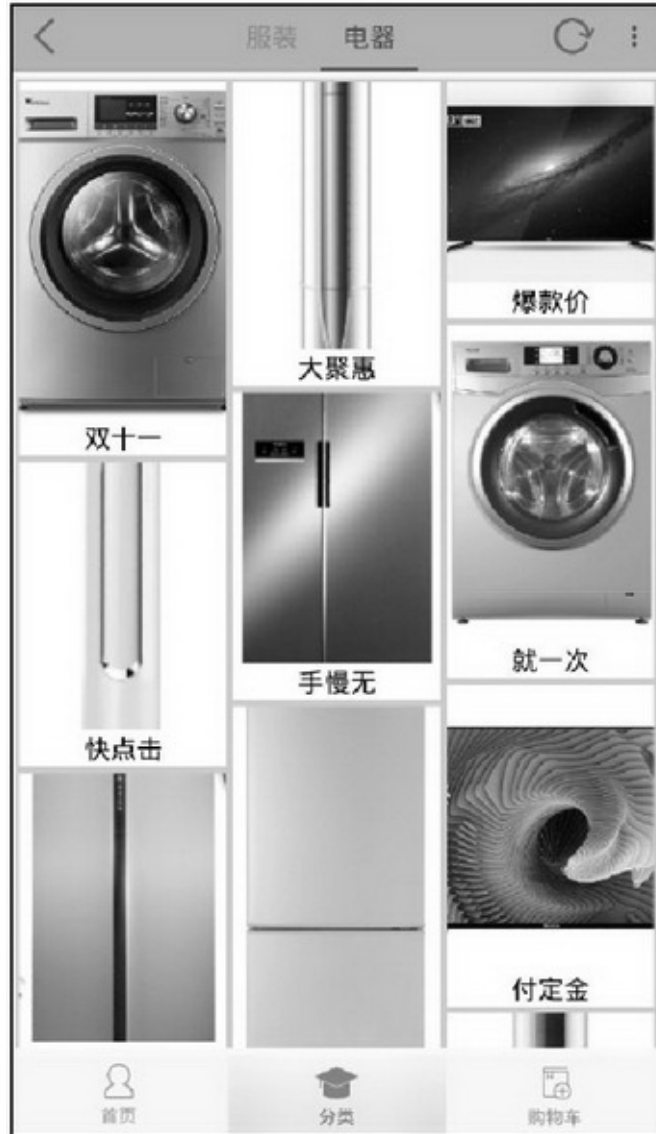
App“”AppAndroidAppUIApp

7.5.1

7-36AppAppAppApp
7-377-38



7-37



7-38

Tabbar

ToolbarToolbar

OverflowMenu

SearchView

BannerBanner

RecyclerViewBanner

TabLayout“”“”

RecyclerView

ViewPagerFragment4Appdemo

7.5.2SwipeRefreshLayout AppAndroid

SwipeRefreshLayout

SwipeRefreshLayout

setOnRefreshListenerOnRefreshListeneronRefresh

setRefreshingtruefalse

isRefreshing

setColorSchemeColors

setProgressBackgroundColorSchemeColor

setProgressViewOffset

setDistanceToTriggerSync

SwipeRefreshLayoutScrollViewListView

GridViewRecyclerViewSwipeRefreshLayout

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="5dp" >

    <android.support.v4.widget.SwipeRefreshLayout
        android:id="@+id/srl_simple"
        android:layout_width="match_parent"
        android:layout_height="match_parent" >

        <ScrollView
            android:layout_width="match_parent"
            android:layout_height="wrap_content" >

            <TextView
                android:id="@+id/tv_simple"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:gravity="center"
                android:paddingTop="10dp"
                android:text="这是一个简单视图"
                android:textColor="#000000"
                android:textSize="17sp" />

            </ScrollView>
        </android.support.v4.widget.SwipeRefreshLayout>
    </LinearLayout>
```

Activity

```

public class SwipeRefreshActivity extends AppCompatActivity implements OnRefreshListener {
    private TextView tv_simple;
    private SwipeRefreshLayout srl_simple;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_swipe_refresh);
        tv_simple = (TextView) findViewById(R.id.tv_simple);
        srl_simple = (SwipeRefreshLayout) findViewById(R.id.srl_simple);
        srl_simple.setOnRefreshListener(this);
        srl_simple.setColorSchemeResources(R.color.red, R.color.orange, R.color.green, R.color.blue);
    }

    @Override
    public void onRefresh() {
        tv_simple.setText("正在刷新");
        mHandler.postDelayed(mRefresh, 2000);
    }

    private Handler mHandler = new Handler();
    private Runnable mRefresh = new Runnable() {
        @Override
        public void run() {
            tv_simple.setText("刷新完成");
            srl_simple.setRefreshing(false);
        }
    };
}

```

7-397-407-397-40





7-40

SwipeRefreshLayoutRecyclerViewSwipeRefreshLayout RecyclerView

```
@Override
public void onRefresh() {
    mHandler.postDelayed(mRefresh, 2000);
}

private Handler mHandler = new Handler();
private Runnable mRefresh = new Runnable() {
    @Override
    public void run() {
        srl_dynamic.setRefreshing(false);
        int position = (int) (Math.random() * 100 % mAllArray.size());
        GoodsInfo old_item = mAllArray.get(position);
        GoodsInfo new_item = new GoodsInfo(old_item.pic_id, old_item.title, old_item.desc);
        mPublicArray.add(0, new_item);
        mAdapter.notifyItemInserted(0);
        // 当循环视图的列表项已经占满整个屏幕时，再往顶部添加一条新记录，感觉屏幕没有发生变化，也没看到插入动画。此时要调用 scrollToPosition(0)方法，表示滚动到第一条记录。
        rv_dynamic.scrollToPosition(0);
    }
};
```

7-417-427-417-42

Group



首都日报

金秋时节香山染红，市民纷纷登山赏叶



海峡时报

台风接踵而来，出门小心暴雨



北方周末

俄罗斯老人在东北，生活滋润乐不思蜀



参照消息

蒙内铁路建成通车，中国标准再下一城



挨踢杂志

米6大战Mate10，千元智能机谁领风骚

Group	
	挨踢杂志 米6大战Mate10，千元智能机谁领风骚
	首都日报 金秋时节香山染红，市民纷纷登山赏叶
	海峡时报 台风接踵而来，出门小心暴雨
	北方周末 俄罗斯老人在东北，生活滋润乐不思蜀
	参照消息 蒙内铁路建成通车，中国标准再下一城
	挨踢杂志 米6大战Mate10，千元智能机谁领风骚

7-42

7.5.3

TabLayoutRecyclerView6

步骤 01 package6

com.example.department.activityAcitivity

com.example.department.adapter

com.example.department.bean

com.example.department.fragment

com.example.department.util

com.example.department.widget

步骤 02 AppDepartmentStoreActivity.java

activity_department_store.xml3DepartmentHomeActivity.java

activity_department_home.xmlDepartmentClassActivity.java

activity_department_class.xmlDepartmentCartActivity.java

activity_department_cart.xml

SearchViewActivitySearchResultActivity

步骤 03

build.gradledependencies3appcompat-v7design
recyclerview-v7

compile 'com.android.support:appcompat-v7:25.1.0'

compile 'com.android.support:design:25.1.0'

compile 'com.android.support:recyclerview-v7:25.1.0'

步骤 04

AndroidManifest.xml

(1)activity

<activity android:name=".DepartmentStoreActivity"

android:theme="@style/AppCompatTheme" /> <activity

android:name=".DepartmentHomeActivity"

android:theme="@style/AppCompatTheme" /> <activity

android:name=".DepartmentClassActivity"

android:theme="@style/AppCompatTheme" /> <activity

android:name=".DepartmentCartActivity"

android:theme="@style/AppCompatTheme" /> <activity

android:name=".SearchViewActivity"

android:theme="@style/AppCompatTheme" /> (2)SearchResultActivity

```
<activity android:name=".SearchResultActivity" android:theme="@style/AppCompatTheme" >
    <intent-filter>
        <action android:name="android.intent.action.SEARCH"/>
    </intent-filter>
    <meta-data android:name="android.app.searchable" android:resource="@xml/searchable"/>
</activity>
```

activityAppCompatThemeActionBarActivityToolBarActionBar

步骤 05 XML5

- (1)res/drawable
- (2)res/layout
- (3)res/menu
- (4)res/values/styles.xmlAppCompatActivity
- (5)res/xmlsearchable.xmlsearchable

步骤 06 java

DepartmentHomeActivity.java

```
public class DepartmentHomeActivity extends AppCompatActivity implements BannerClickListener {
    private final static String TAG = "DepartmentHomeActivity";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_department_home);
        Toolbar tl_head = (Toolbar) findViewById(R.id.tl_head);
        tl_head.setTitle("商城首页");
        setSupportActionBar(tl_head);
        initBanner();
        initGrid();
        initCombine();
    }
}
```

```

private void initBanner() {
    BannerPager banner = (BannerPager) findViewById(R.id.banner_pager);
    LayoutParams params = (LayoutParams) banner.getLayoutParams();
    params.height = (int) (DisplayUtil.getScreenWidth(this) * 250f / 640f);
    banner.setLayoutParams(params);
    ArrayList<Integer> bannerArray = new ArrayList<Integer>();
    bannerArray.add(Integer.valueOf(R.drawable.banner_1));
    bannerArray.add(Integer.valueOf(R.drawable.banner_2));
    bannerArray.add(Integer.valueOf(R.drawable.banner_3));
    bannerArray.add(Integer.valueOf(R.drawable.banner_4));
    bannerArray.add(Integer.valueOf(R.drawable.banner_5));
    banner.setImage(bannerArray);
    banner.setOnBannerListener(this);
    banner.start();
}

@Override
public void onBannerClick(int position) {
    String desc = String.format("您点击了第%d张图片", position+1);
    Toast.makeText(this, desc, Toast.LENGTH_LONG).show();
}

private void initGrid() {
    RecyclerView rv_grid = (RecyclerView) findViewById(R.id.rv_grid);
    GridLayoutManager manager = new GridLayoutManager(this, 5);
    rv_grid.setLayoutManager(manager);
    GridAdapter adapter = new GridAdapter(this, GoodsInfo.getDefaultGrid());
    adapter.setOnItemClickListener(adapter);
    adapter.setOnItemLongClickListener(adapter);
    rv_grid.setAdapter(adapter);
    rv_grid.setItemAnimator(new DefaultItemAnimator());
    rv_grid.addItemDecoration(new SpacesItemDecoration(1));
}

private void initCombine() {
    RecyclerView rv_combine = (RecyclerView) findViewById(R.id.rv_combine);
    GridLayoutManager manager = new GridLayoutManager(this, 4);
    manager.setSpanSizeLookup(new GridLayoutManager.SpanSizeLookup() {
        @Override
        public int getSpanSize(int position) {
            if (position == 0 || position == 1) {
                return 2;
            }
        }
    });
}

```

```

        } else {
            return 1;
        }
    }
});
rv_combine.setLayoutManager(manager);
CombineAdapter adapter = new CombineAdapter(this, GoodsInfo.getDefaultCombine());
adapter.setOnItemClickListener(adapter);
adapter.setOnItemLongClickListener(adapter);
rv_combine.setAdapter(adapter);
rv_combine.setItemAnimator(new DefaultItemAnimator());
rv_combine.addItemDecoration(new SpacesItemDecoration(1));
}

@Override
public boolean onOptionsItemSelected(int featureId, Menu menu) {
    Utils.setOverflowIconVisible(featureId, menu); // 显示菜单项左侧的图标
    return super.onOptionsItemSelected(featureId, menu);
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.menu_home, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    int id = item.getItemId();
    if (id == android.R.id.home) {
        finish();
    } else if (id == R.id.menu_search) {
        Intent intent = new Intent(this, SearchViewActivity.class);
        intent.putExtra("collapse", false);
        startActivity(intent);
    } else if (id == R.id.menu_refresh) {
        Toast.makeText(this, "当前刷新时间: "+Utils.getNowDateTime("yyyy-MM-dd
HH:mm:ss"), Toast.LENGTH_LONG).show();
        return true;
    } else if (id == R.id.menu_about) {
        Toast.makeText(this, "这个是商城首页", Toast.LENGTH_LONG).show();
        return true;
    } else if (id == R.id.menu_quit) {

```

```

        finish();
    }
    return super.onOptionsItemSelected(item);
}
}
}

```

7.6

App(3)()(Banner)(3)

“”App

4

(1)

(2)

(3)Banner

(4)3

第8章 调试与上线



AppAppAppAppApp

8.1

AppAPKAPK

8.1.1

AppAndroid Studio

(1)

(2)

(3)

(4)IMEI

(5)

(6)

AppAppGenymotion——

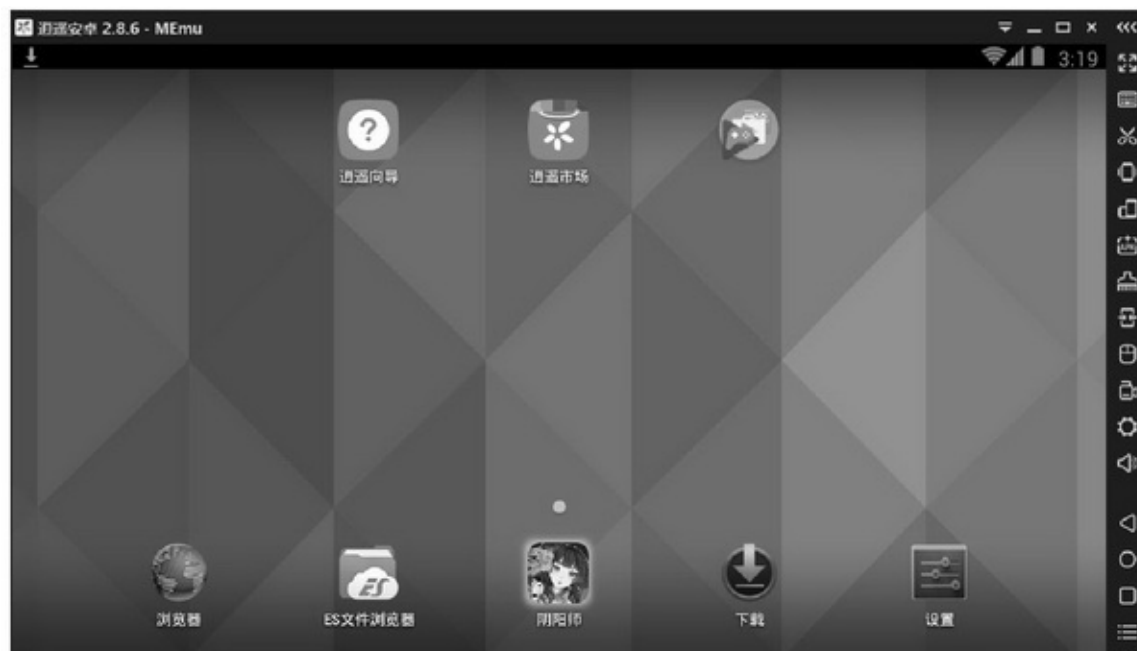
1.

Android4.2.2(SDK19)<http://www.xyaz.cn/8-1>



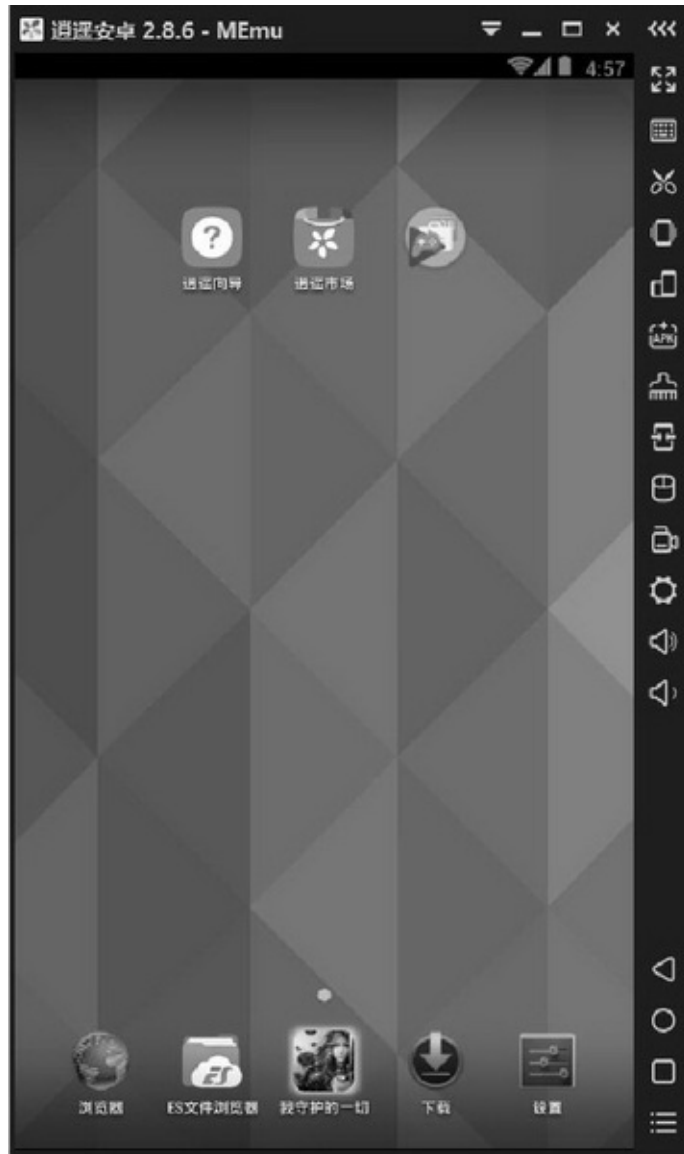
8-1

“”””08-2



8-2

56“”8-3
“”()“”CPU“”
IMEI8-4“”



8-3



8-4

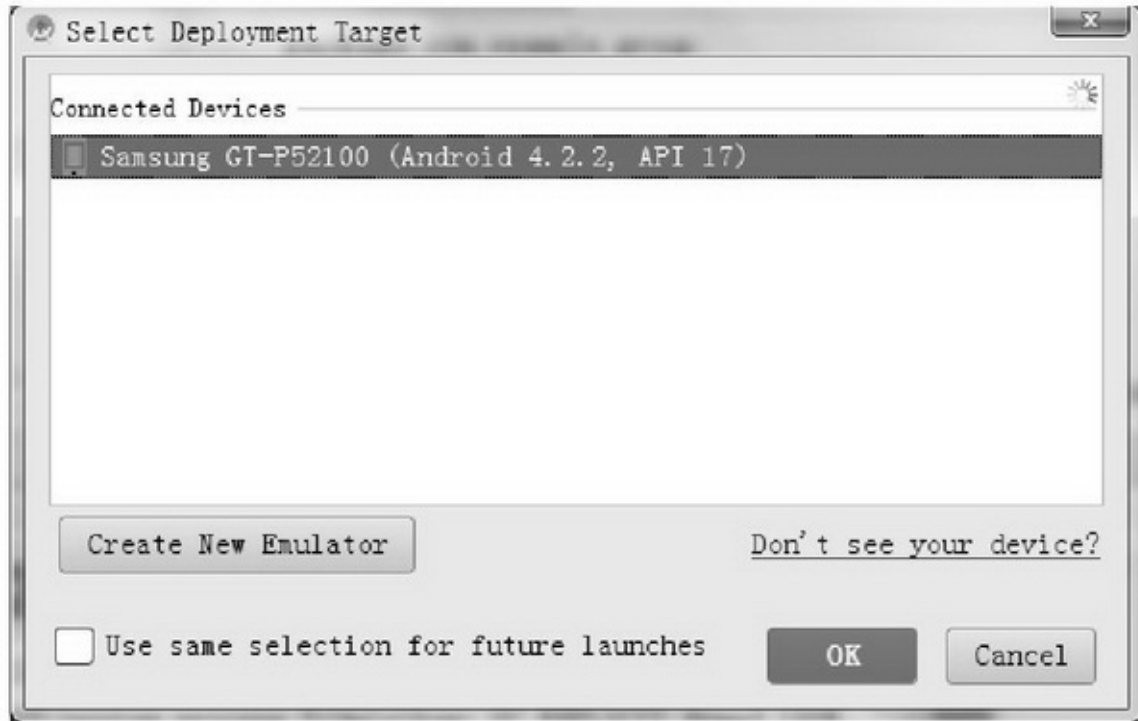
(4)(App)44

AppAndroid StudioAndroid StudioRun → Run '***'8-5

Samsung GT-P52100(Android 4.2.2, API 17)OKAndroid Studio
AppApp

2.

Android4.4.2(SDK17)<http://www.yeshen.com/8-6>



8-5App



8-6

“””8-7



8-7

“””8-8



8-8

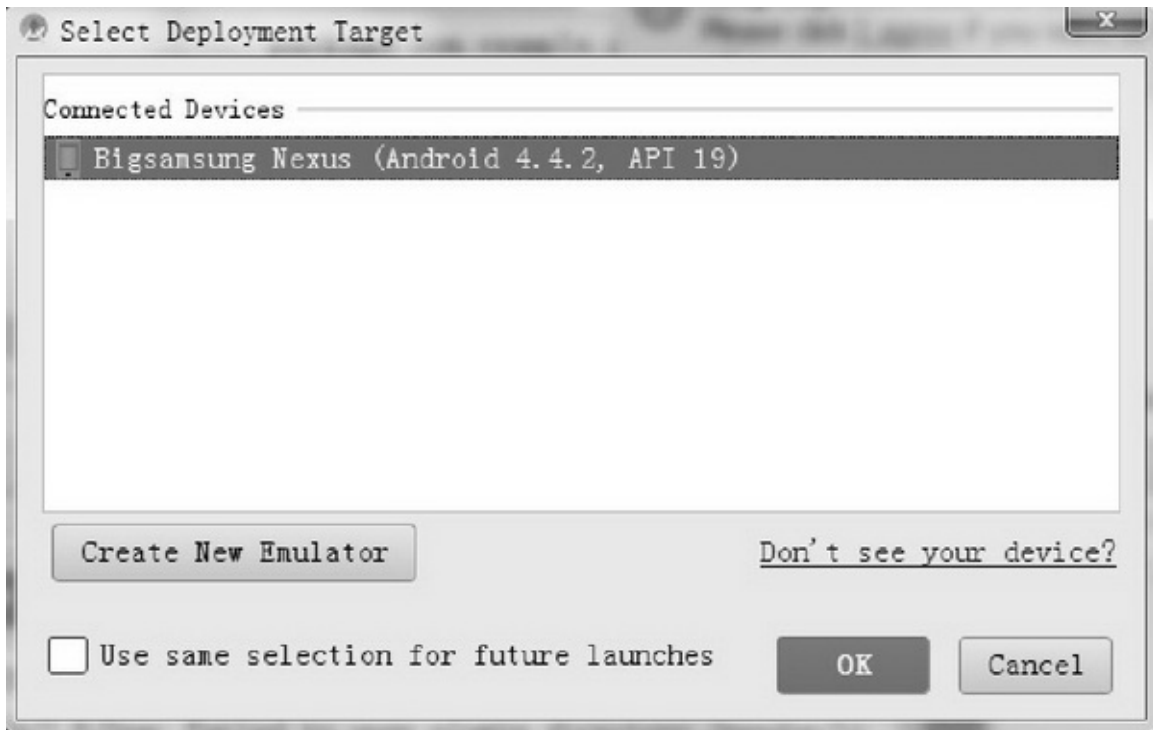
8-9



8-9

“CPU()”
3)AltCtrl“
”0

AppAndroid StudioAndroid StudioRun → Run '***'8-10



8-10App

Bigsamsung Nexus(Android 4.4.2, API 19)OKAndroid StudioApp

8.1.2

4

1.

USB

2.

USB91

3.USB

“USB”USB“USB”USB8-11

USB“USB”8-12“”“”App



允许 USB 调试吗？

这台计算机的 RSA 密钥指纹如下：

96:6B:73:33:C3:A9:F1:6D:D1:4A:0D:

69:9A:8F:B4:DE

一律允许使用这台计算机进行调试

取消

确定

8-12USB



8-13

USB“” → “”(“” → “” → “”)USB8-13

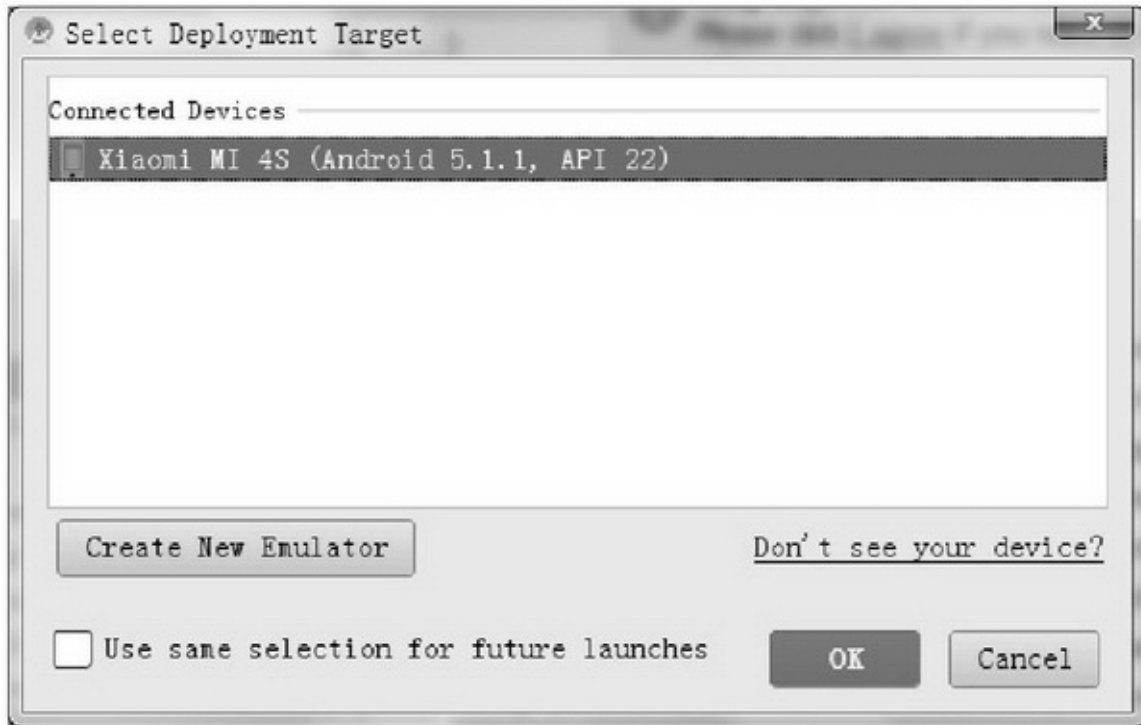
“”“”USB“” → “” → “”“”“

”“”

4.

Android StudioAppApp

AppAndroid StudioRun → Run '***'8-14OKAndroid StudioApp

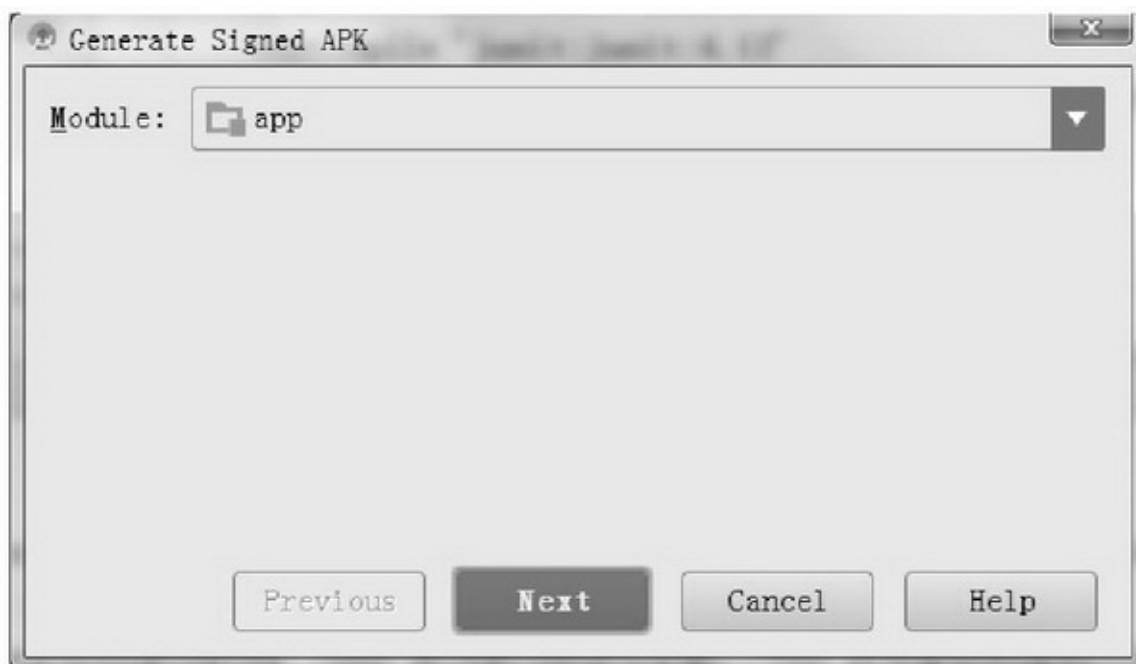


8-14App

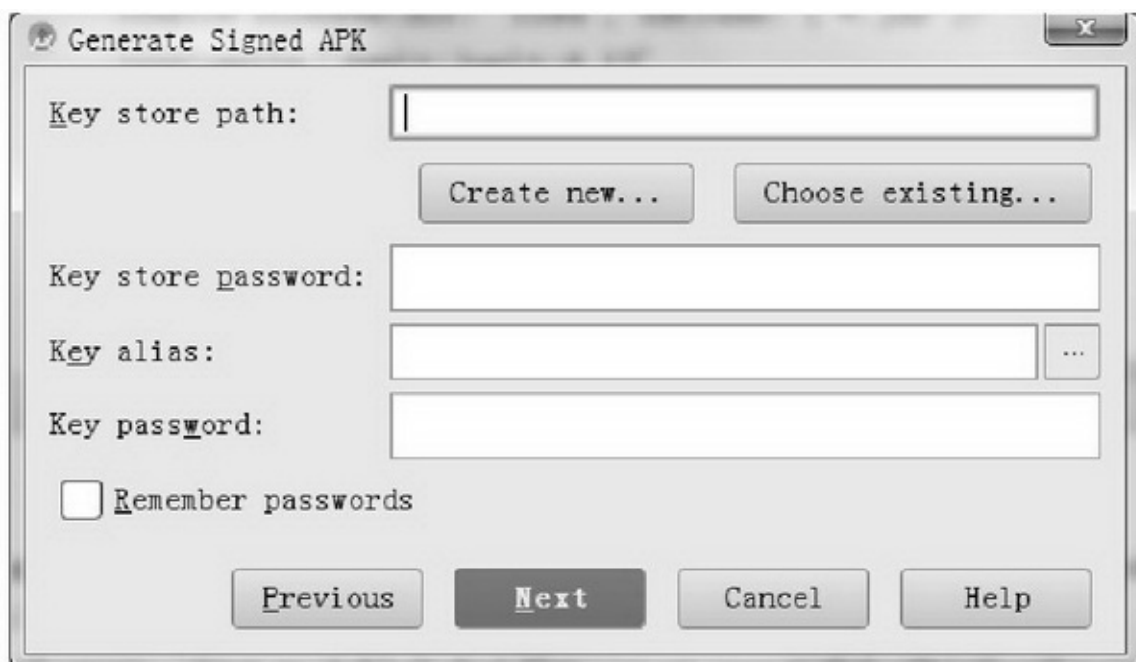
8.1.3APK

AppAppAppAppAPK(App)APKAndroid
StudioAPK

- 步骤 01 Build → Generate Signed APK...8-15
- 步骤 02 (test)NextAPK8-16



8-15



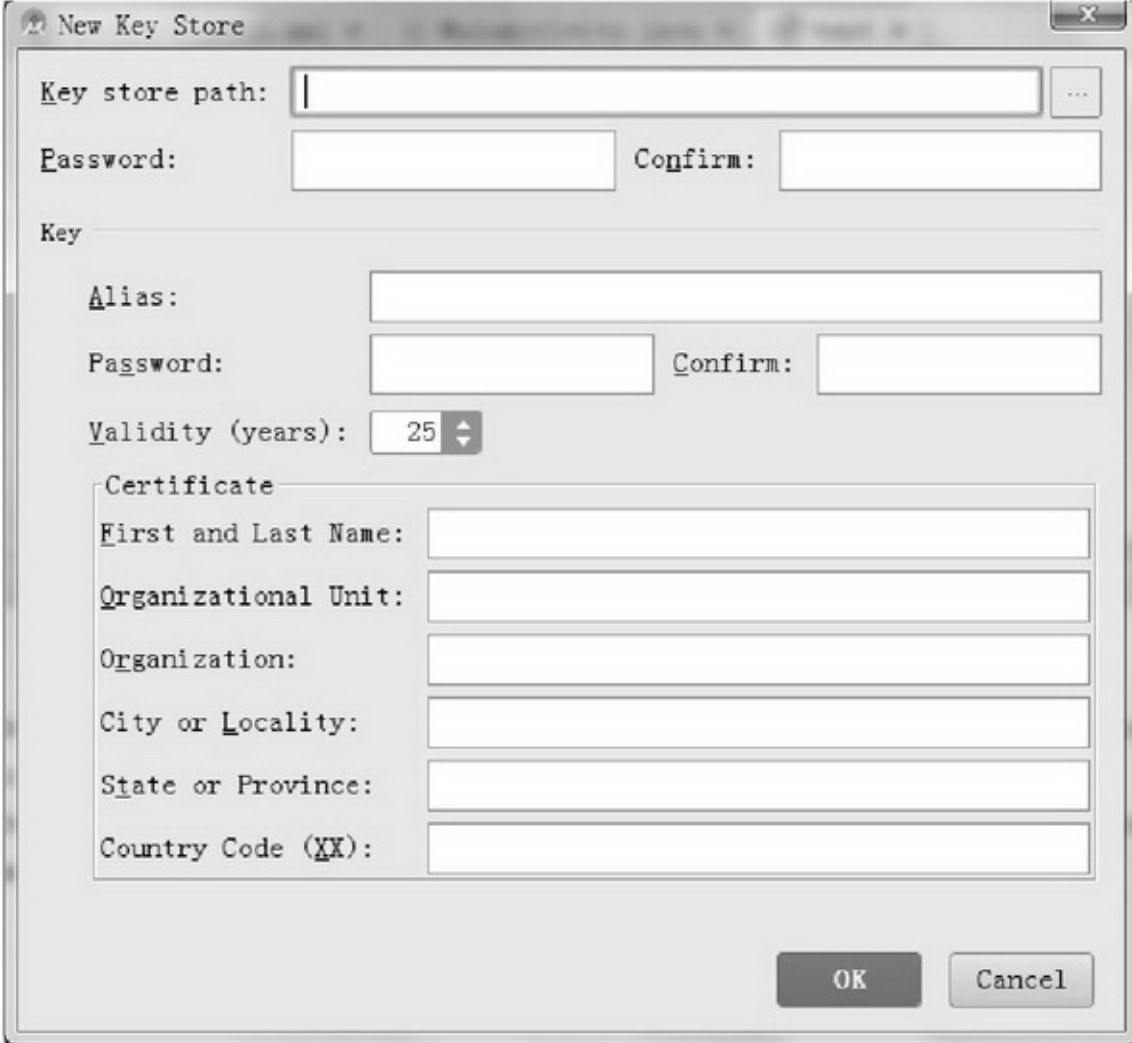
8-16APK

步骤 03

Choose existing...Create new...

8-17

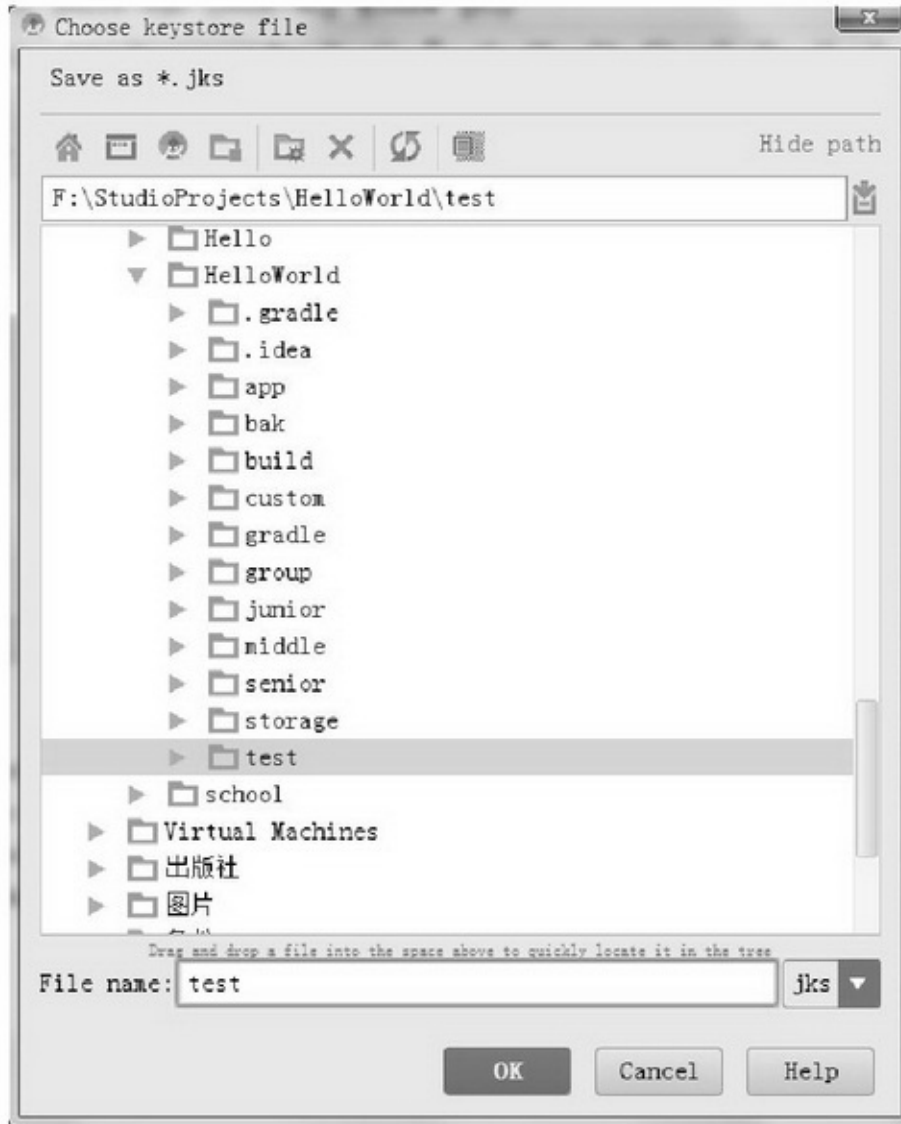
步骤 04  8-18



The image shows a 'New Key Store' dialog box with the following fields and controls:

- Key store path:** A text input field with a browse button (...).
- Password:** A text input field.
- Confirm:** A text input field.
- Key** section:
 - Alias:** A text input field.
 - Password:** A text input field.
 - Confirm:** A text input field.
 - Validity (years):** A spin box set to 25.
- Certificate** section:
 - First and Last Name:** A text input field.
 - Organizational Unit:** A text input field.
 - Organization:** A text input field.
 - City or Locality:** A text input field.
 - State or Province:** A text input field.
 - Country Code (XX):** A text input field.
- Buttons:** 'OK' and 'Cancel' buttons at the bottom right.

8-17



8-18

步骤 05

File name OK Password Confirm Alias

Password Confirm Validity (First and Last Name) 8-19

步骤 06

OK APK Android Studio 8-20

New Key Store

Key store path: F:\StudioProjects\HelloWorld\test\test.jks

Password: Confirm:

Key

Alias: test

Password: Confirm:

Validity (years): 30

Certificate

First and Last Name: ouyangshen

Organizational Unit:

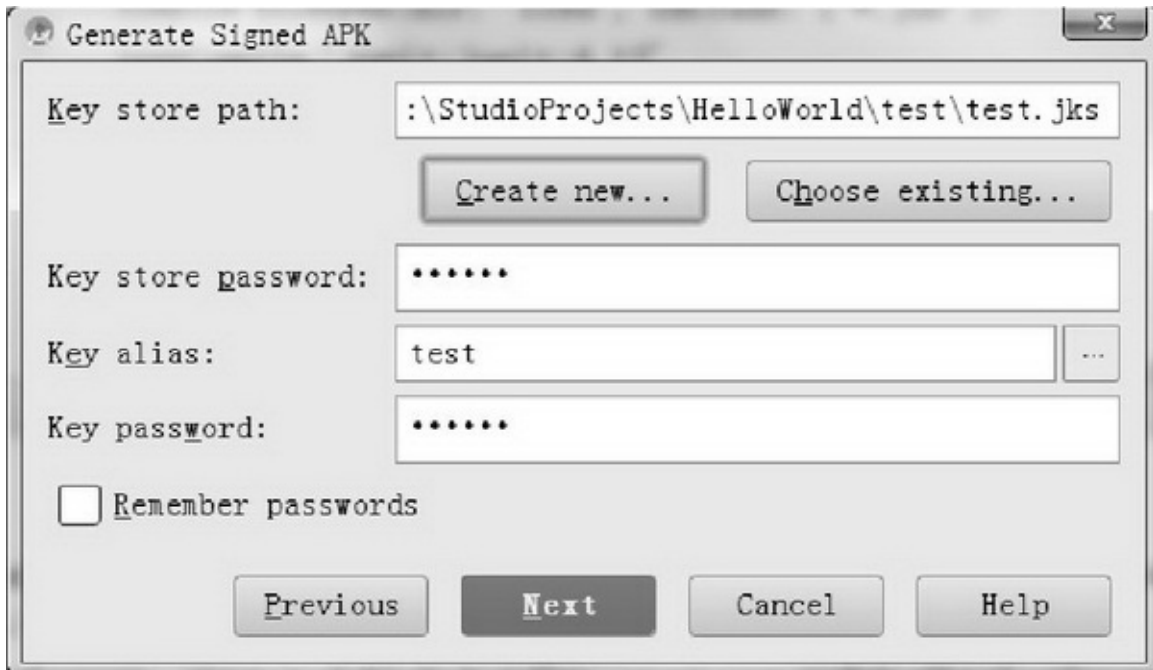
Organization:

City or Locality:

State or Province:

Country Code (XX):

OK Cancel

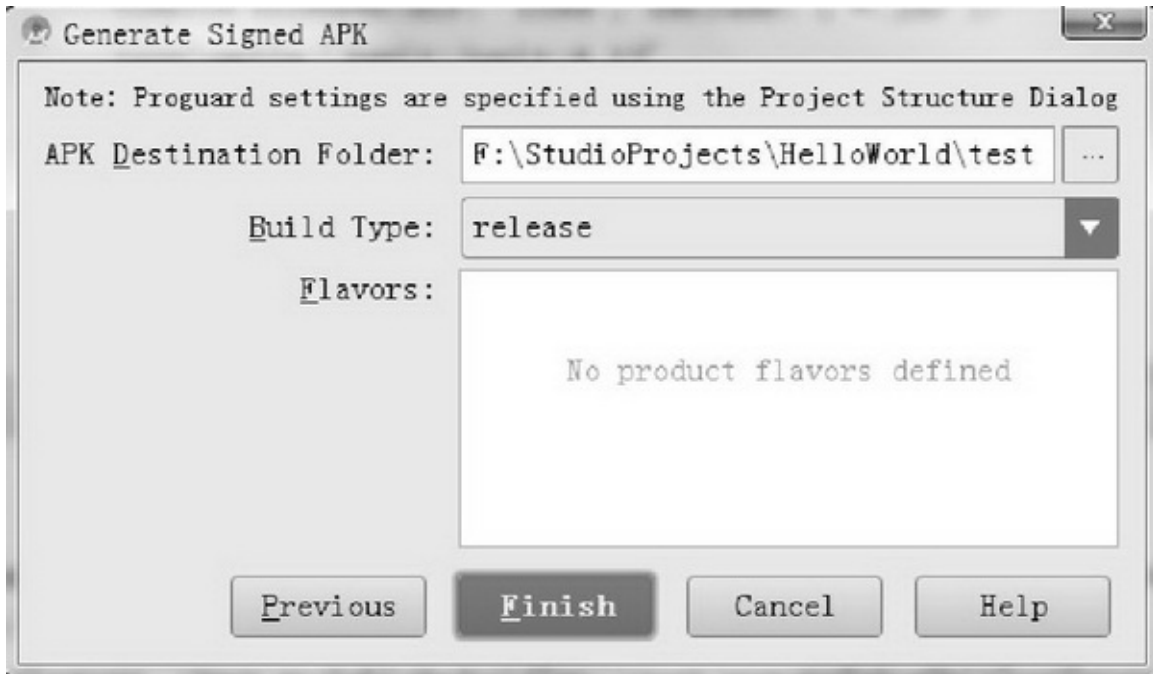


8-20

步骤 07 Next8-21OKAPK8-22Apk



8-21



8-22APK

APKAPK(Build Type)debugreleaseFinishAndroid Studio
APK
APKtest-debug.apkSD

8.2

AppAppAppAppAndroidManifest.xml
MD5RSAAES3DES

8.2.1

AppAppAppAppApp3AppAppApp
1. App
Appres/mipmap-***ic_launcher.pngmipmap-***ic_launcher.png
2. App
Appres/values/strings.xmlapp_namestrings.xmlapp_name
3. App
Appbuild.gradleversionCodeversionNameversionCode1
versionName“..”111
AppversionCodeversionNameversionCodeversionName

(1)App
(2)versionNameversionCodeAndroidversionCode
versionCode
App

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_version);
    ImageView iv_icon = (ImageView) findViewById(R.id.iv_icon);
    TextView tv_desc = (TextView) findViewById(R.id.tv_desc);
    iv_icon.setImageResource(R.mipmap.ic_launcher);
    try {
        PackageInfo pi = getPackageManager().getPackageInfo(getPackageName(), 0);
        String desc = String.format("App 名称为 : %s\nApp 版本号为 : %d\nApp 版本名称为 : %s",
            getResources().getString(R.string.app_name), pi.versionCode, pi.versionName);
        tv_desc.setText(desc);
    } catch (NameNotFoundException e) {
        e.printStackTrace();
    }
}
```



8-23App

App8-23App

8.2.2

bugApp

3

(1)

(2)

(3)

App

3LogToastAlertDialog

1. Log

LogAppconsoleAppApp

```
public class LogTool {
    public static boolean isShow = false; // false 表示上线模式, true 表示开发模式

    public static void v(String tag, String msg) {
        if (isShow == true) {
            Log.v(tag, msg);
        }
    }

    public static void d(String tag, String msg) {
        if (isShow == true) {
            Log.d(tag, msg);
        }
    }

    public static void i(String tag, String msg) {
        if (isShow == true) {
            Log.i(tag, msg);
        }
    }

    public static void w(String tag, String msg) {
        if (isShow == true) {
            Log.w(tag, msg);
        }
    }

    public static void e(String tag, String msg) {
        if (isShow == true) {
            Log.e(tag, msg);
        }
    }

    public static void wtf(String tag, String msg) {
        if (isShow == true) {
            Log.wtf(tag, msg);
        }
    }
}
```

2. Toast

ToastToast(“”)


```
public class ToastTool {
    public static boolean isShow = false; // false 表示上线模式， true 表示开发模式

    public static void showShort(Context ctx, String msg) {
        if (isShow == true) {
            Toast.makeText(ctx, msg, Toast.LENGTH_SHORT).show();
        }
    }

    public static void showLong(Context ctx, String msg) {
        if (isShow == true) {
            Toast.makeText(ctx, msg, Toast.LENGTH_LONG).show();
        }
    }

    public static void showQuit(Context ctx) {
        Toast.makeText(ctx, "再按一次返回键退出!", Toast.LENGTH_SHORT).show();
    }
}
```

3. AlertDialog

“”

```

public class DialogTool {
    public static boolean isShow = false; // false 表示上线模式， true 表示开发模式
    public static int SYSTEM = 0;
    public static int IO = 1;
    public static int NETWORK = 2;
    private static String[] mError = { "系统异常，请稍候再试", "读写失败，请清理内存空间后再试",
        "网络连接失败，请检查网络设置是否开启" };

    public static void showError(Context ctx, int type, String title, Exception e) {
        AlertDialog.Builder builder = new AlertDialog.Builder(ctx);
        builder.setTitle(title);
        if (isShow == true) {
            String desc = String.format("%s\n 异常描述 : %s", mError[type], e.getMessage());
            builder.setMessage(desc);
        } else {
            builder.setMessage(mError[type]);
        }
        builder.setPositiveButton("确定", null);
        builder.create().show();
    }

    public static void showError(Context ctx, int type, String title, int code, String msg) {
        AlertDialog.Builder builder = new AlertDialog.Builder(ctx);
        builder.setTitle(title);
        if (isShow == true) {
            String desc = String.format("%s\n 异常代码: %d\n 异常描述: %s", mError[type], code, msg);
            builder.setMessage(desc);
        } else {
            builder.setMessage(mError[type]);
        }
        builder.setPositiveButton("确定", null);
        builder.create().show();
    }
}

```

AndroidManifest.xml3

(1) application android:debuggable="true" false Android Studio App
debuggable false

(2) App activity service activity service android:exported=
"false"

(3) App AppSD manifest android:installLocation 8-1

安装位置的类型	说明
internalOnly	默认值，只能安装在内部存储。无法通过安全软件的应用搬家功能将其挪到 SD 卡
auto	优先装在内部存储，但若内部存储空间不足，则会安装在 SD 卡。安装之后，用户可通过安全软件选择是否将其挪到 SD 卡。推荐设为该值
preferExternal	安装在 SD 卡上。但若 SD 卡不存在或 SD 卡空间不足，则安装在内部存储

8.2.3

Android

SharedPreferencesXML/data/data//shared_prefs/***.xml

shared.xml

```
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
  <string name="name">Mr Lee</string>
  <int name="age" value="30" />
  <boolean name="married" value="true" />
  <float name="weight" value="100.0" />
</map>
```

App

SQLite/data/data//databases/***.dbdbsqliteManagerSQLite

8-24sqliteManagerApp

Table: NavItem					
id	value	label	logo	updateTime	parentId
1	2	教育阅读	app/appLogo/2.png	2016-07-06 16:56:35	-1
2	3	生活助手	app/appLogo/3.png	2016-07-06 16:57:07	-1
3	4	亲子乐园	app/appLogo/4.png	2016-07-06 16:57:24	-1
4	1	游戏	app/appLogo/1.png	2016-07-06 16:57:54	-1
5	201	认知教学	app/appLogo/201.png	2016-08-02 17:03:13	2
6	202	英语学习	app/appLogo/202.png	2016-08-02 17:03:29	2
7	203	课堂辅导	app/appLogo/203.png	2016-08-02 17:03:45	2
8	204	儿童绘本	app/appLogo/204.png	2016-08-02 17:03:55	2

8-24SQLite

SharedPreferencesSQLiteMD5RSAAES3DES

1. MD5

MD5MD5

```

public class MD5Util {
    public static String encryp(String raw) {
        String md5Str = raw;
        try {
            MessageDigest md = MessageDigest.getInstance("MD5");
            md.update(raw.getBytes());
            byte[] encryContext = md.digest();
            int i;
            StringBuffer buf = new StringBuffer("");
            for (int offset = 0; offset < encryContext.length; offset++) {
                i = encryContext[offset];
                if (i < 0) {
                    i += 256;
                }
                if (i < 16) {
                    buf.append("0");
                }
                buf.append(Integer.toHexString(i));
            }
            md5Str = buf.toString();
        } catch (NoSuchAlgorithmException e) {
            e.printStackTrace();
        }
        return md5Str;
    }
}

```

Test

要加密的字符串： 123456

MD5加密

RSA加密

AES加密

3DES加密

MD5的加密结果

是:e10adc3949ba59abbe56e057f20f883e

MD58-25MD532

2. RSA

RSA

RSA3

(1)bcprov-jdk16-1.46.jarjarlibs

(2)RSABASE64

(3)reverse

RSA

```
public class RSAUtil {  
    private static final String Algorithm = "RSA";  
  
    private static byte[] encrypt(PublicKey pk, byte[] data) throws Exception { //加密  
        try {  
            Cipher cipher = Cipher.getInstance(Algorithm,  
                new org.bouncycastle.jce.provider.BouncyCastleProvider());
```

```

cipher.init(Cipher.ENCRYPT_MODE, pk);
int blockSize = cipher.getBlockSize();
int outputSize = cipher.getOutputSize(data.length);
int leavedSize = data.length % blockSize;
int blocksSize = leavedSize != 0 ? data.length / blockSize + 1 : data.length / blockSize;
byte[] raw = new byte[outputSize * blocksSize];
int i = 0;
while (data.length - i * blockSize > 0) {
    if (data.length - i * blockSize > blockSize) {
        cipher.doFinal(data, i * blockSize, blockSize, raw, i * outputSize);
    } else {
        cipher.doFinal(data, i * blockSize, data.length - i * blockSize, raw, i * outputSize);
    }
    i++;
}
return raw;
} catch (Exception e) {
    throw new Exception(e.getMessage());
}
}

private static byte[] decrypt(PrivateKey pk, byte[] raw) throws Exception { //解密
    try {
        Cipher cipher = Cipher.getInstance(Algorithm,
            new org.bouncycastle.jce.provider.BouncyCastleProvider());
        cipher.init(cipher.DECRYPT_MODE, pk);
        int blockSize = cipher.getBlockSize();
        ByteArrayOutputStream bout = new ByteArrayOutputStream(64);
        int j = 0;
        while (raw.length - j * blockSize > 0) {
            bout.write(cipher.doFinal(raw, j * blockSize, blockSize));
            j++;
        }
        return bout.toByteArray();
    } catch (Exception e) {
        throw new Exception(e.getMessage());
    }
}

//使用 N、e 值还原公钥
private static PublicKey getPublicKey(String modulus, String publicExponent, int radix)
    throws NoSuchAlgorithmException, InvalidKeySpecException {
    BigInteger bigIntModulus = new BigInteger(modulus, radix);

```

```

        BigInteger bigIntPrivateExponent = new BigInteger(publicExponent, radix);
        RSAPublicKeySpec keySpec = new RSAPublicKeySpec(bigIntModulus, bigIntPrivateExponent);
        KeyFactory keyFactory = KeyFactory.getInstance("RSA");
        PublicKey publicKey = keyFactory.generatePublic(keySpec);
        return publicKey;
    }

    // 使用 N、d 值还原私钥
    private static PrivateKey getPrivateKey(String modulus, String privateExponent, int radix)
        throws NoSuchAlgorithmException, InvalidKeySpecException {
        BigInteger bigIntModulus = new BigInteger(modulus, radix);
        BigInteger bigIntPrivateExponent = new BigInteger(privateExponent, radix);
        RSAPrivateKeySpec keySpec = new RSAPrivateKeySpec(bigIntModulus, bigIntPrivateExponent);
        KeyFactory keyFactory = KeyFactory.getInstance("RSA");
        PrivateKey privateKey = keyFactory.generatePrivate(keySpec);
        return privateKey;
    }

    public static String encodeRSA(RSAKeyData key_data, String src) { //加密函数
        if (key_data == null) { //以下为测试用的密钥对
            key_data = new RSAKeyData();
            key_data.public_key = "10001";
            key_data.private_key = "";
            key_data.modulus = "c7f668eccc579bb75527424c21bc31c104bb44e921b4788ebc82cddab
5042909eaea2dd706431531392d79890f9091e13714285a7e79e9d1836397f847046ef2519c9b65022b48bf157fe409
f8a42155734e65467d04ac844dfa0c2ae512517102986ba9b62d67d4c920eae40b2f11c363b218a703467d342faa8171
9f57e2c3";
            key_data.radix = 16;
        }
        try {
            PublicKey key = getPublicKey(key_data.modulus, key_data.public_key, key_data.radix);
            String rev = encodeURL(new StringBuilder(src).reverse().toString());
            byte[] en_byte = encrypt(key, rev.getBytes());
            String base64 = encodeURL(ConvertBytesToBase64.BytesToBase64String(en_byte));
            return base64;
        } catch (Exception e) {
            e.printStackTrace();
            return "RSA 加密失败";
        }
    }

    private static String encodeURL(String str) { //URL 编码
        String encode_str = str;

```

```

        try {
            encode_str = URLEncoder.encode(str, "utf-8");
        } catch (Exception e) {
            e.printStackTrace();
        }
        return encode_str;
    }

    private static String decodeURL(String str) { //URL 解码
        String decode_str = str;
        try {
            decode_str = URLDecoder.decode(str, "utf-8");
        } catch (Exception e) {
            e.printStackTrace();
        }
        return decode_str;
    }

    public static class RSAKeyData {
        public String modulus;
        public String public_key;
        public String private_key;
        public int radix;

        public RSAKeyData() {
            modulus = "";
            public_key = "";
            private_key = "";
            radix = 0;
        }
    }
}

```

RSA8-26URL

Test

要加密的字符串：123456

MD5加密

RSA加密

AES加密

3DES加密

加密结果

是:aLZOoc%2F%2BFL3fbHQFiYgCGdhRcY2Oz0bb%2F
L3tYufeRmnFJ1ahn9ANwc4GMBsqpfyYWJ1W9y0nbd
XMwwGItoFjuavwOD7QndP%2BbdwRqhJ526stQc35a
2BwoyZJhENjWHwSdtETdfab8oYX7YLTaeljCQX9eX2e
TERmzdkOnXwtVJM%3D

8-26RSA

3. AES

AESDES

```

public class AesUtil {
    private static final String Algorithm = "AES";
    private final static String HEX = "0123456789ABCDEF";

    //加密函数。key 为密钥
    public static String encrypt(String key, String src) throws Exception {
        byte[] rawKey = getRawKey(key.getBytes());
        byte[] result = encrypt(rawKey, src.getBytes());
        return toHex(result);
    }

    //解密函数。key 值必须和加密时的 key 一致
    public static String decrypt(String key, String encrypted) throws Exception {
        byte[] rawKey = getRawKey(key.getBytes());
        byte[] enc = toByte(encrypted);
        byte[] result = decrypt(rawKey, enc);
        return new String(result);
    }

    private static void appendHex(StringBuffer sb, byte b) {
        sb.append(HEX.charAt((b >> 4) & 0x0f)).append(HEX.charAt(b & 0x0f));
    }

    private static byte[] getRawKey(byte[] seed) throws Exception {
        KeyGenerator kgen = KeyGenerator.getInstance(Algorithm);
        // SHA1PRNG 强随机种子算法，要区别 Android 4.2.2 以上版本的调用方法
        SecureRandom sr = null;
        if (android.os.Build.VERSION.SDK_INT >= 17) {
            sr = SecureRandom.getInstance("SHA1PRNG", "Crypto");
        } else {
            sr = SecureRandom.getInstance("SHA1PRNG");
        }
        sr.setSeed(seed);
        kgen.init(256, sr); // 256 位/128 位/192 位
        SecretKey skey = kgen.generateKey();
        byte[] raw = skey.getEncoded();
        return raw;
    }

    private static byte[] encrypt(byte[] key, byte[] src) throws Exception {
        SecretKeySpec skeySpec = new SecretKeySpec(key, Algorithm);
        Cipher cipher = Cipher.getInstance(Algorithm);
    }

```

```

        cipher.init(Cipher.ENCRYPT_MODE, skeySpec);
        byte[] encrypted = cipher.doFinal(src);
        return encrypted;
    }

    private static byte[] decrypt(byte[] key, byte[] encrypted) throws Exception {
        SecretKeySpec skeySpec = new SecretKeySpec(key, Algorithm);
        Cipher cipher = Cipher.getInstance(Algorithm);
        cipher.init(Cipher.DECRYPT_MODE, skeySpec);
        byte[] decrypted = cipher.doFinal(encrypted);
        return decrypted;
    }

    private static byte[] toByte(String hexString) {
        int len = hexString.length() / 2;
        byte[] result = new byte[len];
        for (int i = 0; i < len; i++) {
            result[i] = Integer.valueOf(hexString.substring(2 * i, 2 * i + 2), 16).byteValue();
        }
        return result;
    }

    private static String toHex(byte[] buf) {
        if (buf == null) {
            return "";
        }
        StringBuffer result = new StringBuffer(2 * buf.length);
        for (int i = 0; i < buf.length; i++) {
            appendHex(result, buf[i]);
        }
        return result.toString();
    }
}

```

AES8-27

Test

要加密的字符串：123456

MD5加密 RSA加密 AES加密 3DES加密

加密结果是:C120EF2142131FDC8B31D96316518464
解密结果是:123456

8-27AES

4.3DES

3DES(Triple DES)3DESDES3DES3DES24

java.security.InvalidKeyException3DESBASE64

```

public class Des3Util {
    private static final String Algorithm = "DESede"; // 定义加密算法 DESede, 即 3DES

    public static String encrypt(String key, String raw) { //加密函数。key 为密钥
        byte[] enBytes = encryptMode(key, raw.getBytes());
        BASE64Encoder encoder = new BASE64Encoder();
        return encoder.encode(enBytes);
    }

    public static String decrypt(String key, String enc) { //解密函数。key 值必须和加密时的 key 一致
        try {
            BASE64Decoder decoder = new BASE64Decoder();
            byte[] enBytes = decoder.decodeBuffer(enc);
            byte[] deBytes = decryptMode(key, enBytes);
            return new String(deBytes);
        } catch (IOException e) {
            e.printStackTrace();
        }
        return enc;
    }

    private static byte[] encryptMode(String key, byte[] src) {
        try {
            SecretKey deskey = new SecretKeySpec(build3DesKey(key), Algorithm);
            Cipher cipher = Cipher.getInstance(Algorithm);
            cipher.init(Cipher.ENCRYPT_MODE, deskey);
            return cipher.doFinal(src);
        } catch (Exception e) {
            e.printStackTrace();
            return null;
        }
    }

    private static byte[] decryptMode(String key, byte[] src) {
        try {
            SecretKey deskey = new SecretKeySpec(build3DesKey(key), Algorithm);

```

```

        Cipher cipher = Cipher.getInstance(Algorithm);
        cipher.init(Cipher.DECRYPT_MODE, deskey);
        return cipher.doFinal(src);
    } catch (Exception e) {
        e.printStackTrace();
        return null;
    }
}

//根据字符串生成密钥 24 位的字节数组
private static byte[] build3DesKey(String keyStr) throws UnsupportedEncodingException {
    byte[] key = new byte[24];
    byte[] temp = keyStr.getBytes("UTF-8");
    if (key.length > temp.length) {
        System.arraycopy(temp, 0, key, 0, temp.length);
    } else {
        System.arraycopy(temp, 0, key, 0, key.length);
    }
    return key;
}
}

```

3DES8-28AES

Test

要加密的字符串：

MD5加密

RSA加密

AES加密

3DES加密

加密结果是:CUyOh1msurY=

解密结果是:123456

8.3

APKAppAPKAPK

8.3.1

AppJavaJavaAppAppAppApp“”

3apktooldex2jarjd-gui

apktoolAPKresAndroidManifest.xml

dex2jarAPKclasses.dexJARJARApp

jd-guidex2jarjarJava

APK(Window)

DOSapktool“apktool.bat d -fAPK”8-29

8-29apktool

apktoolresAndroidManifest.xmlres/layoutres/valuesres/drawable

(Winrar)APKAPKWinrarAPK8-30

8-30Apk

APKclasses.dexdex2jard2j-dex2jar.bat classes.dex
classes_dex2jar.jarJARApp

jd-gui.execlasses_dex2jar.jarjd-guiJARJavaJava8-31



8-31java

jd-guiFile → Save All SourceszipzipJava

JavaresApp

8.3.2

AppAppAndroid StudioProGuard

(1)APK

(2)

Android Studioproguard-rules.probuild.gradle
android → buildTypes → release

minifyEnabled false

proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'

Android StudiominifyEnabledfalsetrueproguard-rules.pro

proguard-rules.pro



5

(1)onClick

(2)JARkeep classdontwarn

(3)JAR“(”“””

(4)jniso

(5)WebViewsguard-rules.proJavajs



APKJava8-32

abcdabcd



8-32Java

8.3.3

AppAppAndroid StudioAPK
360<http://jiagu.360.cn/8-33>



8-33360

“”“”8-34



8-34

“”8-35



8-35

“”“”test-release.encrypted.apk

APKAPKjarsignerzipalign

1. jarsigner

jarsignerJavaJARJavajdk/bin/jarsigner.exe“jarsigner -verbose -keystore
-storepass-keypass -digestalg SHA1 -sigalg MD5withRSA -signedjar

”

2. zipalign

zipalignAndroidSDKAPKSDKbuild-tools//zipalign.exe“zipalign -v
4”

APK



test-release-signed.apktest-release-signed-align.apk

APKSignAPKSign8-36

APKSignAPK8-37“”



8-36



8-37

8.4

AppAppApp

8.4.1

APKApp360App



8-38

App<http://op.open.qq.com/QQQQ8-38>

“”“”“”8-398-40

“”

姓名	<input type="text" value="请填写与有效身份证上一致的姓名"/>
个人证件	<input checked="" type="radio"/> 身份证 <input type="radio"/> 护照
	<input type="text" value="请填写18位身份证号码"/>
手持身份证照片	<div style="border: 1px solid gray; width: 100px; height: 100px; display: flex; align-items: center; justify-content: center; margin: 0 auto;"></div> <p>上传图片示例</p> <ol style="list-style-type: none">1.2M以内，JPG/PNG格式的图片2.开发者手持身份证/护照正面进行拍照，要求五官可见，证件信息清晰无遮挡3.目前我们只针对境外开发者开放使用本人出境手持护照截图，如您属于中国大陆国籍的必须手持身份证截图4.我司资质验证系统已与工商部门联网，请勿提供虚假证件
<input type="button" value="返回上一步"/> <input type="button" value="下一步"/>	

手机号码

同意接收审核通知短信

我们会第一时间将审核结果通知您，并保证不会对外公布此号码

电子邮箱

联系地址

同意接受 腾讯开放平台开发者协议
我同意承担以QQ为标识进行的全部行为的 法律责任
收听官方微博 腾讯开放平台

[返回上一步](#) [下一步](#)

8-40

8.4.2

“”8-41

APP ID 1105789090 [关联QQ互联](#)

APP KEY CT7IzKvEHXJsg2uH

选择平台



移动应用 安卓
应用接入后在应用宝市场分发

iOS

移动应用 iOS
应用接入只用于获取QQ登录能力



网页应用
应用接入QQ空间、3366等PC平台



机器人
创建自己的机器人

[创建应用](#)

8-41

“”“”“”8-42



8-42

“”8-43



8-43

8-44

安装包

* 安装包



* 发布类型 审核通过后立即发布 定时发布

图标素材

适配信息

版权证明 (可选)

允许平台为我的应用提供 最佳用户体验
页面内容已于 22点12分 自动保存

返回 保存 提交审核

8-44

“”1~3QQ8-45

应用名称	状态	上线时间	应用等级	日下载量	总下载量	操作
 岗位助手	已上线	2015-03-19 11:32:14	C	0	87	更新安装包

8-45

AppAppAppApp8-46



8-46App

“”App

8.5

App(APK)()()

AppApp“” → “” → “” → “”

5

(1)App

(2)App

(3)APK

(4)APK

(5)App

第9章 设备操作



App“”

9.1

SurfaceViewCamera()——TextureViewCamera2()

9.1.1 SurfaceView Android UI UI

SurfaceViewAndroidUI

UIonDrawdispatchDrawSurfaceHolderSurfaceViewgetHolder

SurfaceHolderSurfaceHolder

lockCanvas

unlockCanvasAndPost



surfaceCreated



surfaceChanged



surfaceDestroyed

addCallbackSurfaceHolder.Callback3

removeCallback

isCreatingSurfaceView

getSurface

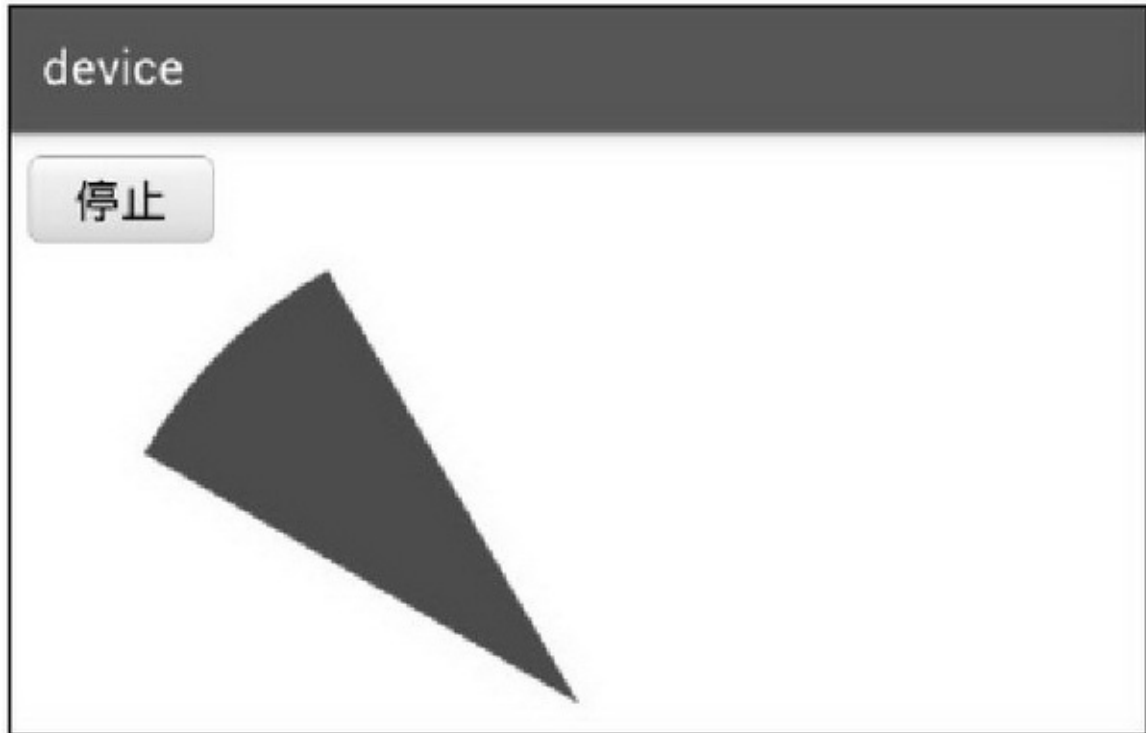
setFixedSize

setFormat

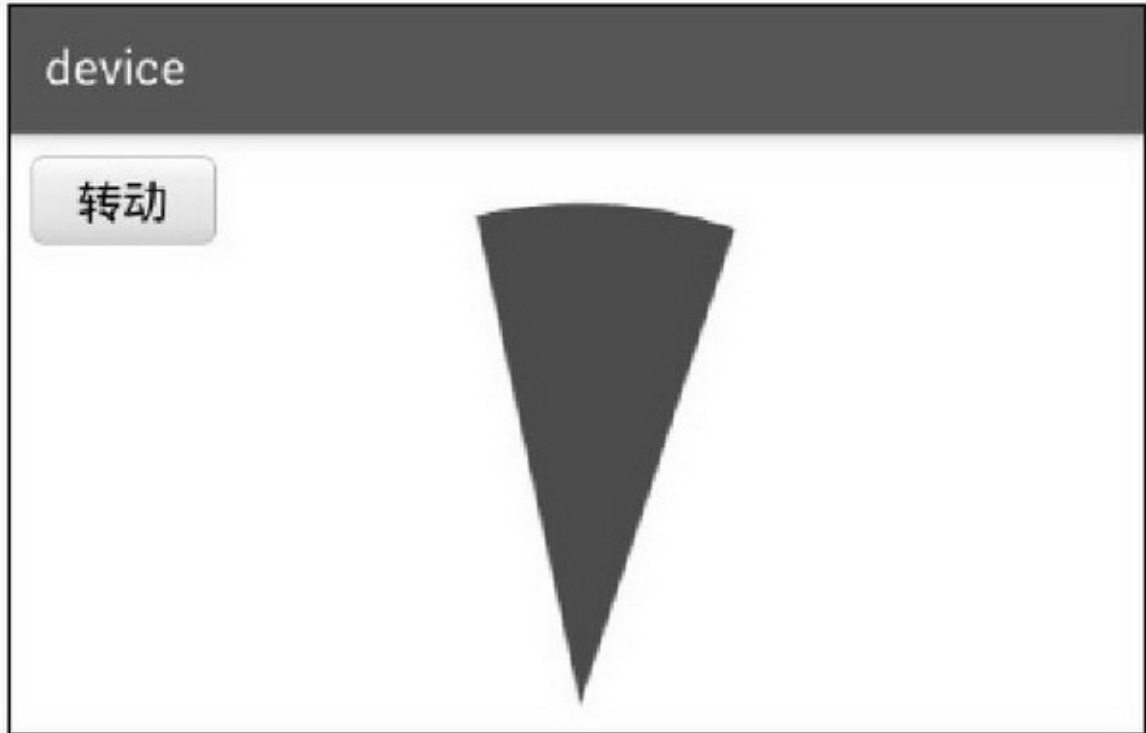
9-1

PixelFormat 类的绘图格式类型	说明
TRANSPAREN	透明
TRANSLUCENT	半透明
OPAQUE	不透明

9-19-2UI

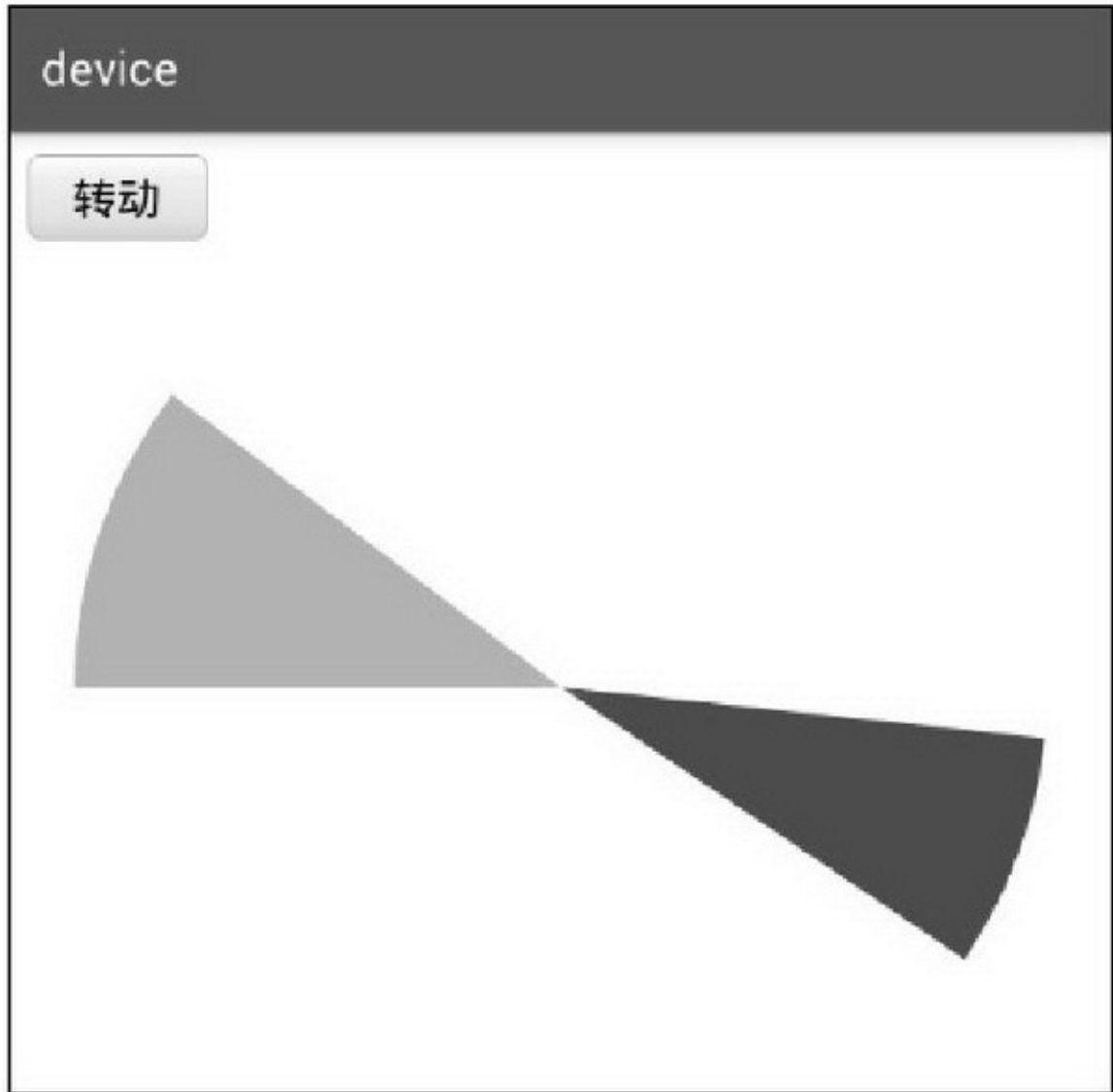


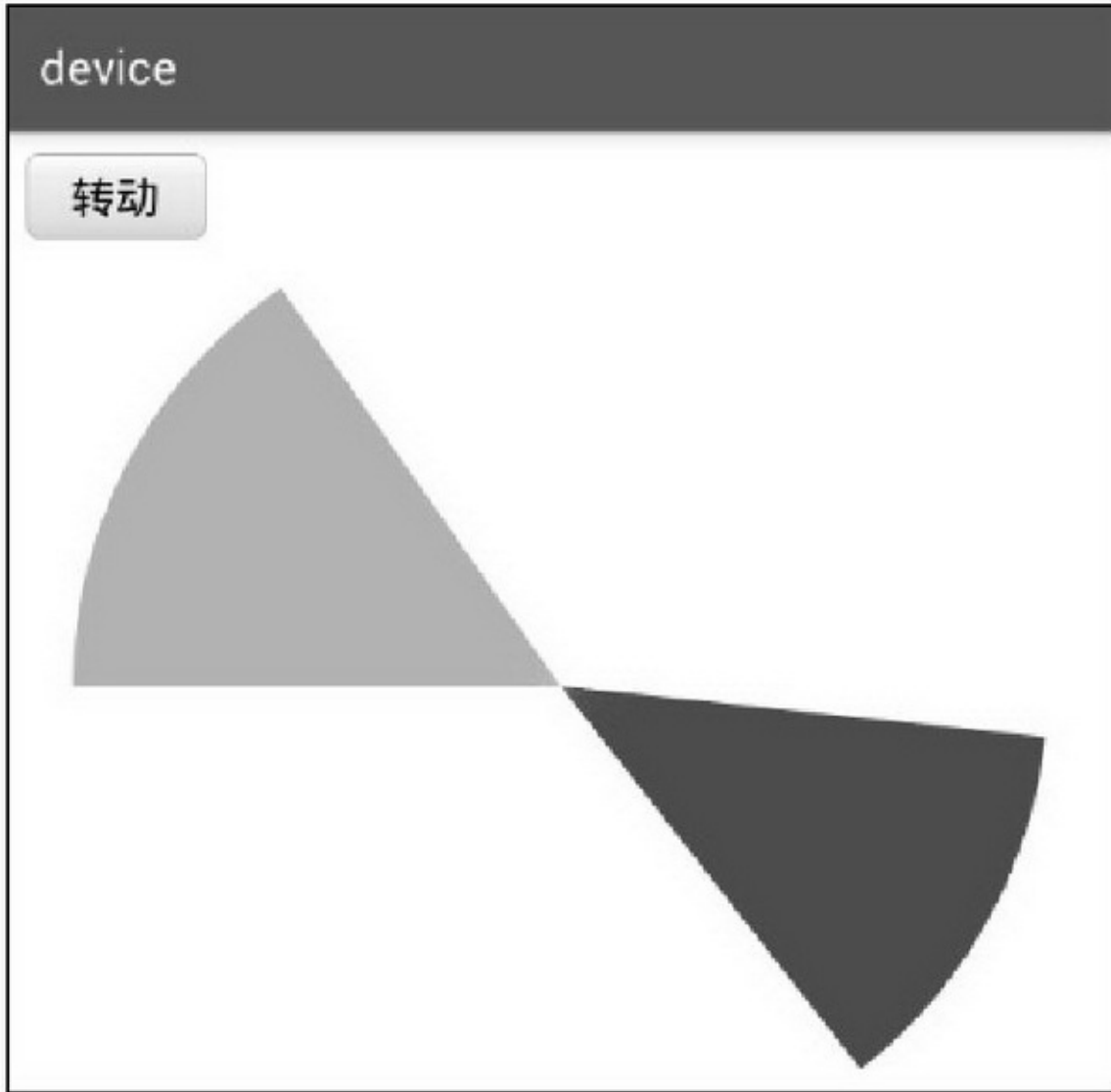
9-11



9-22

9-39-4





9-42

(1)UI

(2)

9.1.2Camera

Camera

getNumberOfCameras

openopen(0)open(1)
getParametersCamera.Parameters
➤ setPreviewSize
➤ setPictureSize
➤ setPictureFormatImageFormat.JPEGJPG
➤ setFocusModeCamera.Parameters.FOCUS_MODE_AUTO
setParametersCamera.Parameters
setPreviewDisplaySurfaceHolderSurfaceHolder.Callback
surfaceCreated
startPreviewsetPreviewDisplay
unlockstartPreview
setDisplayOrientationAndroid090
autoFocusAutoFocusCallbackonAutoFocus
takePictureShutterCallbackonShutter“”
PictureCallbacknullPictureCallbackJPGonPictureTaken
setZoomChangeListenerOnZoomChangeListeneronZoomChange
setPreviewCallbackPreviewCallbackonPreviewFrame
stopPreview
lockstopPreview
release
()


```

public CameraView(Context context, AttributeSet attrs) {
    super(context, attrs);
    mContext = context;
    mHolder = getHolder();
    mHolder.setFormat(PixelFormat.TRANSPARENT);
    mHolder.addCallback(mSurfaceCallback);
}

private String mPhotoPath;

//外部调用该方法获得拍摄照片的路径
public String getPhotoPath() {
    return mPhotoPath;
}

//外部调用该方法进行拍照动作
public void doTakePicture() {
    if(isPreviewing && mCamera!=null) {
        mCamera.takePicture(mShutterCallback, null, mPictureCallback);
    }
}

//快门按下的回调，在这里可以设置类似播放“咔嚓”声之类的操作。默认的是“咔嚓”
private ShutterCallback mShutterCallback = new ShutterCallback() {
    public void onShutter() {
        Log.d(TAG, "onShutter...");
    }
};

//获得拍照图片的回调。在此保存图片
private PictureCallback mPictureCallback = new PictureCallback() {
    public void onPictureTaken(byte[] data, Camera camera) {
        Log.d(TAG, "onPictureTaken...");
        Bitmap raw = null;
        if(null != data) {
            raw = BitmapFactory.decodeByteArray(data, 0, data.length);//将 data 解析成位图
            mCamera.stopPreview();
            isPreviewing = false;
        }
        Bitmap bitmap = BitmapUtil.getRotateBitmap(raw, (mCameraType==
CAMERA_BEHIND)?90:-90);

```

```

        mPhotoPath = String.format("%s%s.jpg", BitmapUtil.getCachePath(mContext),
Utils.getNowDateTime());
        BitmapUtil.saveBitmap(mPhotoPath, bitmap, "jpg", 80);
        Log.d(TAG, "bitmap.size="+bitmap.getByteCount()/1024+"K", path="+mPhotoPath);
        try {
            Thread.sleep(1000);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
        //再次进入预览
        mCamera.startPreview();
        isPreviewing = true;
    }
};

//预览界面状态变更时的回调
private SurfaceHolder.Callback mSurfaceCallback = new SurfaceHolder.Callback() {
    @Override
    public void surfaceCreated(SurfaceHolder holder) {
        // 当预览视图创建的时候开启相机
        mCamera = Camera.open(mCameraType);
        try {
            // 设置预览
            mCamera.setPreviewDisplay(holder);
            mCameraSize = MetricsUtil.getCameraSize(mCamera.getParameters(),
MetricsUtil.getSize(mContext));
            Log.d(TAG, "width="+mCameraSize.x+", height="+mCameraSize.y);
            Camera.Parameters parameters = mCamera.getParameters();
            // 设置预览大小
            parameters.setPreviewSize(mCameraSize.x, mCameraSize.y);
            // 设置图片保存时的分辨率大小
            parameters.setPictureSize(mCameraSize.x, mCameraSize.y);
            // 设置格式
            parameters.setPictureFormat(ImageFormat.JPEG);
            // 设置自动对焦。前置摄像头似乎无法自动对焦
            if (mCameraType == CameraView.CAMERA_BEHIND) {
                parameters.setFocusMode(Camera.Parameters.FOCUS_MODE_AUTO);
            }
            mCamera.setParameters(parameters);
        } catch (Exception e) {
            e.printStackTrace();
            mCamera.release();
            mCamera = null;
        }
    }
};

```

```
    }  
    return;  
}  
  
@Override  
public void surfaceChanged(SurfaceHolder holder, int format, int width, int height) {  
    mCamera.setDisplayOrientation(90);  
    mCamera.startPreview();  
    isPreviewing = true;  
    mCamera.autoFocus(null);  
    //setPreviewCallback 给连拍使用  
    mCamera.setPreviewCallback(mPreviewCallback);  
}  
  
@Override  
public void surfaceDestroyed(SurfaceHolder holder) {  
    mCamera.setPreviewCallback(null);  
    mCamera.stopPreview();  
    mCamera.release();  
    mCamera = null;  
}  
};
```

9-5



9-5Camera

setPreviewCallbackonPreviewFrame

```

private boolean bShooting = false;
private int shooting_num = 0;
private ArrayList<String> mShootingArray;

//外部调用该方法获得连拍照片的路径
public ArrayList<String> getShootingList() {
    Log.d(TAG, "mShootingArray.size()="+mShootingArray.size());
    return mShootingArray;
}

//外部调用该方法进行连拍动作
public void doTakeShooting() {
    mShootingArray = new ArrayList<String>();
    bShooting = true;
    shooting_num = 0;
}

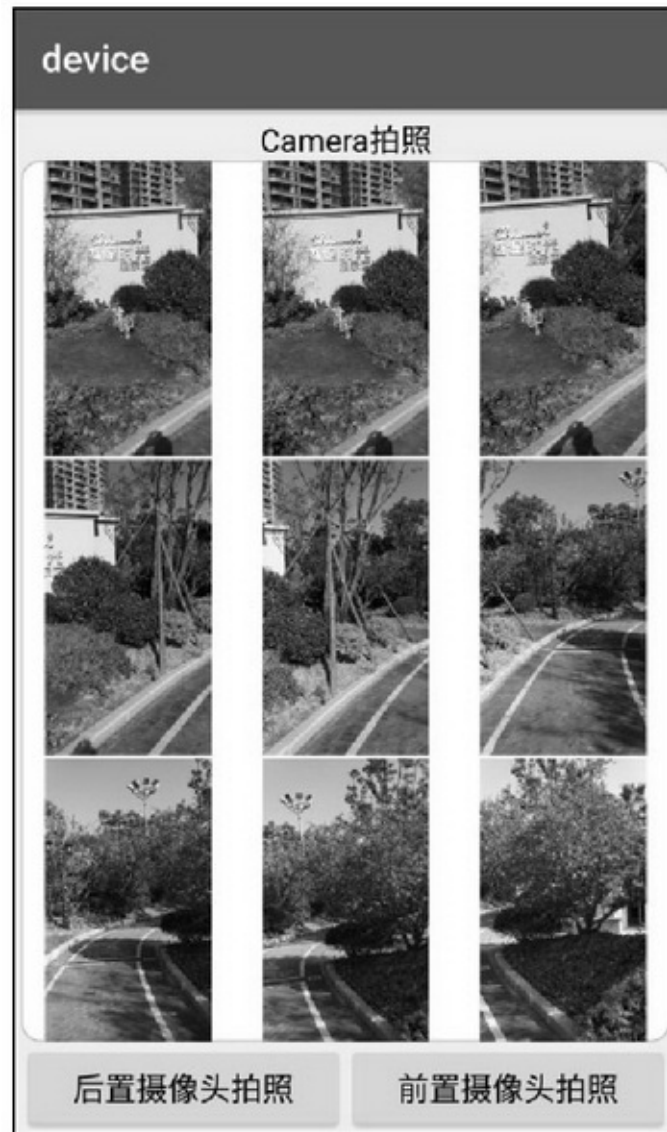
private PreviewCallback mPreviewCallback = new PreviewCallback() {
    @Override
    public void onPreviewFrame(byte[] data, Camera camera) {
        Log.d(TAG, "onPreviewFrame bShooting="+bShooting+", shooting_num="+
+shooting_num);
        if (!bShooting) {
            return;
        }
        Camera.Parameters parameters = camera.getParameters();
        int imageFormat = parameters.getPreviewFormat();
        int w = parameters.getPreviewSize().width;
        int h = parameters.getPreviewSize().height;
        Rect rect = new Rect(0, 0, w, h);
        YuvImage yuvImg = new YuvImage(data, imageFormat, w, h, null);
        try {
            ByteArrayOutputStream bos = new ByteArrayOutputStream();
            yuvImg.compressToJpeg(rect, 80, bos);
            Bitmap raw = BitmapFactory.decodeByteArray(bos.toByteArray(), 0, bos.size());
            Bitmap bitmap = BitmapUtil.getRotateBitmap(raw, (mCameraType==
CAMERA_BEHIND)?90:-90);
            String path = String.format("%s%s.jpg", BitmapUtil.getCachePath(mContext),
Utils.getNowDateTimeFull());
            BitmapUtil.saveBitmap(path, bitmap, "jpg", 80);

```

```

        Log.d(TAG, "bitmap.size="+(bitmap.getByteCount()/1024)+"K"+", path="+path);
        //再次进入预览
        camera.startPreview();
        shooting_num++;
        mShootingArray.add(path);
        if (shooting_num > 8) { //每次连拍 9 张
            bShooting = false;
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}
};

```



9-6Camera

9.1.3 TextureView SurfaceView onDraw dispatchDraw

ViewsetAlpha setTranslation setScale setRotation

Android 4.0 TextureView TextureView
setBackground

lockCanvas

unlockCanvasAndPost

 onSurfaceTextureAvailable

 onSurfaceTextureSizeChanged

 onSurfaceTextureDestroyed

 onSurfaceTextureUpdated

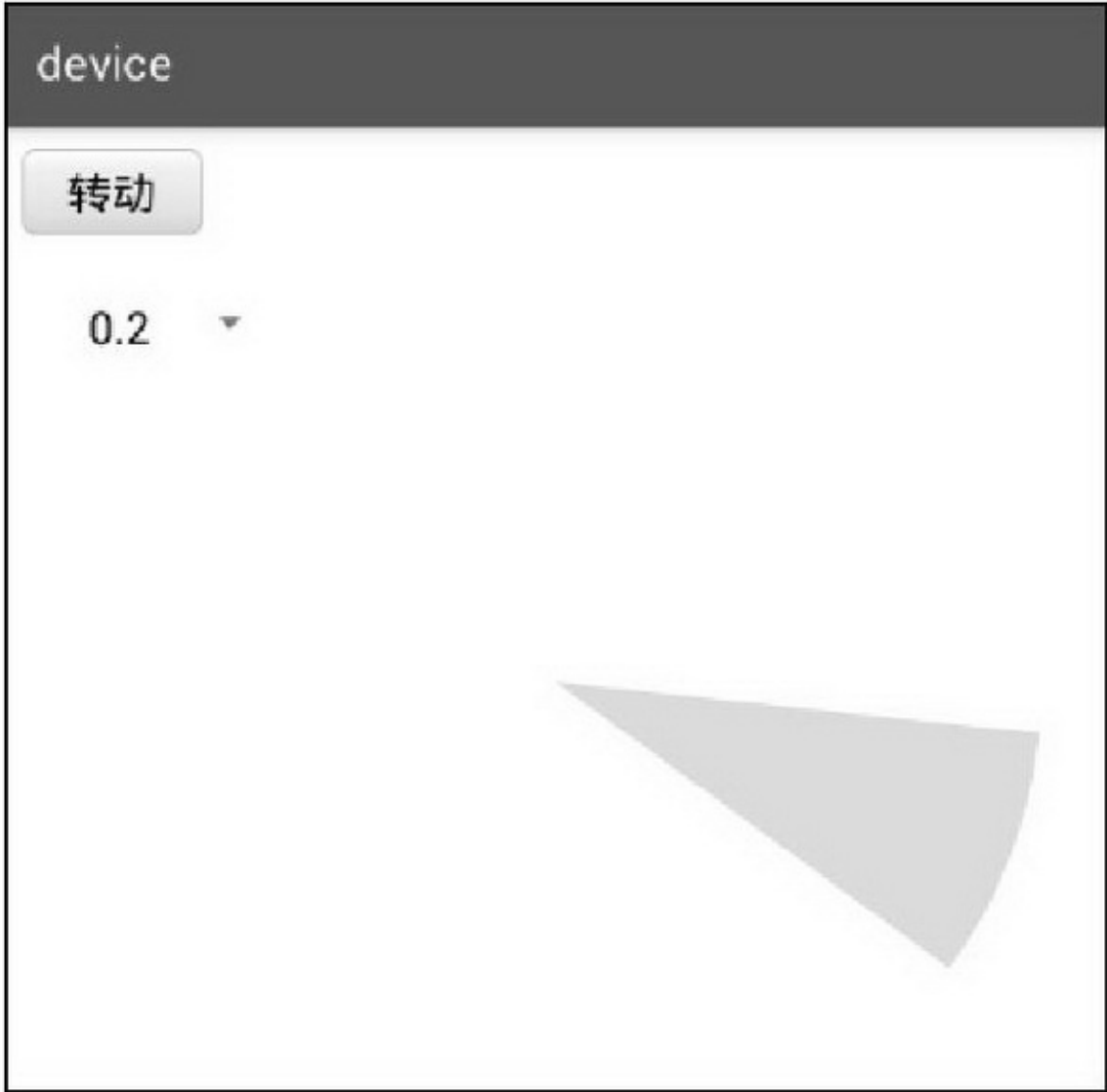
setSurfaceTextureListenerSurfaceHolderaddCallback

SurfaceTextureListener4

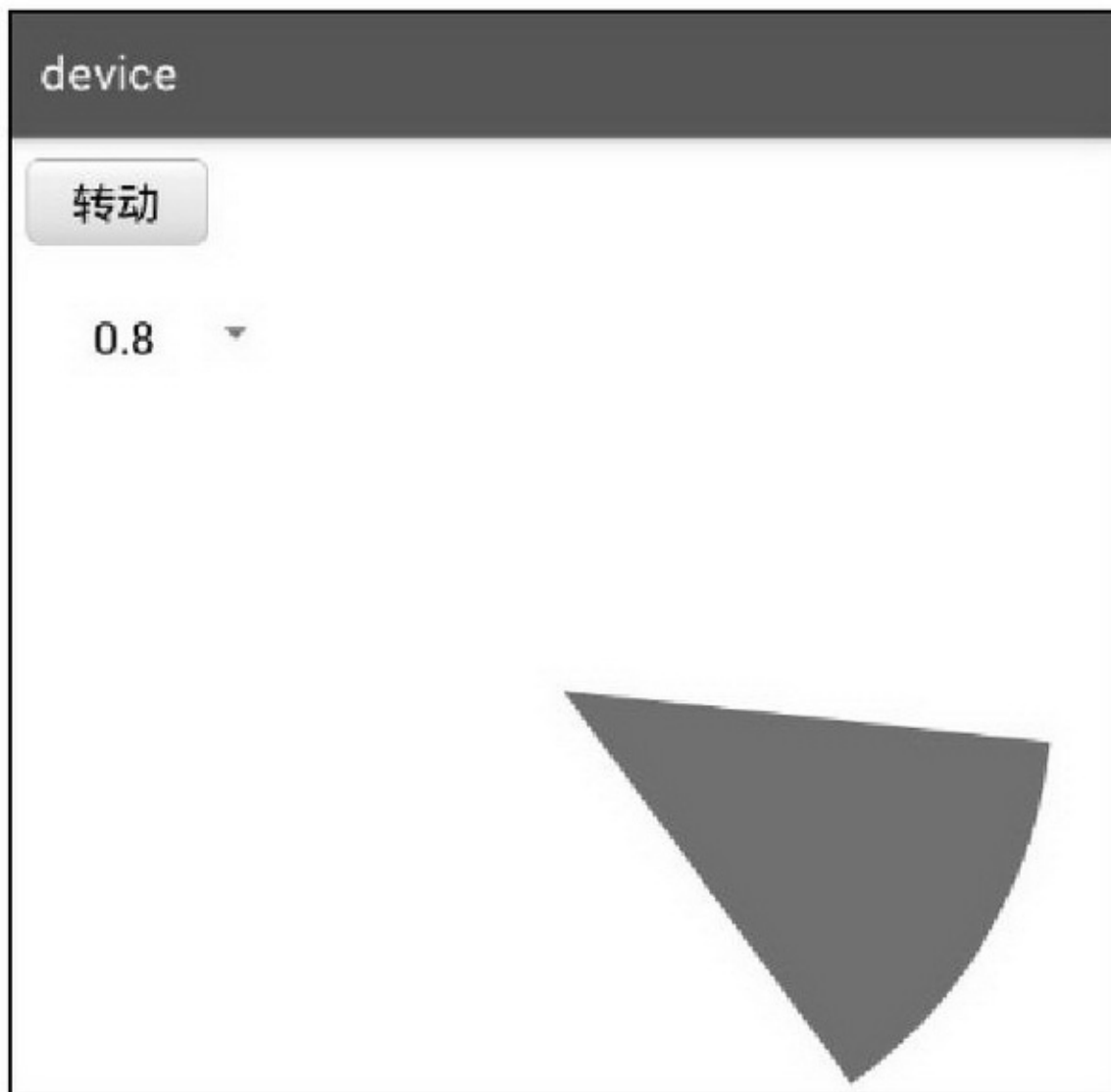
isAvailable

getSurfaceTexture

9-70.29-80.8



9-70.2



9-80.8

```
TextureView.setBackgroundSurfaceView  
setZOrderOnTop(true);  
mHolder.setFormat(PixelFormat.TRANSLUCENT);
```

9.1.4 Camera 2

Android 5.0 Camera——camera 2 Android camera 25
(1)30

(2)

(3)

(4)

(5)

camera2CameraCameraManagerCameraDevice

CameraCaptureSessionImageReader

1.CameraManager

CAMERA_SERVICE

getCameraIdList

getCameraCharacteristics

camera2Android5.0

INFO_SUPPORTED_HARDWARE_LEVEL_LEGACYAppcamera2

CameraCharacteristics cc = cm.getCameraCharacteristics(cameraid); //

CameraCharacteristics.INFO_SUPPORTED_HARDWARE_LEVEL_FULL

//

CameraCharacteristics.INFO_SUPPORTED_HARDWARE_LEVEL_LIMITED

//

CameraCharacteristics.INFO_SUPPORTED_HARDWARE_LEVEL_LEGACY

int level =

cc.get(CameraCharacteristics.INFO_SUPPORTED_HARDWARE_LEVEL);

openCameraidCameraDevice.StateCallbackonOpened(

CameraDevicecreateCaptureRequest)

setTorchModetruefalse

2.CameraDevice

createCaptureRequestCameraCaptureSession.StateCallback

onConfigured(CameraCaptureSessionsetRepeatingRequest)

createCaptureRequestCaptureRequest

close

3.CameraCaptureSession

00

getDevice

captureCaptureRequestImageReader

setRepeatingRequestCaptureRequestImageReader

stopRepeating

4.ImageReader

onImageAvailable

getSurface

setOnImageAvailableListener

ImageReader.OnImageAvailableListeneronImageAvailable

Camera camera 2

camera 29-99-109-99-10

device

新版Camera2拍照



后置摄像头拍照

前置摄像头拍照

9-9Camera2

device

新版Camera2拍照



后置摄像头拍照

前置摄像头拍照

9.2

SeekBarMediaRecorderMediaPlayer9.1

9.2.1SeekBar SeekBarProgressBar

SeekBar4

setThumb

setThumbOffset

setKeyProgressIncrement



onProgressChanged3truefalse



onStartTrackingTouch



onStopTrackingTouch

setOnSeekBarChangeListenerOnSeekBarChangeListener3

```

public class SeekbarActivity extends AppCompatActivity implements OnSeekBarChangeListener {
    private SeekBar sb_progress;
    private TextView tv_progress;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_seekbar);
        tv_progress = (TextView) findViewById(R.id.tv_progress);
        sb_progress = (SeekBar) findViewById(R.id.sb_progress);
        sb_progress.setOnSeekBarChangeListener(this);
        sb_progress.setProgress(50);
    }

    @Override
    public void onProgressChanged(SeekBar seekBar, int progress, boolean fromUser) {
        String desc = "当前进度为 : "+seekBar.getProgress()+"，最大进度为"+seekBar.getMax();
        tv_progress.setText(desc);
    }

    @Override
    public void onStartTrackingTouch(SeekBar seekBar) {
    }

    @Override
    public void onStopTrackingTouch(SeekBar seekBar) {
    }
}

```

9-119-129-11509-1273



9-11SeekBar



9-12SeekBar

9.2.2

Android69-2

9-2

AudioManager 类的铃声类型	铃声名称	说明
STREAM_VOICE_CALL	通话音	
STREAM_SYSTEM	系统音	
STREAM_RING	铃声	来电与收短信的铃声
STREAM_MUSIC	媒体音	音频、视频、游戏等的声音
STREAM_ALARM	闹钟音	
STREAM_NOTIFICATION	通知音	

AudioManager.AUDIO_SERVICE.AudioManager

getStreamMaxVolume

getStreamVolume

getRingerMode

9-3

AudioManager 类的响铃模式	说明
RINGER_MODE_NORMAL	正常
RINGER_MODE_SILENT	静音
RINGER_MODE_VIBRATE	震动

setStreamVolume

setRingerMode9-3

adjustStreamVolume9-4FLAG_PLAY_SOUND

9-4

AudioManager 类的音量调整方向	说明
ADJUST_RAISE	调大一级
ADJUST_LOWER	调小一级
ADJUST_SAME	保持不变
ADJUST_MUTE	静音
ADJUST_UNMUTE	取消静音
ADJUST_TOGGLE_MUTE	静音取反，即原来不是静音就设置静音，原来是静音就取消静音



9-13

setStreamVolumeadjustStreamVolumesetStreamVolume
adjustStreamVolume
9-13

9.2.3

AndroidMediaRecorderMediaPlayer
1.MediaRecorder

MediaRecorderAndroid
 MediaRecorder()
 reset
 prepare
 start
 stop
 release
 setOnErrorListenerMediaRecorder.OnErrorListeneronError
 setOnInfoListenerMediaRecorder.OnInfoListeneronInfo
 setMaxDuration
 setMaxFileSize
 setOutputFile
 MediaRecorder()
 setAudioSourceAudioSource.MIC
 setOutputFormat9-5

9-5

OutputFormat 类的输出格式	格式分类	扩展名	格式说明
AMR_NB	音频	.amr	窄带格式
AMR_WB	音频	.amr	宽带格式
AAC_ADTS	音频	.aac	高级的音频传输流格式
MPEG_4	视频	.mp4	MPEG4 格式
THREE_GPP	视频	.3gp	3GP 格式

setAudioEncoder9-6setOutputFormatsetAudioEncoder
 called in an invalid state(2)

9-6

AudioEncoder 类的音频编码器	说明
AMR_NB	窄带编码
AMR_WB	宽带编码
AAC	低复杂度的高级编码
HE_AAC	高效率的高级编码
AAC_ELD	高效率的高级编码

setAudioSamplingRate(kHz)AMR_NB8kHzAMR_WB16kHz

setAudioChannels12

setAudioEncodingBitRate

MediaRecorder

```

public class AudioRecorder extends LinearLayout implements OnErrorListener,
    OnInfoListener, OnCheckedChangeListener {
    private MediaRecorder mMediaRecorder;
    private ProgressBar pb_record;
    private CheckBox ck_record;
    private Timer mTimer; // 计时器
    private int mRecordMaxTime = 10; // 一次录音最长时间
    private int mTimeCount; // 时间计数
    private String mRecordFilePath;

    public AudioRecorder(Context context) {
        this(context, null);
    }

    public AudioRecorder(Context context, AttributeSet attrs) {
        this(context, attrs, 0);
    }

    public AudioRecorder(Context context, AttributeSet attrs, int defStyleAttr) {
        super(context, attrs, defStyleAttr);
        LayoutInflater.from(context).inflate(R.layout.audio_recorder, this);
        pb_record = (ProgressBar) findViewById(R.id.pb_record);
        pb_record.setMax(mRecordMaxTime);
        ck_record = (CheckBox) findViewById(R.id.ck_record);
        ck_record.setOnCheckedChangeListener(this);
    }

    // 开始录音
    public void start() {
        mRecordFilePath = MediaUtil.getRecordFilePath("RecordAudio", ".amr");
        try {
            initRecord();
            mTimeCount = 0;
            mTimer = new Timer();
            mTimer.schedule(new TimerTask() {
                @Override
                public void run() {
                    pb_record.setProgress(mTimeCount++);
                }
            }, 0, 1000);
        } catch (Exception e) {

```

```
        e.printStackTrace();
    }
}

// 停止录音
public void stop() {
    if (mOnRecordFinishListener != null) {
        mOnRecordFinishListener.onRecordFinish();
    }
    stopRecord();
    releaseRecord();
}

// 获得最近一次的录音文件路径
public String getRecordFilePath() {
    return mRecordFilePath;
}

private void initRecord() {
    mMediaRecorder = new MediaRecorder();
    mMediaRecorder.setOnErrorListener(this);
    mMediaRecorder.setOnInfoListener(this);
    mMediaRecorder.setAudioSource(AudioSource.MIC); // 音频源
    mMediaRecorder.setOutputFormat(OutputFormat.AMR_NB);
    mMediaRecorder.setAudioEncoder(AudioEncoder.AMR_NB); // 音频格式
    mMediaRecorder.setMaxDuration(10 * 1000); // 设置录制时长
    mMediaRecorder.setOutputFile(mRecordFilePath);
    try {
        mMediaRecorder.prepare();
        mMediaRecorder.start();
    } catch (Exception e) {
        e.printStackTrace();
    }
}

private void stopRecord() {
    pb_record.setProgress(0);
    if (mTimer != null) {
        mTimer.cancel();
    }
    if (mMediaRecorder != null) {
        mMediaRecorder.setOnErrorListener(null);
        mMediaRecorder.setPreviewDisplay(null);
    }
}
```

```

        try {
            mMediaRecorder.stop();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

private void releaseRecord() {
    if (mMediaRecorder != null) {
        mMediaRecorder.setOnErrorListener(null);
        mMediaRecorder.release();
        mMediaRecorder = null;
    }
}

private OnRecordFinishListener mOnRecordFinishListener; // 录制完成回调接口
public interface OnRecordFinishListener {
    public void onRecordFinish();
}

public void setOnRecordFinishListener(OnRecordFinishListener listener) {
    mOnRecordFinishListener = listener;
}

@Override
public void onError(MediaRecorder mr, int what, int extra) {
    if (mr != null) {
        mr.reset();
    }
}

@Override
public void onInfo(MediaRecorder mr, int what, int extra) {
    if (what == MediaRecorder.MEDIA_RECORDER_INFO_MAX_DURATION_REACHED
        || what == MediaRecorder.MEDIA_RECORDER_INFO_MAX_FILESIZE_REACHED) {
        ck_record.setChecked(false);
    }
}

@Override
public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
    if (buttonView.getId() == R.id.ck_record) {

```

```

        if (isChecked == true) {
            ck_record.setText("停止录制");
            start();
        } else {
            ck_record.setText("开始录制");
            stop();
        }
    }
}
}
}

```

AndroidManifest.xml(SDSD)

<!--/ --> <uses-permission

```
android:name="android.permission.CAMERA" /> <uses-permission
android:name="android.permission.RECORD_VIDEO"/> <uses-permission
android:name="android.permission.RECORD_AUDIO" /> <!-- SD -->
    <uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission
android:name="android.permission.READ_EXTERNAL_STORAGE" /> <uses-
permission
android:name="android.permission.MOUNT_UNMOUNT_FILESYSTEMS" />
```

2. MediaPlayer

```
MediaPlayerAndroidMediaRecorder9-5MP3WAVMID
OGG
MediaPlayer()
reset
prepare
start
pause
stop
setOnPreparedListenerMediaPlayer.OnPreparedListeneronPrepared
setOnCompletionListenerMediaPlayer.OnCompletion-Listener
onCompletion
setOnSeekCompleteListenerMediaPlayer.OnSeek-CompleteListener
onSeekComplete
createUri
setDataSourcecreatesetDataSource
setVolume0~1
setAudioStreamType9-2
setLoopingtruefalse
isPlaying
```

seekToSeekBar

getCurrentPosition

getDuration

MediaPlayer

```
public class AudioPlayer extends LinearLayout implements OnCompletionListener, OnCheckedChangeListener {
    private static final String TAG = "AudioPlayer";
    private MediaPlayer mMediaPlayer;
    private ProgressBar pb_play;
    private CheckBox ck_play;
    private Timer mTimer; // 计时器
    private String mAudioPath;
    private boolean bFinish = true;

    public AudioPlayer(Context context) {
        this(context, null);
    }

    public AudioPlayer(Context context, AttributeSet attrs) {
        this(context, attrs, 0);
    }

    public AudioPlayer(Context context, AttributeSet attrs, int defStyleAttr) {
        super(context, attrs, defStyleAttr);
        LayoutInflater.from(context).inflate(R.layout.audio_player, this);
        pb_play = (ProgressBar) findViewById(R.id.pb_play);
        ck_play = (CheckBox) findViewById(R.id.ck_play);
        ck_play.setOnCheckedChangeListener(this);
    }

    // 初始化音频文件路径与播放器
    public void init(String path) {
        mAudioPath = path;
        ck_play.setEnabled(true);
        ck_play.setTextColor(Color.BLACK);
        mMediaPlayer = new MediaPlayer();
        mMediaPlayer.setOnCompletionListener(this);
    }
}
```



```

private void play() {
    try {
        mMediaPlayer.reset();
        mMediaPlayer.setAudioStreamType(AudioManager.STREAM_MUSIC);
        Log.d(TAG, "audio path = " + mAudioPath);
        // 录制完毕要等一秒钟再调用 setDataSource 方法，因为此时可能尚未完成写入
        mMediaPlayer.setDataSource(mAudioPath);
        mMediaPlayer.prepare();
        mMediaPlayer.start();
        pb_play.setMax(mMediaPlayer.getDuration());
        mTimer = new Timer();
        mTimer.schedule(new TimerTask() {
            @Override
            public void run() {
                pb_play.setProgress(mMediaPlayer.getCurrentPosition());
            }
        }, 0, 1000);
    } catch (Exception e) {
        e.printStackTrace();
    }
}

@Override
public void onCompletion(MediaPlayer mp) {
    bFinish = true;
    ck_play.setChecked(false);
    if (mTimer != null) {
        mTimer.cancel();
    }
}

@Override
public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
    if (buttonView.getId() == R.id.ck_play) {
        if (isChecked == true) {
            ck_play.setText("暂停播放");
            if (bFinish == true) {
                play();
            } else {
                mMediaPlayer.start();
            }
        }
        bFinish = false;
    }
}

```

```

    } else {
        ck_play.setText("开始播放");
        mMediaPlayer.pause();
    }
}
}
}
}

```

9-149-159-149-15



9-14



9-15

9.2.4

AndroidMediaRecorderMediaPlayerMediaRecorderMediaPlayer
MediaRecorderMediaPlayer
1.MediaRecorder()

MediaRecorder()
setCamera
setPreviewDisplaySurfaceHoldergetSurface
setOrientationHint9090
setVideoSourceVideoSource.CAMERA
setOutputFormat9-5
setVideoEncoderVideoEncoder.MPEG_4_SPMPEG4



setOutputFormatjava.lang.IllegalStateException
setVideoSize
setVideoFrameRate
setVideoEncodingBitRatesetVideoFrameRate
setVideoEncodingBitRate
SurfaceViewSurfaceHolderCamera

```
private SurfaceHolder mHolder;
private SurfaceHolder.Callback mSurfaceCallback = new SurfaceHolder.Callback() {
    @Override
    public void surfaceCreated(SurfaceHolder holder) {
        initCamera();
    }

    @Override
    public void surfaceChanged(SurfaceHolder holder, int format, int width, int height) {
    }

    @Override
    public void surfaceDestroyed(SurfaceHolder holder) {
        freeCamera();
    }
};

private void initCamera() {
    if (mCamera != null) {
        freeCamera();
    }
    try {
        mCamera = Camera.open();
        mCamera.setDisplayOrientation(90);
        mCamera.setPreviewDisplay(mHolder);
        mCamera.startPreview();
        mCamera.unlock();
    } catch (Exception e) {
        e.printStackTrace();
        freeCamera();
    }
}

private void freeCamera() {
    if (mCamera != null) {
        mCamera.stopPreview();
    }
}
```

```

        mCamera.lock();
        mCamera.release();
        mCamera = null;
    }
}

private void initRecord() {
    mHolder = sv_record.getHolder();
    mHolder.addCallback(mSurfaceCallback);
}

private void startRecord() {
    mMediaRecorder = new MediaRecorder();
    mMediaRecorder.setCamera(mCamera);
    mMediaRecorder.setOnErrorListener(this);
    mMediaRecorder.setOnInfoListener(this);
    mMediaRecorder.setPreviewDisplay(mHolder.getSurface());
    mMediaRecorder.setVideoSource(VideoSource.CAMERA); // 视频源
    mMediaRecorder.setAudioSource(AudioSource.MIC); // 音频源
    mMediaRecorder.setOutputFormat(OutputFormat.MPEG_4); // 视频输出格式
    mMediaRecorder.setAudioEncoder(AudioEncoder.AMR_NB); // 音频格式
    // 如果录像报错 : MediaRecorder start failed: -19
    // 可注释 setVideoSize 和 setVideoFrameRate, 因为尺寸设置必须为摄像头所支持, 否则报错
    mMediaRecorder.setVideoEncodingBitRate(1 * 1024 * 512); // 设置帧频率
    mMediaRecorder.setOrientationHint(90); // 输出旋转 90 度, 保持全屏录制
    mMediaRecorder.setVideoEncoder(VideoEncoder.MPEG_4_SP); // 视频录制格式
    mMediaRecorder.setMaxDuration(mRecordMaxTime * 1000); // 设置录制时长
    // setMaxFileSize 与 setMaxDuration 设置一个即可
    mMediaRecorder.setOutputFile(mRecordFilePath);
    try {
        mMediaRecorder.prepare();
        mMediaRecorder.start();
    } catch (Exception e) {
        e.printStackTrace();
    }
}

private void stopRecord() {
    pb_record.setProgress(0);
    if (mTimer != null) {
        mTimer.cancel();
    }
    if (mMediaRecorder != null) {
        mMediaRecorder.setOnErrorListener(null);
    }
}

```

```

        mMediaRecorder.setPreviewDisplay(null);
        try {
            mMediaRecorder.stop();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

private void releaseRecord() {
    if (mMediaRecorder != null) {
        mMediaRecorder.setOnErrorListener(null);
        mMediaRecorder.release();
        mMediaRecorder = null;
    }
}
}

```

2. MediaPlayer()

MediaPlayer()

setDisplaySurfaceHolder

setSurfaceSurfaceHoldergetSurfacesetDisplaysetSurface

setScreenOnWhilePlayingSurfaceHoldertruesetDisplay

setSurface

setVideoScalingModeMediaPlayer.VIDEO_SCALING_
 MODE_SCALE_TO_FIT

setOnVideoSizeChangeListenerMediaPlayer.

OnVideoSizeChangeListeneronVideoSizeChanged

SurfaceViewSurfaceViewSurfaceViewplay

```

private void play() {
    try {
        mMediaPlayer.reset();
        mMediaPlayer.setAudioStreamType(AudioManager.STREAM_MUSIC);
        Log.d(TAG, "video path = " + mVideoPath);
        // 录制完毕要等一秒钟再 setDataSource, 因为有可能此时视频文件尚未完全写入
        // 否则会报异常 java.io.IOException: setDataSourceFD failed
        mMediaPlayer.setDataSource(mVideoPath);
        // 把视频界面输出到 SurfaceView
        mMediaPlayer.setDisplay(sv_play.getHolder());
        mMediaPlayer.prepare();
        mMediaPlayer.start();
        pb_play.setMax(mMediaPlayer.getDuration());
        mTimer = new Timer();
        mTimer.schedule(new TimerTask() {
            @Override
            public void run() {
                pb_play.setProgress(mMediaPlayer.getCurrentPosition());
            }
        }, 0, 1000);
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

9-169-179-169-17



device



开始录制



暂停播放

9.3

Android

9.3.1

SensorAndroidAndroid

Android9-7

编号	Sensor 类的传感器类型	传感器名称	说明
1	TYPE_ACCELEROMETER	加速度	常用于摇一摇功能
2	TYPE_MAGNETIC_FIELD	磁场	
3	TYPE_ORIENTATION	方向	已弃用，取而代之的是 getOrientation 方法
4	TYPE_GYROSCOPE	陀螺仪	用来感应手机的旋转和倾斜
5	TYPE_LIGHT	光线	用来感应手机正面的光线强弱
6	TYPE_PRESSURE	压力	用来感应气压
7	TYPE_TEMPERATURE	温度	已弃用，取而代之的是类型 13
8	TYPE_PROXIMITY	距离	
9	TYPE_GRAVITY	重力	
10	TYPE_LINEAR_ACCELERATION	线性加速度	
11	TYPE_ROTATION_VECTOR	旋转矢量	
12	TYPE_RELATIVE_HUMIDITY	相对湿度	
13	TYPE_AMBIENT_TEMPERATURE	环境温度	
14	TYPE_MAGNETIC_FIELD_UNCALIBRATED	无标定磁场	
15	TYPE_GAME_ROTATION_VECTOR	无标定旋转矢量	
16	TYPE_GYROSCOPE_UNCALIBRATED	未校准陀螺仪	
17	TYPE_SIGNIFICANT_MOTION	特殊动作	
18	TYPE_STEP_DETECTOR	步行检测	用户每走一步就触发一次事件
19	TYPE_STEP_COUNTER	步行计数	记录激活后的步伐数
20	TYPE_GEOMAGNETIC_ROTATION_VECTOR	地磁旋转矢量	
21	TYPE_HEART_RATE	心跳速率	可穿戴设备使用，如手环
22	TYPE_TILT_DETECTOR	倾斜检测	
23	TYPE_WAKE_GESTURE	唤醒手势	
24	TYPE_GLANCE_GESTURE	掠过手势	
25	TYPE_PICK_UP_GESTURE	拾起手势	

SensorManager.getSensorList

```

private void showSensorInfo() {
    mSensroMgr = (SensorManager) getSystemService(Context.SENSOR_SERVICE);
    List<Sensor> sensorList = mSensroMgr.getSensorList(Sensor.TYPE_ALL);
    String show_content = "当前支持的传感器包括 : \n";
    for (Sensor sensor : sensorList) {
        if (sensor.getType() >= mSensorType.length) {
            continue;
        }
        mapSensor.put(sensor.getType(), sensor.getName());
    }
    for (Map.Entry<Integer, String> item_map : mapSensor.entrySet()) {
        int type = item_map.getKey();
        String name = item_map.getValue();
        String content = String.format("%d %s : %s\n", type, mSensorType[type-1], name);
        show_content += content;
    }
    tv_sensor.setText(show_content);
}
}

```

9-18Android

device
当前支持的传感器包括:
8 距离: Proximity Sensor
11 旋转矢量: LSM6DB0 iNemoEngine Rotation Vector Sensor
22 倾斜检测: LSM6DB0 Tilt Sensor
19 计步器: LSM6DB0 Step Counter Sensor
5 光线: Ambient Light Sensor
3 方向: LSM6DB0 iNemoEngine Orientation Sensor
4 陀螺仪: LSM6DB0 Gyroscope Sensor
1 加速度: LSM6DB0 Accelerometer Sensor
15 无标定旋转矢量: LSM6DB0 iNemoEngine Game Rotation Vector Sensor
18 步行检测: LSM6DB0 Step Detector Sensor
9 重力: LSM6DB0 iNemoEngine Gravity Sensor
17 特殊动作: LSM6DB0 Significant Motion Sensor
10 线性加速度: LSM6DB0 iNemoEngine Linear Acceleration Sensor
14 无标定磁场: LSM6DB0 Magnetometer Sensor
16 未校准陀螺仪: LSM6DB0 Gyroscope Sensor
20 地磁旋转矢量: LSM6DB0 iNemoEngine GeoMagnetic Rotation Vector Sensor
2 磁场: LSM6DB0 Magnetometer Sensor

9.3.2

步骤 01 `SensorManager.SENSOR_SERVICE`

步骤 02 `Activity.onResume`

```
mSensroMgr.registerListener(this,  
    mSensroMgr.getDefaultSensor(Sensor.TYPE_ACCELEROMETER),  
    SensorManager.SENSOR_DELAY_NORMAL);
```

步骤 03 `Activity.onPause`

`mSensroMgr.unregisterListener(this);` 步骤 04
`SensorEventListener.onSensorChanged`

```

public class AccelerationActivity extends AppCompatActivity implements SensorEventListener {
    private TextView tv_shake;
    private SensorManager mSensroMgr;
    private Vibrator mVibrator;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_acceleration);
        tv_shake = (TextView) findViewById(R.id.tv_shake);
        mSensroMgr = (SensorManager) getSystemService(Context.SENSOR_SERVICE);
        mVibrator = (Vibrator) getSystemService(Context.VIBRATOR_SERVICE);
    }

    @Override
    protected void onPause() {
        super.onPause();
        mSensroMgr.unregisterListener(this);
    }

    @Override
    protected void onResume() {
        super.onResume();
        mSensroMgr.registerListener(this,
            mSensroMgr.getDefaultSensor(Sensor.TYPE_ACCELEROMETER),
            SensorManager.SENSOR_DELAY_NORMAL);
    }

    @Override
    public void onSensorChanged(SensorEvent event) {
        if (event.sensor.getType() == Sensor.TYPE_ACCELEROMETER) {
            // values[0]:x 轴, values[1] : y轴, values[2] : z 轴
            float[] values = event.values;
            if ((Math.abs(values[0]) > 15 || Math.abs(values[1]) > 15 || Math.abs(values[2]) > 15)) {

```

```

                tv_shake.setText(Utils.getNowTime()+" 恭喜您摇一摇啦");
                mVibrator.vibrate(500); //系统检测到摇一摇事件后, 震动手机提示用户
            }
        }
    }

    @Override
    public void onAccuracyChanged(Sensor sensor, int accuracy) {
        //当传感器精度改变时回调该方法, 一般无须处理
    }
}

```

device

22:50:11 恭喜您摇一摇啦

9-19

9-19

9.3.3

AppSensorManager.getRotationMatrix.getOrientation

```
public class DirectionActivity extends AppCompatActivity implements SensorEventListener {
    private TextView tv_direction;
    private CompassView cv_sourth;
    private SensorManager mSensroMgr;
    private float[] mAcceValues;
    private float[] mMagnValues;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_direction);
        tv_direction = (TextView) findViewById(R.id.tv_direction);
        cv_sourth = (CompassView) findViewById(R.id.cv_sourth);
        mSensroMgr = (SensorManager) getSystemService(Context.SENSOR_SERVICE);
    }

    @Override
    protected void onPause() {
        super.onPause();
        mSensroMgr.unregisterListener(this);
    }

    @Override
```

```

protected void onResume() {
    super.onResume();
    int suitable = 0;
    List<Sensor> sensorList = mSensroMgr.getSensorList(Sensor.TYPE_ALL);
    for (Sensor sensor : sensorList) {
        if (sensor.getType() == Sensor.TYPE_ACCELEROMETER) {
            suitable += 1;
        } else if (sensor.getType() == Sensor.TYPE_MAGNETIC_FIELD) {
            suitable += 10;
        }
    }
    if (suitable/10>0 && suitable%10>0) {
        mSensroMgr.registerListener(this,
            mSensroMgr.getDefaultSensor(Sensor.TYPE_ACCELEROMETER),
            SensorManager.SENSOR_DELAY_NORMAL);
        mSensroMgr.registerListener(this,
            mSensroMgr.getDefaultSensor(Sensor.TYPE_MAGNETIC_FIELD),
            SensorManager.SENSOR_DELAY_NORMAL);
    } else {
        cv_sourth.setVisibility(View.GONE);
        tv_direction.setText("当前设备不支持指南针，请检查是否存在加速度和磁场传感器");
    }
}

@Override
public void onSensorChanged(SensorEvent event) {
    if (event.sensor.getType() == Sensor.TYPE_ACCELEROMETER) {
        mAcceValues = event.values;
    } else if (event.sensor.getType() == Sensor.TYPE_MAGNETIC_FIELD) {
        mMagnValues = event.values;
    }
    if (mAcceValues!=null && mMagnValues!=null) {
        calculateOrientation();
    }
}

@Override
public void onAccuracyChanged(Sensor sensor, int accuracy) {
}

// 计算方向
private void calculateOrientation() {
    float[] values = new float[3];
    float[] R = new float[9];
}

```

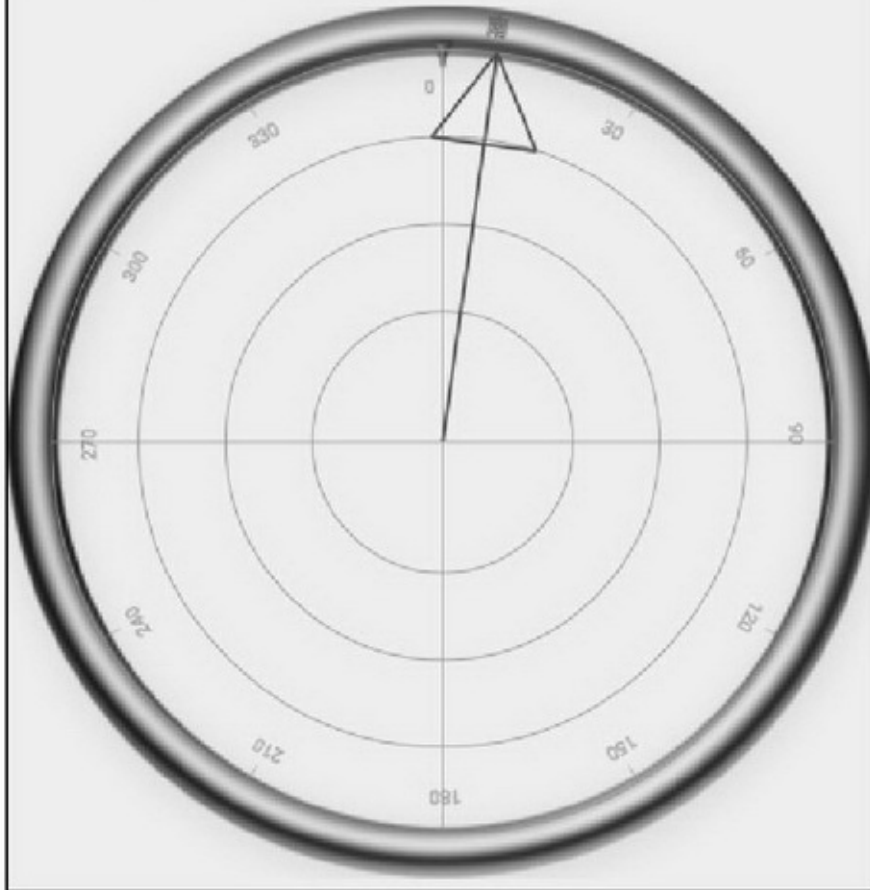


```
SensorManager.getRotationMatrix(R, null, mAcceValues, mMagnValues);
SensorManager.getOrientation(R, values);
values[0] = (float) Math.toDegrees(values[0]);
cv_sourth.setDirection((int) values[0]);
if (values[0] >= -10 && values[0] < 10) {
    tv_direction.setText("手机上部方向是正北");
} else if (values[0] >= 10 && values[0] < 80) {
    tv_direction.setText("手机上部方向是东北");
} else if (values[0] >= 80 && values[0] <= 100) {
    tv_direction.setText("手机上部方向是正东");
} else if (values[0] >= 100 && values[0] < 170) {
    tv_direction.setText("手机上部方向是东南");
} else if ((values[0] >= 170 && values[0] <= 180) || (values[0] >= -180 && values[0] < -170)) {
    tv_direction.setText("手机上部方向是正南");
} else if (values[0] >= -170 && values[0] < -100) {
    tv_direction.setText("手机上部方向是西南");
} else if (values[0] >= -100 && values[0] < -80) {
    tv_direction.setText("手机上部方向是正西");
} else if (values[0] >= -80 && values[0] < -10) {
    tv_direction.setText("手机上部方向是西北");
}
}
}
```

9-209-219-209-21

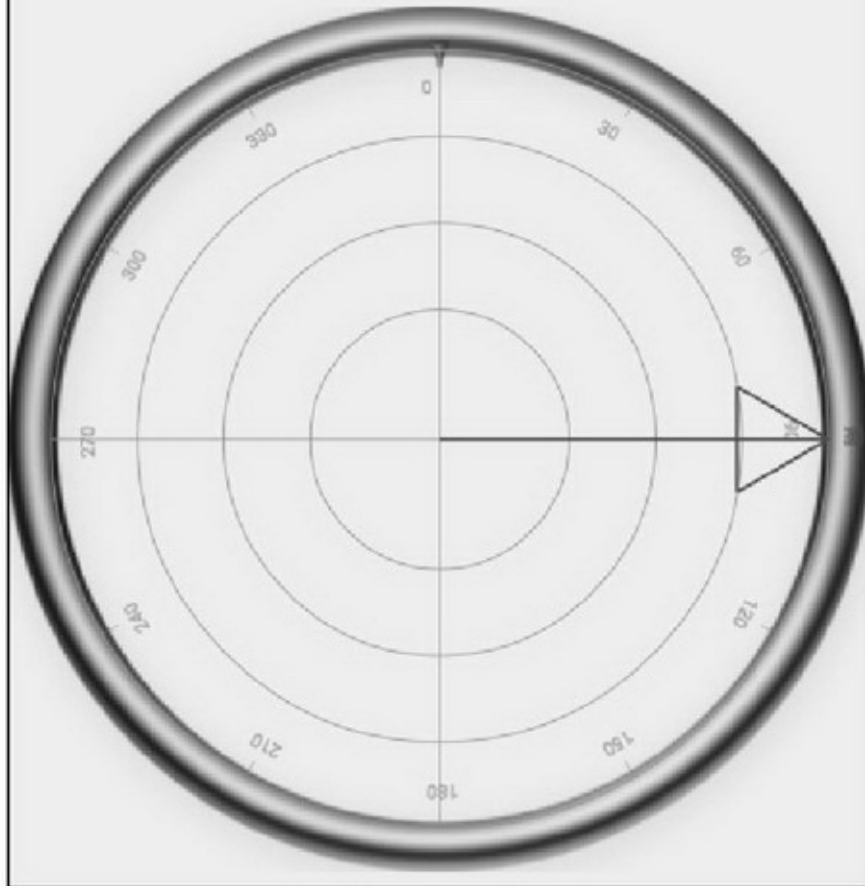
device

手机上部方向是正南



device

手机上部方向是正东



9-21

9.3.4

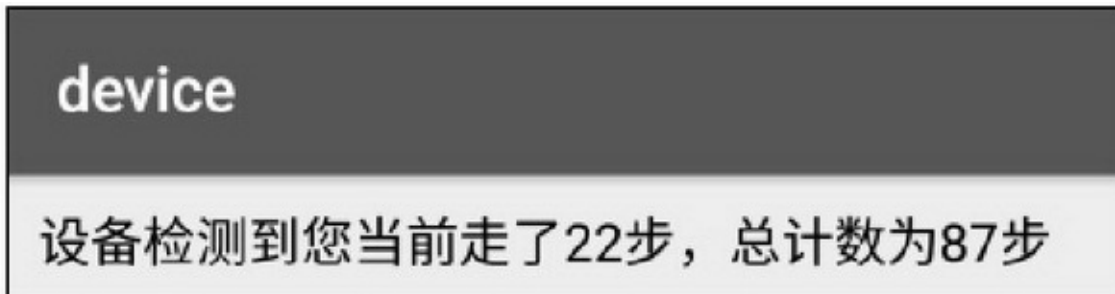
1. `Android(TYPE_STEP_DETECTOR)(TYPE_STEP_COUNTER)1`

```

@Override
public void onSensorChanged(SensorEvent event) {
    if (event.sensor.getType() == Sensor.TYPE_STEP_DETECTOR) {
        if (event.values[0] == 1.0f) {
            mStepDetector++;
        }
    } else if (event.sensor.getType() == Sensor.TYPE_STEP_COUNTER) {
        mStepCounter = (int) event.values[0];
    }
    String desc = String.format("设备检测到您当前走了%d步，总计数为%d步",
        mStepDetector, mStepCounter);
    tv_step.setText(desc);
}

```

9-22



9-22

2.

```

@Override
public void onSensorChanged(SensorEvent event) {
    if (event.sensor.getType() == Sensor.TYPE_LIGHT) {
        float light_strength = event.values[0];
        tv_light.setText(Utils.getNowTime() + " 当前光线强度为" + light_strength);
    }
}

```

9-23

device



亮度自动调节

22:58:55 当前光线强度为19.756496

9.4

CriteriaLocationManagerLocationListener

9.4.1

App

GPSPGPS

WIFISIMSIMSIM

WIFIWIFIWIFIMACIPWIFIWIFIWIFIWLAN

WIFIWIFIWIFI()WIFIMACWIFI

WIFI

GPSWLAN

```
// 获取 GPS 的开关状态
public static boolean getGpsStatus(Context ctx) {
    LocationManager lm = (LocationManager) ctx.getSystemService(Context.LOCATION_SERVICE);
    boolean gps_enabled = lm.isProviderEnabled(LocationManager.GPS_PROVIDER);
    return gps_enabled;
}

// 打开或关闭 GPS
@TargetApi(Build.VERSION_CODES.KITKAT)
public static void setGpsStatus(Context ctx, boolean enabled) {
    Intent gpsIntent = new Intent();
    gpsIntent.setClassName("com.android.settings",
        "com.android.settings.widget.SettingsAppWidgetProvider");
    gpsIntent.addCategory("android.intent.category.ALTERNATIVE");
    gpsIntent.setData(Uri.parse("custom:3"));
    try {
        PendingIntent.getBroadcast(ctx, 0, gpsIntent, 0).send();
    } catch (Exception e) {
        e.printStackTrace();
    }
}

// 获取定位的开关状态
public static boolean getLocationStatus(Context ctx) {
    LocationManager lm = (LocationManager) ctx.getSystemService(Context.LOCATION_SERVICE);
    boolean gps_enabled = lm.isProviderEnabled(LocationManager.GPS_PROVIDER);
    boolean network_enabled = lm.isProviderEnabled(LocationManager.NETWORK_PROVIDER);
    return gps_enabled || network_enabled;
}

// 获取 WIFI 的开关状态
public static boolean getWlanStatus(Context ctx) {
    WifiManager wm = (WifiManager) ctx.getSystemService(Context.WIFI_SERVICE);
    return wm.isWifiEnabled();
}
```

```

// 打开或关闭 WIFI
public static void setWlanStatus(Context ctx, boolean enabled) {
    WifiManager wm = (WifiManager) ctx.getSystemService(Context.WIFI_SERVICE);
    wm.setWifiEnabled(enabled);
}

// 获取数据连接的开关状态
public static boolean getMobileDataStatus(Context ctx) {
    ConnectivityManager cm = (ConnectivityManager) ctx.getSystemService(Context.
CONNECTIVITY_SERVICE);
    boolean isOpen = false;
    try {
        Method method = cm.getClass().getMethod("getMobileDataEnabled");
        isOpen = (Boolean) method.invoke(cm);
    } catch (Exception e) {
        e.printStackTrace();
    }
    return isOpen;
}

// 打开或关闭数据连接
public static void setMobileDataStatus(Context ctx, boolean enabled) {
    ConnectivityManager cm = (ConnectivityManager) ctx.getSystemService(Context.
CONNECTIVITY_SERVICE);
    try {
        Method method = cm.getClass().getMethod("setMobileDataEnabled", Boolean.TYPE);
        method.invoke(cm, enabled);
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

AndroidManifest.xml

```
<!-- -->
```

```
<uses-permission
```

```
android:name="android.permission.ACCESS_FINE_LOCATION" /> <uses-
permission
```

```
android:name="android.permission.ACCESS_COARSE_LOCATION" /> <!--
-->
```

```
<uses-permission
```

```
android:name="android.permission.ACCESS_NETWORK_STATE" /> <uses-
```



```
permission android:name="android.permission.ACCESS_WIFI_STATE" /> <!--  
-->
```

```
<uses-permission
```

```
android:name="android.permission.READ_PHONE_STATE" /> 9.4.2
```

```
CriteriaLocationManagerLocationListener3
```

```
1.Criteria
```

```
4
```

```
setAccuracyCriteria.ACCURACY_FINE
```

```
Criteria.ACCURACY_COARSE
```

```
setSpeedAccuracy9-8
```

9-8

Criteria 类的速度精确度	说明
ACCURACY_HIGH	精度高，误差小于 100 米
ACCURACY_MEDIUM	精度中等，误差在 100 米到 500 米之间
ACCURACY_LOW	精度低，误差大于 500 米

```
setAltitudeRequiredtruefalse
```

```
setBearingRequiredtruefalse
```

```
setCostAllowedtruefalse
```

```
setPowerRequirement3Criteria.POWER_LOW
```

```
Criteria.POWER_MEDIUMCriteria.POWER_HIGH
```

```
2.LocationManager
```

```
LOCATION_SERVICE7
```

```
getBestProviderCriteriatrue9-9
```

9-9

定位提供者的名称	说明	定位功能的开启状态
gps	卫星定位	开启 GPS 功能
network	网络定位	开启数据连接或 WLAN 功能
passive	无法定位	未开启定位相关功能

isEnabled
 getLastKnownLocation
 requestLocationUpdates
 removeUpdates
 addGpsStatusListenerGpsStatus.ListeneronGpsStatusChanged
 removeGpsStatusListener
 3.LocationListener
 4
 onLocationChanged
 onProviderDisabled
 onProviderEnabled
 onStatusChanged9-10

9-10

LocationProvider 类的状态类型	说明
OUT_OF_SERVICE	在服务范围外
TEMPORARILY_UNAVAILABLE	暂时不可用
AVAILABLE	可用状态

```

public class LocationActivity extends AppCompatActivity {
    private final static String TAG = "LocationActivity";
    private TextView tv_location;
    private String mLocation="";
    private LocationManager mLocationMgr;
    private Criteria mCriteria = new Criteria();
    private Handler mHandler = new Handler();
    private boolean bLocationEnable = false;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_location);
        initView();
        initLocation();
        mHandler.postDelayed(mRefresh, 100);
    }
}
  
```

```

}

private void initView() {
    tv_location = (TextView) findViewById(R.id.tv_location);
    mLocationMgr = (LocationManager) getSystemService(Context.LOCATION_SERVICE);
    mCriteria.setAccuracy(Criteria.ACCURACY_FINE); // 设置定位精确度
    mCriteria.setAltitudeRequired(true); // 设置是否需要海拔信息 Altitude
    mCriteria.setBearingRequired(true); // 设置是否需要方位信息 Bearing
    mCriteria.setCostAllowed(true); // 设置是否允许运营商收费
    mCriteria.setPowerRequirement(Criteria.POWER_LOW); // 设置对电源的需求
}

private void initLocation() {
    String bestProvider = mLocationMgr.getBestProvider(mCriteria, true);
    if (bestProvider == null) {
        bestProvider = LocationManager.NETWORK_PROVIDER;
    }
    if (mLocationMgr.isProviderEnabled(bestProvider)) {
        tv_location.setText("正在获取"+bestProvider+"定位对象");
        mLocation = String.format("定位类型=%s", bestProvider);
        beginLocation(bestProvider);
        bLocationEnable = true;
    } else {
        tv_location.setText("\n"+bestProvider+"定位不可用");
        bLocationEnable = false;
    }
}

private void setLocationText(Location location) {
    if (location != null) {
        String desc = String.format("%s\n 定位对象信息如下 : " +
            "\n\t其中时间 : %s" + "\n\t其中经度 : %f, 纬度 : %f" +
            "\n\t其中高度 : %d 米, 精度 : %d 米",
            mLocation, Utils.getNowDateTimeFormat(),
            location.getLongitude(), location.getLatitude(),
            Math.round(location.getAltitude()), Math.round(location.getAccuracy()));
        Log.d(TAG, desc);
        tv_location.setText(desc);
    } else {
        tv_location.setText(mLocation+"\n 暂未获取定位对象");
    }
}
}

```

```

private void beginLocation(String method) {
    mLocationMgr.requestLocationUpdates(method, 300, 0, mLocationListener);
    Location location = mLocationMgr.getLastKnownLocation(method);
    setLocationText(location);
}

// 位置监听器
private LocationListener mLocationListener = new LocationListener() {
    @Override
    public void onLocationChanged(Location location) {
        setLocationText(location);
    }

    @Override
    public void onProviderDisabled(String arg0) {
    }

    @Override
    public void onProviderEnabled(String arg0) {
    }

    @Override
    public void onStatusChanged(String arg0, int arg1, Bundle arg2) {
    }
};

private Runnable mRefresh = new Runnable() {
    @Override
    public void run() {
        if (bLocationEnable == false) {
            initLocation();
            mHandler.postDelayed(this, 1000);
        }
    }
};

@Override
protected void onDestroy() {
    if (mLocationMgr != null) {
        mLocationMgr.removeUpdates(mLocationListener);
    }
    super.onDestroy();
}
}

```

9-2411926776

device

定位类型=gps

定位对象信息如下：

其中时间：2016-11-05 23:10:58

其中经度：119.332101，纬度：26.146713

其中高度：77米，精度：6米

9.5

AppAppApp

9.5.1

9-25

2016

——“”39-26



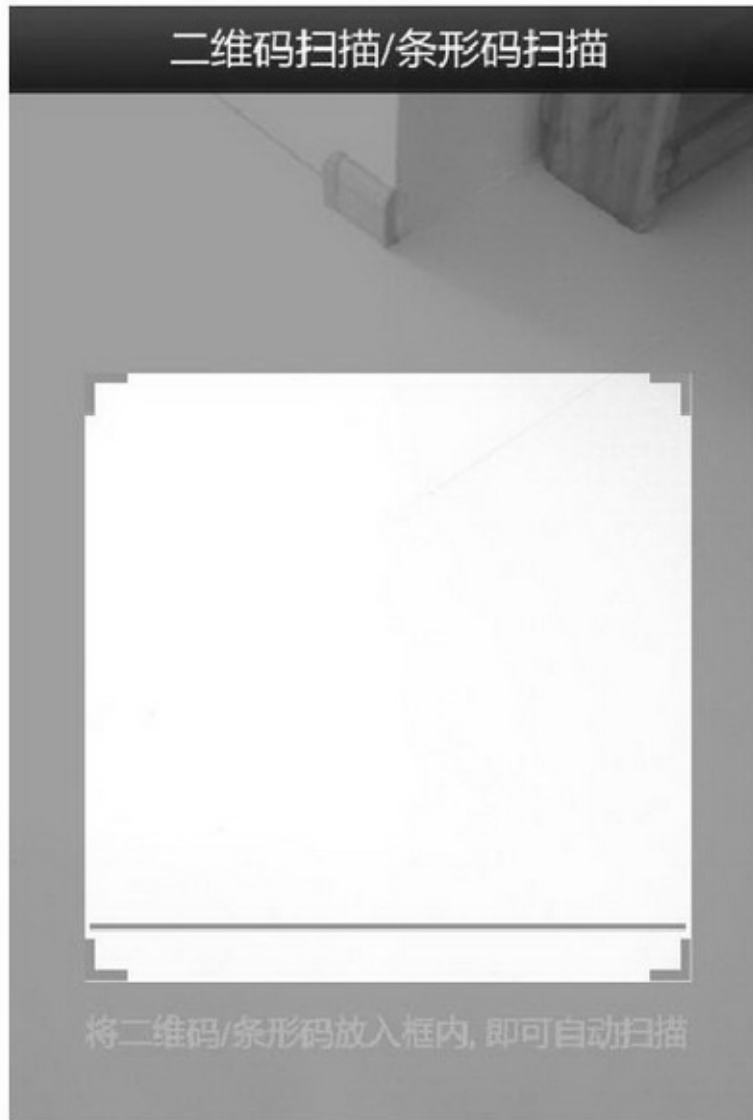


发现

扫一扫（扫描二维码）

摇一摇（博饼中大奖）

咻一咻（卫星浑天仪）



9-27()

3

1.0

“”Androidzxingzxing9-27

2.0

“”____“”300“”

9-28

device



摇一摇，博饼中大奖



9-29()

3.0

(09-29

34

(1)CameraCamera

(2)SensorManager

(3)LocationManager

(4)MediaPlayer""""MediaPlayer

9.5.2

4GPS3

(1)GPSGPSGlobal Positioning System()1964199324GPS

(2)(GLONASS)Global Navigation Satellite System2007201124

(3)(BeiDou Navigation Satellite SystemBDS)GPS32007
201216202035

(4)(Galileo Satellite Navigation System)20134

GPS

LocationManageraddGpsStatusListenerGpsStatus.Listener
onGpsStatusChanged9-11

9-11GPS

GpsStatus 类的事件类型	说明
GPS_EVENT_STARTED	GPS 功能开启
GPS_EVENT_STOPPED	GPS 功能停止
GPS_EVENT_FIRST_FIX	首次定位
GPS_EVENT_SATELLITE_STATUS	周期地报告卫星状态

LocationManagergetGpsStatusGpsStatusGpsStatus
getSatellitesGpsSatelliteGpsSatellite

getPrn

getAzimuth

getElevation

getSnr

hasAlmanac

hasEphemeris

usedInFixGPS

3000

```

private GpsStatus.Listener mStatusListener = new GpsStatus.Listener() {
    @Override
    public void onGpsStatusChanged(int event) {
        GpsStatus gpsStatus = mLocationMgr.getGpsStatus(null);
        switch (event) {
            case GpsStatus.GPS_EVENT_FIRST_FIX:
                break;
            case GpsStatus.GPS_EVENT_SATELLITE_STATUS: // 周期的报告卫星状态
                // 得到所有卫星信息，包括高度角、方位角、信噪比和伪随机机号（卫星编号）
                Iterable<GpsSatellite> satellites = gpsStatus.getSatellites();
                for (GpsSatellite satellite : satellites) {
                    Satellite item = new Satellite();
                    item.seq = satellite.getPrn();
                    item.signal = Math.round(satellite.getSnr());
                    item.elevation = Math.round(satellite.getElevation());
                    item.azimuth = Math.round(satellite.getAzimuth());
                    item.time = Utils.getNowDateTime();
                    if (item.seq <= 64 || (item.seq >= 120 && item.seq <= 138)) {
                        item.nation = "美国";
                        item.name = "GPS";
                    } else if (item.seq >= 201 && item.seq <= 237) {
                        item.nation = "中国";
                        item.name = "北斗";
                    } else if (item.seq >= 65 && item.seq <= 89) {
                        item.nation = "俄罗斯";
                        item.name = "格洛纳斯";
                    } else {
                        item.nation = "其他";
                        item.name = "未知";
                    }
                }
                mapSatellite.put(item.seq, item);
            }
}

```

```

        }
        cv_satellite.setSatelliteMap(mapSatellite);
    case GpsStatus.GPS_EVENT_STARTED:
        break;
    case GpsStatus.GPS_EVENT_STOPPED:
        break;
    default:
        break;
    }
}
};

```

9.5.3

5

(1)zxinglibsJARzxing3.2.1.jarJavacom.app.zxing

(2)AndroidManifest.xml

```
<uses-permission android:name="android.permission.CAMERA" /> <uses-  
permission android:name="android.permission.ACCESS_FINE_LOCATION"  
/> <uses-permission  
android:name="android.permission.ACCESS_COARSE_LOCATION" /> (3)  
res/raw""res/valueszxingids.xml
```

(4)BettingViewCompassView

(5)AppGPSGPS

8App

9-309-31HTTP

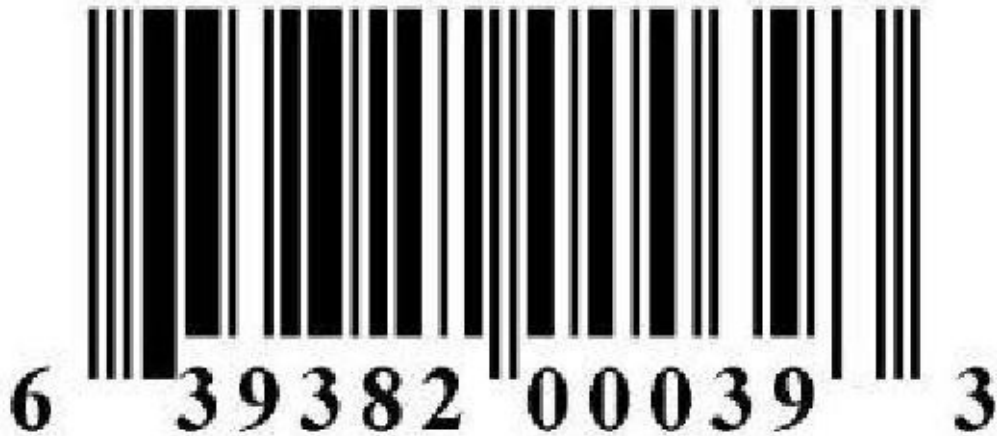


device

扫码结果为: <http://weixin.qq.com/r/U0MMFPvEqabWrb939xZB>

9-31

9-329-33

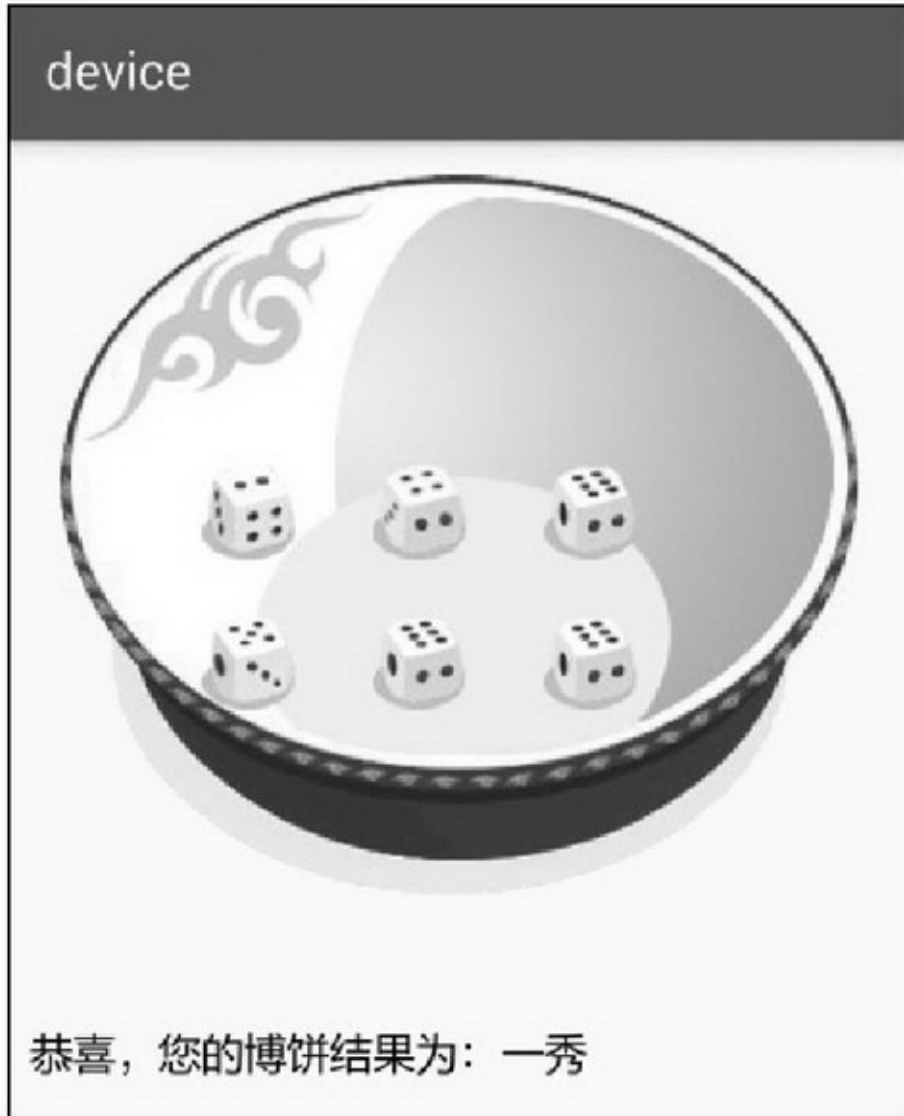


9-32

device

扫码结果为: 639382000393

9-349-359-34()9-35(4)



device



恭喜，您的博饼结果为：状元(四点红)

9-35

4
6
6
5
4
5(5)

06
04(4)
3
0
0

9-36GPS9-37GPS



9-36GPS



9-373

GPS105

9-3711GPS612GPS

CompassView()

```

public class CompassView extends View {
    private final static String TAG = "CompassView";
    private Context mContext;
    private int mWidth;
    private Paint mPaintLine;
    private Paint mPaintText;
    private Paint mPaintAngle;
    private Bitmap mCompassBg;
    private Rect mRectSrc;
    private Rect mRectDest;
    private RectF mRectAngle;
    private RectF mRectSourth;
    private Bitmap mSatelliteChina;

```

```

    private Bitmap mSatelliteAmerica;
    private Bitmap mSatelliteRussia;
    private Map<Integer, Satellite> mapSatellite = new HashMap<Integer, Satellite>();
    private int mScaleLength = 25;
    private float mBorder = 0.9f;
    private int mDirection = -1024;
    private Paint mPaintSourth;

    public CompassView(Context context) {
        this(context, null);
    }

    public CompassView(Context context, AttributeSet attr) {
        super(context, attr);
        mContext = context;
        mPaintLine = new Paint();
        mPaintLine.setAntiAlias(true);
        mPaintLine.setColor(Color.GREEN);
        mPaintLine.setStrokeWidth(2);
        mPaintLine.setStyle(Style.STROKE);
        mPaintText = new Paint();
        mPaintText.setAntiAlias(true);
        mPaintText.setColor(Color.RED);
        mPaintText.setStrokeWidth(1);
        mPaintText.setStyle(Style.STROKE);
        mPaintText.setTextSize(28);
        mPaintAngle = new Paint();
        mPaintAngle.setAntiAlias(true);
        mPaintAngle.setColor(Color.GRAY);
        mPaintAngle.setStrokeWidth(1);
        mPaintAngle.setStyle(Style.STROKE);
        mPaintAngle.setTextSize(23);
        mPaintSourth = new Paint();
        mPaintSourth.setAntiAlias(true);
        mPaintSourth.setColor(Color.RED);
        mPaintSourth.setStrokeWidth(4);
        mPaintSourth.setStyle(Style.STROKE);
        mCompassBg = BitmapFactory.decodeResource(getResources(), R.drawable.compass_bg);
        mRectSrc = new Rect(0, 0, mCompassBg.getWidth(), mCompassBg.getHeight());
        mSatelliteChina = BitmapFactory.decodeResource(getResources(), R.drawable.satellite_china);
        mSatelliteAmerica = BitmapFactory.decodeResource(getResources(), R.drawable.satellite_america);
        mSatelliteRussia = BitmapFactory.decodeResource(getResources(), R.drawable.satellite_russia);
    }

    @Override

```

```

protected void onMeasure(int widthMeasureSpec, int heightMeasureSpec) {
    int width = View.MeasureSpec.getSize(widthMeasureSpec);
    int height = View.MeasureSpec.getSize(heightMeasureSpec);
    mWidth = getMeasuredWidth();
    if (width < height) {
        super.onMeasure(widthMeasureSpec, widthMeasureSpec);
    } else {
        super.onMeasure(heightMeasureSpec, heightMeasureSpec);
    }
    mRectDest = new Rect(0, 0, mWidth, mWidth);
    mRectAngle = new RectF(mWidth/10, mWidth/10, mWidth*9/10, mWidth*9/10);
    mRectSouth = new RectF(mWidth*0.3f/10, mWidth*0.3f/10, mWidth*9.7f/10, mWidth*9.7f/10);
    Log.d(TAG, "mWidth="+mWidth);
}

@Override
protected void dispatchDraw(Canvas canvas) {
    super.dispatchDraw(canvas);
    int radius = mWidth/2;
    int margin = radius/10;
    canvas.drawBitmap(mCompassBg, mRectSrc, mRectDest, new Paint());
    canvas.drawCircle(radius, radius, radius*3/10, mPaintLine);
    canvas.drawCircle(radius, radius, radius*5/10, mPaintLine);
    canvas.drawCircle(radius, radius, radius*7/10, mPaintLine);
    canvas.drawCircle(radius, radius, radius*9/10, mPaintLine);
    canvas.drawLine(radius, margin, radius, mWidth-margin, mPaintLine);
    canvas.drawLine(margin, radius, mWidth-margin, radius, mPaintLine);
    //画罗盘的刻度
    for (int i=0; i<360; i+=30) {
        Path path = new Path();
        path.addArc(mRectAngle, i-3, i+3);
        int angle = (i+90)%360;
        canvas.drawTextOnPath(""+angle, path, 0, 0, mPaintAngle);
        canvas.drawLine(getXpos(radius, angle, radius*mBorder),
            getYpos(radius, angle, radius*mBorder),
            getXpos(radius, angle, (radius-mScaleLength)*mBorder),
            getYpos(radius, angle, (radius-mScaleLength)*mBorder), mPaintAngle);
    }
    //画卫星分布图
    for (Map.Entry<Integer, Satellite> item_map : mapSatellite.entrySet()) {
        Satellite item = item_map.getValue();
        Bitmap bitmap;
        if (item.nation.equals("中国")) {
            bitmap = mSatelliteChina;
        }
    }
}

```

```

    } else if (item.nation.equals("美国")) {
        bitmap = mSatelliteAmerica;
    } else if (item.nation.equals("俄罗斯")) {
        bitmap = mSatelliteRussia;
    } else {
        continue;
    }
    float left = getXpos(radius, item.azimuth, radius*mBorder*getCos(item.elevation));
    float top = getYpos(radius, item.azimuth, radius*mBorder*getCos(item.elevation));
    canvas.drawBitmap(bitmap, left-bitmap.getWidth()/2, top-bitmap.getHeight()/2, new Paint());
}
//画指南针
if (mDirection > -1024) {
    int angle = (-mDirection+450)%360;
    canvas.drawLine(getXpos(radius, angle, radius*mBorder),
        getYpos(radius, angle, radius*mBorder),
        getXpos(radius, angle, 0), getYpos(radius, angle, 0),mPaintSouth);
    canvas.drawLine(getXpos(radius, angle, radius*mBorder),
        getYpos(radius, angle, radius*mBorder),
        getXpos(radius, angle-10, radius*7/10), getYpos(radius, angle-10, radius*7/10),
        mPaintSouth);
    canvas.drawLine(getXpos(radius, angle, radius*mBorder),
        getYpos(radius, angle, radius*mBorder),
        getXpos(radius, angle+10, radius*7/10), getYpos(radius, angle+10, radius*7/10),
        mPaintSouth);
    canvas.drawLine(getXpos(radius, angle-10, radius*7/10),
        getYpos(radius, angle-10, radius*7/10),
        getXpos(radius, angle+10, radius*7/10), getYpos(radius, angle+10, radius*7/10),
        mPaintSouth);
    Path path = new Path();
    path.addArc(mRectSouth, angle-2, angle+2);
    canvas.drawTextOnPath("南", path, 0, 0, mPaintText);
} else {
    canvas.drawText("北", radius-15, margin-15, mPaintText);
}
}

private float getXpos(int radius, int angle, double length) {
    return (float) (radius + getCos(angle) * length);
}

private float getYpos(int radius, int angle, double length) {
    return (float) (radius + getSin(angle) * length);
}
}

```

```
private double getSin(int angle) {
    return Math.sin(Math.PI*angle/180.0);
}

private double getCos(int angle) {
    return Math.cos(Math.PI*angle/180.0);
}

public void setSatelliteMap(Map<Integer, Satellite> map) {
    mapSatellite = map;
    invalidate();
}

public void setDirection(int direction) {
    mDirection = direction;
    invalidate();
}
}
```

9.6

000(
)"App0003

5

(1)0

(2)0

(3)

(4)0

(5)0

第10章 网络通信



AppHTTPSocket“QQ”

10.1

AppMessageHandlerAsyncTaskIntentService

10.1.1 Message AppAppAndroid9Android

SurfaceViewTextureView

AndroidMessageHandlerMessage

4

1.Handler

Handler2Handler6Handler+RunnableThreadRunnable

HandlerRunnablepostThreadHandlerpostThreadThreadstart

2.Message

MessageIntentBundleobtainHandlerobtainMessage

Message

what

arg1

arg2

objObject

replyToMessenger

3.HandlerMessage

Handlersend

obtainMessage

sendMessage

sendMessageDelayed

sendMessageAtTime

sendEmptyMessage

sendEmptyMessageDelayed

sendEmptyMessageAtTime

removeMessages

hasMessages

4.Handler

HandlerhandleMessageMessagehandleMessage(UI)

HandlerMessage

```
public class MessageActivity extends AppCompatActivity implements OnClickListener {
    private TextView tv_message;
    private boolean bPlay = false;
    private int BEGIN = 0, SCROLL = 1, END = 2;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_message);
        tv_message = (TextView) findViewById(R.id.tv_message);
        tv_message.setGravity(Gravity.LEFT|Gravity.BOTTOM);
        tv_message.setLines(8);
        tv_message.setMaxLines(8);
        tv_message.setMovementMethod(new ScrollingMovementMethod());
        findViewById(R.id.btn_start_message).setOnClickListener(this);
        findViewById(R.id.btn_stop_message).setOnClickListener(this);
    }

    @Override
    public void onClick(View v) {
        if (v.getId() == R.id.btn_start_message) {
            if (bPlay != true) {
```

```

        bPlay = true;
        new PlayThread().start();
    }
} else if (v.getId() == R.id.btn_stop_message) {
    bPlay = false;
}
}

private String[] mNewsArray = { "中国航天员在天宫二号泡茶，美煞歪果仁",
    "美国大选顺利闭幕，特朗普高票当选", "越南撤销日本获得订单的核电站计划",
    "上海建成卓越全球城市，新市镇轨交全覆盖", "土耳其老人怀抱受伤山羊错跑入医院急诊室" };
private class PlayThread extends Thread {
    @Override
    public void run() {
        mHandler.sendEmptyMessage(BEGIN);
        while (bPlay == true) {
            try {
                sleep(2000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
            Message message = Message.obtain();
            message.what = SCROLL;
            message.obj = mNewsArray[(int) (Math.random()*30%5)];
            mHandler.sendMessage(message);
        }
        bPlay = true;
        try {
            sleep(2000);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
        mHandler.sendEmptyMessage(END);
        bPlay = false;
    }
}

private Handler mHandler = new Handler() {
    @Override
    public void handleMessage(Message msg) {
        String desc = tv_message.getText().toString();
        if (msg.what == BEGIN) {
            desc = String.format("%s\n%s %s", desc, DateUtil.getNowTime(), "下面开始播放新闻");

```

```

        } else if (msg.what == SCROLL) {
            desc = String.format("%s\n%s %s", desc, DateUtil.getNowTime(), (String) msg.obj);
        } else if (msg.what == END) {
            desc = String.format("%s\n%s %s", desc, DateUtil.getNowTime(), "新闻播放结束，谢
谢观看");
        }
        tv_message.setText(desc);
    };
};
}

```

10-110-210-110-2END“”

network

开始播放新闻 停止播放新闻

15:27:12 下面开始播放新闻
15:27:14 越南撤销日本获得订单的核电站计划
15:27:16 美国大选顺利闭幕，特朗普高票当选
15:27:18 土耳其老人怀抱受伤山羊措跑入医院急诊室

10-1

network

开始播放新闻 停止播放新闻

15:27:50 土耳其老人怀抱受伤山羊措跑入医院急诊室
15:27:52 越南撤销日本获得订单的核电站计划
15:27:54 上海建成卓越全球城市，新市镇轨交全覆盖
15:27:56 上海建成卓越全球城市，新市镇轨交全覆盖
15:27:58 中国航天员在天宫二号泡茶，羡慕歪果仁
15:28:00 土耳其老人怀抱受伤山羊措跑入医院急诊室
15:28:02 越南撤销日本获得订单的核电站计划
15:28:04 新闻播放结束，谢谢观看

10-2

10.1.2ProgressDialog “”

ProgressDialog

AlertDialogProgressBarAlertDialogProgressBarAPI

setTitle

setMessage

setIcon

setProgress

setSecondaryProgress

setMax

setProgressStyleProgressDialog.STYLE_SPINNER()

ProgressDialog.STYLE_HORIZONTAL

showshow

isShowing

dismiss

show

ProgressDialog.show(this, "", "");

```

private String[] descArray={"圆圈进度", "水平进度条"};
private int[] styleArray={ProgressDialog.STYLE_SPINNER, ProgressDialog.STYLE_HORIZONTAL};
private class StyleSelectedListener implements OnItemSelectedListener {
    public void onItemSelected(AdapterView<?> arg0, View arg1, int arg2, long arg3) {
        if (mProgressDialog == null || mProgressDialog.isShowing() != true) {
            mStyleDesc = descArray[arg2];
            int style = styleArray[arg2];
            if (style == ProgressDialog.STYLE_SPINNER) {
                mProgressDialog = ProgressDialog.show(ProgressDialogActivity.this,
                    "请稍候", "正在努力加载页面");
                mHandler.postDelayed(mCloseDialog, 1500);
            } else {
                mProgressDialog = new ProgressDialog(ProgressDialogActivity.this);
                mProgressDialog.setTitle("请稍候");
                mProgressDialog.setMessage("正在努力加载页面");
                mProgressDialog.setMax(100);
                mProgressDialog.setProgressStyle(style);
                mProgressDialog.show();
                new RefreshThread().start();
            }
        }
    }
    public void onNothingSelected(AdapterView<?> arg0) {
    }
}

private Runnable mCloseDialog = new Runnable() {
    @Override
    public void run() {
        if (mProgressDialog.isShowing() == true) {
            mProgressDialog.dismiss();
            tv_result.setText(DateUtil.getNowTime()+" "+mStyleDesc+"加载完成");
        }
    }
};

```

```

private class RefreshThread extends Thread {
    @Override
    public void run() {
        for (int i=0; i<10; i++) {
            Message message = Message.obtain();
            message.what = 0;
            message.arg1 = i*10;
            mHandler.sendMessage(message);
            try {
                sleep(500);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
        mHandler.sendEmptyMessage(1);
    }
}

private Handler mHandler = new Handler() {
    @Override
    public void handleMessage(Message msg) {
        if (msg.what == 0) {
            mProgressDialog.setProgress(msg.arg1);
        } else if (msg.what == 1) {
            post(mCloseDialog);
        }
    }
};
};

```

10-310-410-31.510-40.5105



10-3



10-4

Android

1.

progressDrawable6“6.4.2ProcessBar”ProgressBaronDraw
canvasdrawText

10-510-610-540%10-680%



10-540%

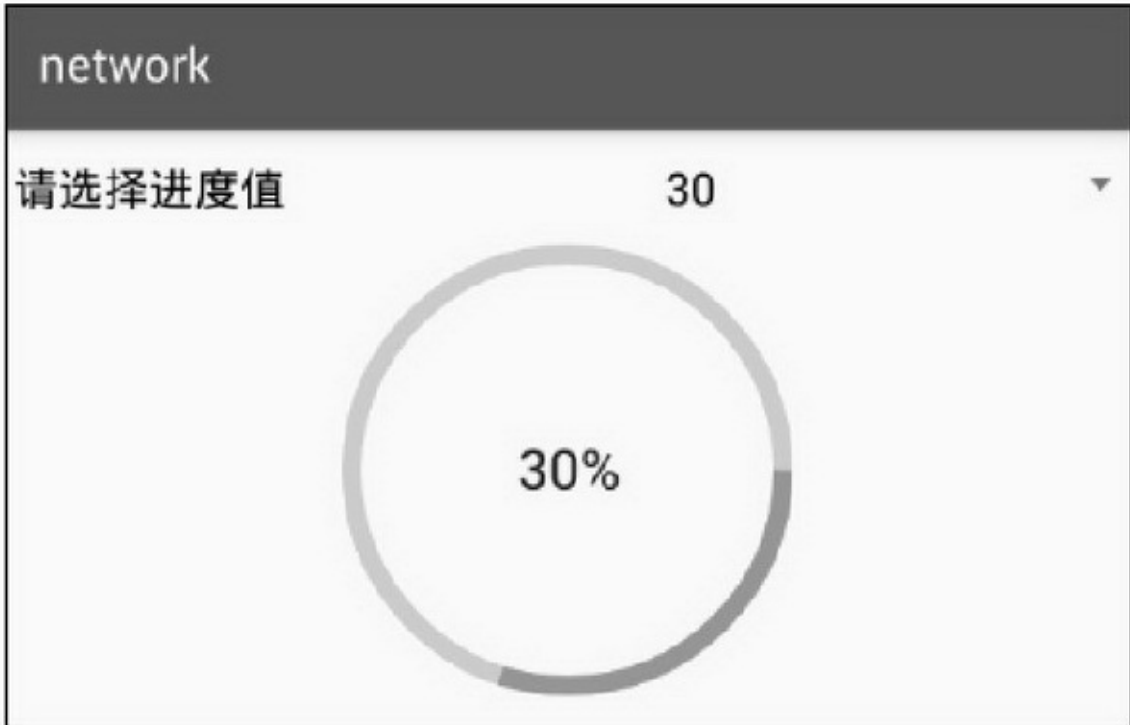


10-680%

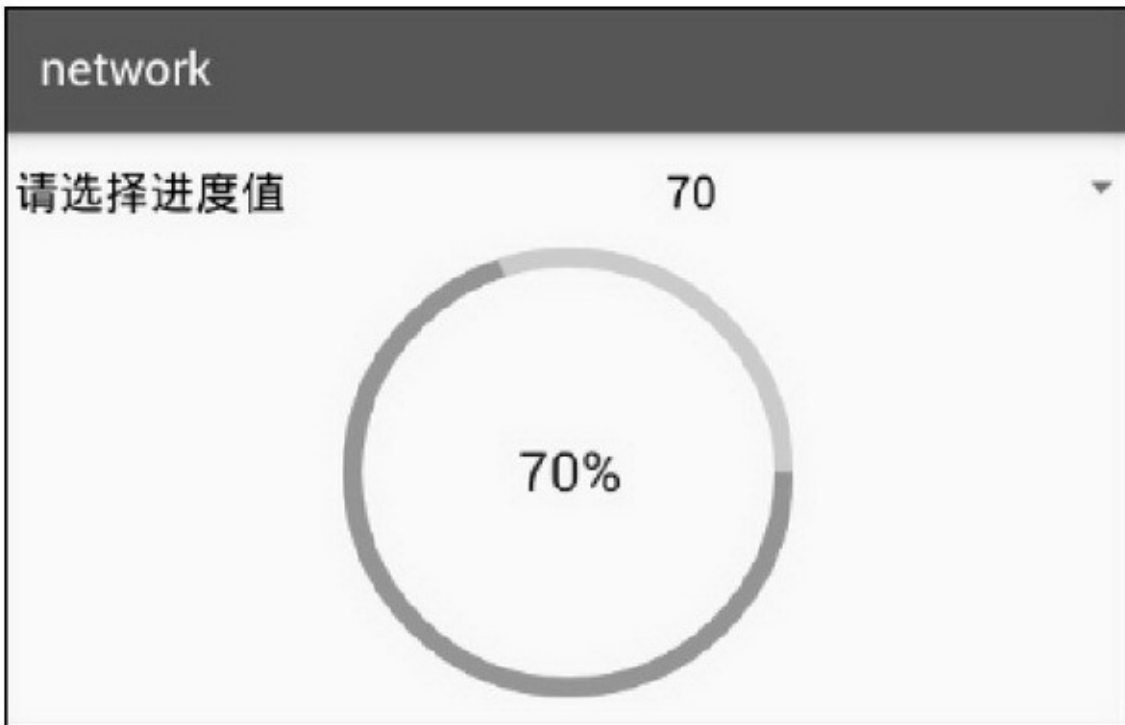
2.

AppProgressBarProgressBar6("6.2.3")

10-710-810-730%10-870%



10-730%



10-870%

10.1.3 AsyncTask Thread+HandlerAndroid

AsyncTaskThread+HandlerAsyncTaskHTTP

AsyncTask(AsyncTask<Params, Progress, Result>)

ParamsHTTPURLString

ProgressInteger

ResultHTTPString

onPreExecutedoInBackground

doInBackgroundexecuteonPostExecute

onProgressUpdatedoInBackgroundpublishProgress

onPostExecuteoInBackgrounddoInBackground

onCancelled :cancelAsyncTask

execute

➤ AsyncTask.THREAD_POOL_EXECUTOR()

➤ AsyncTask.SERIAL_EXECUTOR()execute

SERIAL_EXECUTOR

executeOnExecutorAsyncTask

publishProgressdoInBackgroundonProgressUpdate

get

canceldoInBackgrounddoInBackgroundisCancelled

isCancelledtruefalse

getStatus10-1

10-1

AsyncTask.Status 类的任务状态	说明	所处时刻
PENDING	还未执行	onPreExecute 处理之前（正在等待）
RUNNING	正在执行	onPreExecute、doInBackground、onPostExecute 运行期间
FINISHED	执行完毕	onPostExecute 处理结束

```
public class ProgressAsyncTask extends AsyncTask<String, Integer, String> {
    private String mBook;
    public ProgressAsyncTask(String title) {
        super();
        mBook = title;
    }

    @Override
    protected String doInBackground(String... params) {
        int ratio = 0;
        for (; ratio <= 100; ratio += 5) {
            try {
                Thread.sleep(200); // 睡眠 200 毫秒模拟网络通信处理
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
            publishProgress(ratio);
        }
        return params[0];
    }

    @Override
    protected void onPreExecute() {
        mListener.onBegin(mBook);
    }

    @Override
    protected void onProgressUpdate(Integer... values) {
        mListener.onUpdate(mBook, values[0], 0);
    }

    @Override
    protected void onPostExecute(String result) {
        mListener.onFinish(result);
    }
}
```

```

@Override
protected void onCancelled(String result) {
    mListener.onCancel(result);
}

private OnProgressListener mListener;
public void setOnProgressListener(OnProgressListener listener) {
    mListener = listener;
}

public static interface OnProgressListener {
    public abstract void onFinish(String result);
    public abstract void onCancel(String result);
    public abstract void onUpdate(String request, int progress, int sub_progress);
    public abstract void onBegin(String request);
}
}

```

Activity

```

public class AsyncTaskActivity extends AppCompatActivity implements OnProgressListener {
    private TextView tv_async;
    private ProgressBar pb_async;
    private ProgressDialog mDialog;
    public int mShowMode;
    public int BAR_HORIZONTAL = 1, DIALOG_CIRCLE = 2, DIALOG_HORIZONTAL = 3;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_async_task);
        tv_async = (TextView) findViewById(R.id.tv_async);
        pb_async = (ProgressBar) findViewById(R.id.pb_async);
        ArrayAdapter<String> styleAdapter = new ArrayAdapter<String>(this,
            R.layout.item_select, bookArray);
        Spinner sp_style = (Spinner) findViewById(R.id.sp_style);
        sp_style.setPrompt("请选择要加载的小说");
        sp_style.setAdapter(styleAdapter);
        sp_style.setOnItemSelectedListener(new StyleSelectedListener());
        sp_style.setSelection(0);
    }

    private String[] bookArray={"三国演义", "西游记", "红楼梦"};
    private int[] styleArray={BAR_HORIZONTAL, DIALOG_CIRCLE, DIALOG_HORIZONTAL};
}

```

```
class StyleSelectedListener implements OnItemSelectedListener {
    public void onItemSelected(AdapterView<?> arg0, View arg1, int arg2, long arg3) {
        startTask(styleArray[arg2], bookArray[arg2]);
    }

    public void onNothingSelected(AdapterView<?> arg0) {
    }
}

private void startTask(int mode, String msg) {
    mShowMode = mode;
    ProgressAsyncTask asyncTask = new ProgressAsyncTask(msg);
    asyncTask.setOnProgressListener(this);
    asyncTask.execute(msg);
}

private void closeDialog() {
    if (mDialog != null && mDialog.isShowing() == true) {
        mDialog.dismiss();
    }
}

@Override
public void onFinish(String result) {
    String desc = String.format("您要阅读的《%s》已经加载完毕", result);
    tv_async.setText(desc);
    closeDialog();
}

@Override
public void onCancel(String result) {
    String desc = String.format("您要阅读的《%s》已经取消加载", result);
    tv_async.setText(desc);
    closeDialog();
}

@Override
public void onUpdate(String request, int progress, int sub_progress) {
    String desc = String.format("%s 当前加载进度为%d%%", request, progress);
    tv_async.setText(desc);
    if (mShowMode == BAR_HORIZONTAL) {
        pb_async.setProgress(progress);
        pb_async.setSecondaryProgress(sub_progress);
    } else if (mShowMode == DIALOG_HORIZONTAL) {
        mDialog.setProgress(progress);
        mDialog.setSecondaryProgress(sub_progress);
    }
}
```

```

    }
}

@Override
public void onBegin(String request) {
    tv_async.setText(request+"开始加载");
    if(mDialog == null || mDialog.isShowing() != true) {
        if(mShowMode == DIALOG_CIRCLE) {
            mDialog = ProgressDialog.show(this, "稍等", request+"页面加载中……");
        } else if (mShowMode == DIALOG_HORIZONTAL) {
            mDialog = new ProgressDialog(this);
            mDialog.setTitle("稍等");
            mDialog.setMessage(request+"页面加载中……");
            mDialog.setIcon(R.drawable.ic_search);
            mDialog.setProgressStyle(ProgressDialog.STYLE_HORIZONTAL);
            mDialog.show();
        }
    }
}
}
}
}

```

10-910-1010-1110-910-1010-11





10-10



10-11

AsyncTaskAsyncTask
(1)AsyncTaskSERIAL_EXECUTOR
(2)AndroidexecuteOnExecutorAsyncTask
THREAD_POOL_EXECUTORCPU1(AsyncTask

“MAXIMUM_POOL_SIZE = CPU_COUNT * 2 + 1”)CPUAsyncTask
2*2+1=55CPUAsyncTask4*2+1=9CPUApp

10.1.4IntentService ServiceActivity

Thread+HandlerAndroidIntentService

IntentServiceIntentServiceHandler

HandlerhandleMessageHandlerHandlerhandleMessage

IntentService

步骤 01

Handlerthread.getLooper

```
public void onCreate() {  
    super.onCreate();  
    HandlerThread thread = new HandlerThread("IntentService[" + mName + "]");  
    thread.start();  
    mServiceLooper = thread.getLooper();  
    mServiceHandler = new ServiceHandler(mServiceLooper);  
}
```

步骤 02

Handler

```
public void onStart(Intent intent, int startId) {  
    Message msg = mServiceHandler.obtainMessage();  
    msg.arg1 = startId;  
    msg.obj = intent;  
    mServiceHandler.sendMessage(msg);  
}
```

步骤 03

HandlerhandleMessageonHandleIntentstopSelf

```
private final class ServiceHandler extends Handler {
    public ServiceHandler(Looper looper) {
        super(looper);
    }

    @Override
    public void handleMessage(Message msg) {
        onHandleIntent((Intent)msg.obj);
        stopSelf(msg.arg1);
    }
}
```

IntentService4

(1)

(2)onStartCommandonStartCommand

(3)onHandleIntentonStartCommandonHandleIntent

onStartCommand

(4)IntentServiceonStartonBind

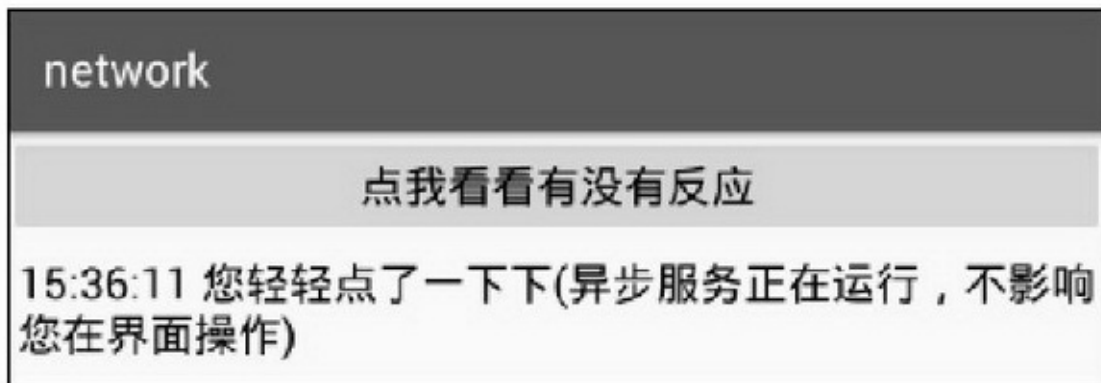
```

public class AsyncService extends IntentService {
    private static final String TAG = "AsyncService";
    public AsyncService() {
        super("com.example.network.service.AsyncService");
    }

    @Override
    public int onStartCommand(Intent intent, int flags, int startid) {
        Log.i(TAG, "onStartCommand");
        //试试在 onStartCommand 里调用 Thread.sleep 方法，页面按钮是不是无法点击了
        return super.onStartCommand(intent, flags, startid);
    }

    @Override
    protected void onHandleIntent(Intent intent) {
        Log.d(TAG, "begin onHandleIntent");
        //在 onHandleIntent 中执行耗时任务，不会影响页面的处理
        try {
            Thread.sleep(30*1000);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
        Log.d(TAG, "end onHandleIntent");
    }
}

```



10-12

10-12onHandleIntent30onStartCommand30

10.2 HTTP

HTTPConnectivityManagerAppJSONURLConnection
GETPOST“”URLConnection

10.2.1

```
“”AndroidApp  
AndroidManifest.xml  
<!-- -->  
<uses-permission android:name="android.permission.INTERNET" /> <!--  
-->  
<uses-permission  
android:name="android.permission.ACCESS_NETWORK_STATE" /> <uses-  
permission android:name="android.permission.ACCESS_WIFI_STATE" />  
ConnectivityManagerContext.CONNECTIVITY_SERVICE  
getActiveNetworkInfoNetworkInfoNetworkInfo  
getType10-2
```

10-2

ConnectivityManager 类的网络类型	说明
TYPE_WIFI	wifi
TYPE_MOBILE	数据连接
TYPE_WIMAX	wimax
TYPE_ETHERNET	以太网
TYPE_BLUETOOTH	蓝牙
TYPE_VPN	vpn

getState10-3

10-3

NetworkInfo.State 的网络状态	说明
CONNECTING	正在连接
CONNECTED	已连接
SUSPENDED	挂起
DISCONNECTING	正在断开
DISCONNECTED	已断开
UNKNOWN	未知

getSubtype2G/3G/4GCDMAEVDOHSDPALTE10-

4

10-4

取值	TelephonyManager 类的网络子类型	制式分类
1	NETWORK_TYPE_GPRS	2G
2	NETWORK_TYPE_EDGE	2G
3	NETWORK_TYPE_UMTS	3G
4	NETWORK_TYPE_CDMA	2G
5	NETWORK_TYPE_EVDO_0	3G
6	NETWORK_TYPE_EVDO_A	3G
7	NETWORK_TYPE_1xRTT	2G
8	NETWORK_TYPE_HSDPA	3G
9	NETWORK_TYPE_HSUPA	3G
10	NETWORK_TYPE_HSPA	3G
11	NETWORK_TYPE_IDEN	2G
12	NETWORK_TYPE_EVDO_B	3G
13	NETWORK_TYPE_LTE	4G
14	NETWORK_TYPE_EHRPD	3G
15	NETWORK_TYPE_HSPAP	3G
16	NETWORK_TYPE_GSM	2G
17	NETWORK_TYPE_TD_SCDMA	3G
18	NETWORK_TYPE_IWLAN	4G

10-1310-1410-13WIFI10-144G

network

当前网络连接的状态是已连接
当前联网的网络类型是WIFI

10-13WIFI

network

当前网络连接的状态是已连接
当前联网的网络类型是LTE 4G

10-14

10.2.2JSON

JSONXMLAppJSONJSONXML

JSONXMLJSON

AndroidJSONJSONObject(JSON)JSONArray(JSON)

1. JSONObject

JSONObject

JSONObjectJSONObject

getJSONJSONObjectJSONObject

getString
getInt
getDouble
getBoolean
getJSONArrayJSONArray
putJSONObject
toStringJSONObjectJSON

2. JSONArray

JSONArray
lengthJSONArray
getJSONObjectJSONArrayJSONObject
putJSONArrayJSONObject
JSONJSONJSON

```
// 构造 JSON 串  
private String getJsonStr() {
```



```

String str = "";
JSONObject obj = new JSONObject();
try {
    obj.put("name", "address");
    JSONArray array = new JSONArray();
    for (int i = 0; i < 3; i++) {
        JSONObject item = new JSONObject();
        item.put("item", "第" + (i + 1) + "个元素");
        array.put(item);
    }
    obj.put("list", array);
    obj.put("count", array.length());
    obj.put("desc", "这是测试串");
    str = obj.toString();
} catch (JSONException e) {
    e.printStackTrace();
}
return str;
}

// 解析 JSON 串
private String parserJson(String jsonStr) {
    String result = "";
    try {
        JSONObject obj = new JSONObject(jsonStr);
        String name = obj.getString("name");
        String desc = obj.getString("desc");
        int count = obj.getInt("count");
        result = String.format("%sname=%s\n", result, name);
        result = String.format("%sdesc=%s\n", result, desc);
        result = String.format("%scount=%d\n", result, count);
        JSONArray listArray = obj.getJSONArray("list");
        for (int i=0; i<listArray.length(); i++) {
            JSONObject list_item = listArray.getJSONObject(i);
            String item = list_item.getString("item");
            result = String.format("%s\titem=%s\n", result, item);
        }
    } catch (JSONException e) {
        e.printStackTrace();
    }
    return result;
}
}

```

10-15 10-16 10-15 JSON 10-16 JSON



10-15JSON



10-16JSON

10.2.3HTTP

HTTPURLConnectionHttpClientJSONXML
HttpURLConnectionHttpURLConnectionHTTP()

URLConnectionURLConnection
setRequestMethodGETgetPOSTpost
setConnectTimeout
setReadTimeout
setRequestProperty
setDoOutputgetOutputStreamsetDoOutputtruePOSTPOST

getOutputStreamHTTPOutputStreamwriteflush
connectHTTPgetOutputStreamgetInputStream
setDoInputgetInputStreamsetDoInputtrue(true)
getInputStreamHTTPInputStreamread
getResponseCodeHTTP
getHeaderField
getHeaderFields
disconnectHTTP
HTTPGETPOSTGETPOSTHTTP

```

//设置默认的连接属性信息
private static void setConnHeader(HttpURLConnection conn, String method, HttpReqData req_data)
    throws ProtocolException {
    conn.setRequestMethod(method);
    conn.setConnectTimeout(5000);
    conn.setReadTimeout(10000);
    conn.setRequestProperty("Accept", "*/*");
    conn.setRequestProperty("Accept-Language", "zh-CN");
    conn.setRequestProperty("Accept-Encoding", "gzip, deflate");
    if (req_data.content_type.equals("") != true) {
        conn.setRequestProperty("Content-Type", req_data.content_type);
    }
}

//get 文本数据
public static HttpRespData getData(HttpReqData req_data) {
    HttpRespData resp_data = new HttpRespData();
    try {
        URL url = new URL(req_data.url);
        HttpURLConnection conn = (HttpURLConnection) url.openConnection();
        setConnHeader(conn, "GET", req_data);
        conn.connect();
        resp_data.content = StreamTool.getUnzipStream(conn.getInputStream(),
            conn.getHeaderField("Content-Encoding"), req_data.charset);
        resp_data.cookie = conn.getHeaderField("Set-Cookie");
        conn.disconnect();
    } catch (Exception e) {
        e.printStackTrace();
        resp_data.err_msg = e.getMessage();
    }
    return resp_data;
}

//post 的内容放在 url 中
public static HttpRespData postUrl(HttpReqData req_data) {
    HttpRespData resp_data = new HttpRespData();
    String s_url = req_data.url;
    if (req_data.params != null) {
        s_url += "?" + req_data.params.toString();
    }
    Log.d(TAG, "s_url="+s_url);
    try {
        URL url = new URL(s_url);

```

```

        HttpURLConnection conn = (HttpURLConnection) url.openConnection();
        setConnHeader(conn, "POST", req_data);
        conn.setDoOutput(true);
        conn.connect();
        resp_data.content = StreamTool.getUnzipStream(conn.getInputStream(),
            conn.getHeaderField("Content-Encoding"), req_data.charset);
        resp_data.cookie = conn.getHeaderField("Set-Cookie");
        conn.disconnect();
    } catch (Exception e) {
        e.printStackTrace();
        resp_data.err_msg = e.getMessage();
    }
    return resp_data;
}

//post 的内容放在输出流中
public static HttpRespData postData(HttpReqData req_data) {
    req_data.content_type = "application/x-www-form-urlencoded";
    HttpRespData resp_data = new HttpRespData();
    String s_url = req_data.url;
    Log.d(TAG, "s_url="+s_url+", params="+req_data.params.toString());
    try {
        URL url = new URL(s_url);
        HttpURLConnection conn = (HttpURLConnection) url.openConnection();
        setConnHeader(conn, "POST", req_data);
        conn.setDoOutput(true);
        conn.setDoInput(true);
        conn.connect();
        PrintWriter out = new PrintWriter(conn.getOutputStream());
        out.print(req_data.params.toString());
        out.flush();
        resp_data.content = StreamTool.getUnzipStream(conn.getInputStream(),
            conn.getHeaderField("Content-Encoding"), req_data.charset);
        resp_data.cookie = getRespCookie(conn, req_data);
        conn.disconnect();
    } catch (Exception e) {
        e.printStackTrace();
        resp_data.err_msg = e.getMessage();
    }
    return resp_data;
}
}
}

```

HTTPURL

1. URL

GETURLUTF8“”“%E4%BD%A0”UTF8

“%E4%BD%A0”“”URLtoUTF8.java

2.

HTTPAccept-Encodinggzip,deflategzipContent-Encoding
gzipGZIPInputStreamStreamTool.java

9Google MapAPIHTTPJSONJSON

AsyncTaskURLConnection

```
public class GetAddressTask extends AsyncTask<Location, Void, String> {
    private final static String TAG = "GetAddressTask";
    private static String mAddressUrl = "http://maps.google.cn/maps/api/geocode/json?latlng={0},
{1}&sensor=true&language=zh-CN";
    public GetAddressTask() {
        super();
    }

    @Override
    protected String doInBackground(Location... params) {
        Location location = params[0];
        String url = MessageFormat.format(mAddressUrl, location.getLatitude(), location.getLongitude());
        HttpReqData req_data = new HttpReqData(url);
        HttpRespData resp_data = HttpRequestUtil.getData(req_data);
        Log.d(TAG, "return json = " + resp_data.content);

        String address = "未知";
        if (resp_data.err_msg.length() <= 0) {
            try {
```

```

        JSONObject obj = new JSONObject(resp_data.content);
        JSONArray resultArray = obj.getJSONArray("results");
        if (resultArray.length() > 0) {
            JSONObject resultObj = resultArray.getJSONObject(0);
            address = resultObj.getString("formatted_address");
        }
    } catch (JSONException e) {
        e.printStackTrace();
    }
}
Log.d(TAG, "address = " + address);
return address;
}

@Override
protected void onPostExecute(String address) {
    mListener.onFindAddress(address);
}

private OnAddressListener mListener;
public void setOnAddressListener(OnAddressListener listener) {
    mListener = listener;
}

public static interface OnAddressListener {
    public abstract void onFindAddress(String address);
}
}

```

ActivityOnAddressListeneronFindAddress

```

    GetAddressTask addressTask = new GetAddressTask();
    addressTask.setOnAddressListener(this); addressTask.execute(location);

```

network

定位类型=gps

定位对象信息如下：

其中时间：2016-11-14 15:40:12

其中经度：119.330222，纬度：26.047125

其中高度：0米，精度：1米

其中地址：中国福建省福州市仓山区工农
路249号-375号

10-17HTTP

10-17

10.2.4HTTP

HTTPURLConnection

URLConnection


```
//get 图片数据
public static HttpRespData getImage(HttpReqData req_data) {
    HttpRespData resp_data = new HttpRespData();
    try {
        URL url = new URL(req_data.url);
        HttpURLConnection conn = (HttpURLConnection) url.openConnection();
        setConnHeader(conn, "GET", req_data);
        conn.connect();
        InputStream is = conn.getInputStream();
        resp_data.bitmap = BitmapFactory.decodeStream(is);
        resp_data.cookie = conn.getHeaderField("Set-Cookie");
        conn.disconnect();
    } catch (Exception e) {
        e.printStackTrace();
        resp_data.err_msg = e.getMessage();
    }
    return resp_data;
}
```

HTTPAsyncTask

```

public class GetImageCodeTask extends AsyncTask<Void, Void, String> {
    private final static String TAG = "GetImageCodeTask";
    private Context mContext;
    private String mImageCodeUrl = "http://220.160.54.47:82/JSPORTLET/radomImage?x=";

    public GetImageCodeTask(Context context) {
        super();
        mContext = context;
    }

    @Override
    protected String doInBackground(Void... params) {
        String url = mImageCodeUrl + DateUtil.getNowDateTime(null);
        Log.d(TAG, "image url="+url);
        HttpReqData req_data = new HttpReqData(url);
        HttpRespData resp_data = HttpRequestUtil.getImage(req_data);
        String path = BitmapUtil.getCachePath(mContext) + DateUtil.getNowDateTime(null) + ".jpg";
        BitmapUtil.saveBitmap(path, resp_data.bitmap, "jpg", 80);
        Log.d(TAG, "image path="+path);
        return path;
    }

    @Override
    protected void onPostExecute(String path) {
        mListener.onGetCode(path);
    }

    private OnImageCodeListener mListener;
    public void setOnImageCodeListener(OnImageCodeListener listener) {
        mListener = listener;
    }

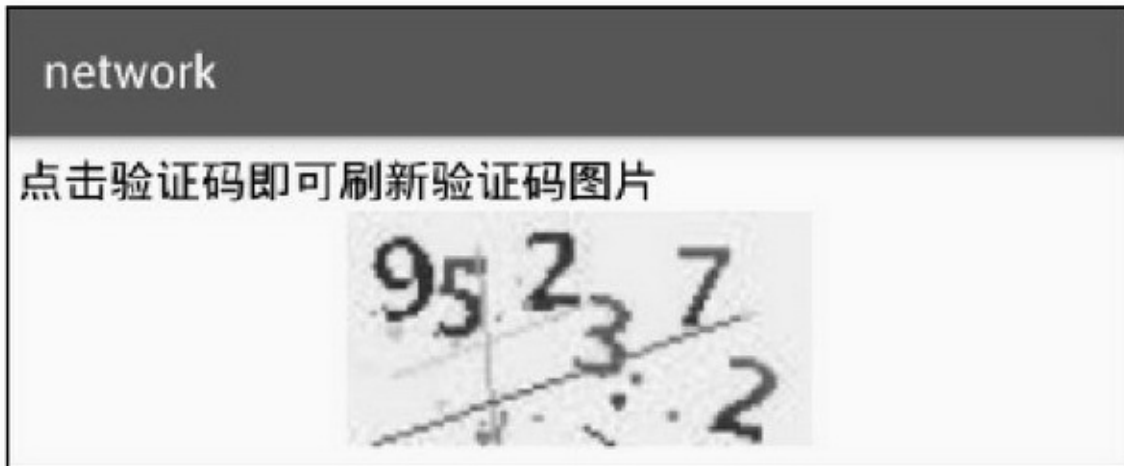
    public static interface OnImageCodeListener {
        public abstract void onGetCode(String path);
    }
}

```

taskonGetCode

```
private void getImageCode() {  
    if (bRunning != true) {  
        bRunning = true;  
        GetImageCodeTask codeTask = new GetImageCodeTask(this);  
        codeTask.setOnImageCodeListener(this);  
        codeTask.execute();  
    }  
}  
  
@Override  
public void onClick(View v) {  
    if (v.getId() == R.id.iv_image_code) {  
        getImageCode();  
    }  
}  
  
@Override  
public void onGetCode(String path) {  
    Uri uri = Uri.parse(path);  
    iv_image_code.setImageURI(uri);  
    bRunning = false;  
}
```

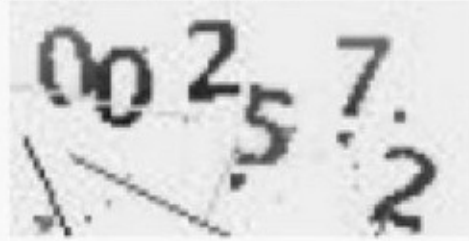
10-1810-1910-1810-19



10-18

network

点击验证码即可刷新验证码图片



10.3

AppDownloadManager33()Fragment
URLConnectionPOST

10.3.1DownloadManager 10.2URLConnection

(1)

(2)

(3)

10.2Android2.3(API9)—— DownloadManager

DownloadManagerContext.DOWNLOAD_SERVICE3

1.

DownloadManagerRequest

setAllowedNetworkTypes10-5“|”

10-5

DownloadManager.Request 类的允许网络类型	说明
NETWORK_WIFI	WIFI 网络
NETWORK_MOBILE	数据连接网络
NETWORK_BLUETOOTH	蓝牙网络

setDestinationInExternalFilesDir44-2

“-1”“-2”

addRequestHeaderHTTP

setMimeType

setTitle

setDescription

setVisibleInDownloadsUi

setNotificationVisibility10-6

10-6

DownloadManager.Request 类的通知可见类型	说明
VISIBILITY_HIDDEN	隐藏
VISIBILITY_VISIBLE	下载时可见（下载完成后消失）
VISIBILITY_VISIBLE_NOTIFY_COMPLETED	下载进行时与完成后都可见
VISIBILITY_VISIBLE_NOTIFY_ONLY_COMPLETION	只有下载完成后可见

2.

DownloadManager

enqueue

remove

restartDownload

openDownloadedFile

getMimeTypeForDownloadedFile

query

3.

AppqueryQueryCursorCursorSQLiteCursor4“4.2

SQLite”

Query

setFilterById

setFilterByStatus

setOnlyIncludeVisibleInDownloadsUi

orderBy

DownloadManagerquery10-7

10-7

DownloadManager 类的下载字段	说明
COLUMN_LOCAL_FILENAME	下载文件的本地保存路径
COLUMN_MEDIA_TYPE	下载文件的媒体类型
COLUMN_TOTAL_SIZE_BYTES	下载文件的总大小
COLUMN_BYTES_DOWNLOADED_SO_FAR	已下载的文件大小
COLUMN_STATUS	下载状态。下载状态的取值说明见表 10-8

10-8

DownloadManager 类的下载状态	说明
STATUS_PENDING	挂起, 即正在等待
STATUS_RUNNING	运行中
STATUS_PAUSED	暂停
STATUS_SUCCESSFUL	成功
STATUS_FAILED	失败

33

1.

DownloadManager.ACTION_DOWNLOAD_COMPLETE(
android.intent.action.DOWNLOAD_COMPLETE)

2.

DownloadManager.ACTION_NOTIFICATION_CLICKED(
android.intent.action.DOWNLOAD_NOTIFICATION_CLICKED)

3.

DownloadManager.ACTION_NOTIFICATION_CLICKED
Intent.ACTION_VIEW()AppRequestsetMimeType

AndroidApp

DownloadManagerAPK

```

private class ApkUrlSelectedListener implements OnItemSelectedListener {
    public void onItemSelected(AdapterView<?> arg0, View arg1, int arg2, long arg3) {
        sp_apk_url.setEnabled(false);
        Uri uri = Uri.parse(apkUrlArray[arg2]);
        Request down = new Request(uri);
        down.setTitle(apkDescArray[arg2]+"下载信息");
        down.setDescription(apkDescArray[arg2]+"安装包正在下载");
        down.setAllowedNetworkTypes(Request.NETWORK_MOBILE | Request.NETWORK_WIFI);
        down.setNotificationVisibility(Request.VISIBILITY_VISIBLE_NOTIFY_COMPLETED);
        down.setVisibleInDownloadsUi(true);
        down.setDestinationInExternalFilesDir(DownloadApkActivity.this,
            Environment.DIRECTORY_DOWNLOADS, arg2 + ".apk");
        mDownloadId = mDownloadManager.enqueue(down);
    }

    public void onNothingSelected(AdapterView<?> arg0) {
    }
}

// 接收下载完成事件
public static class DownloadCompleteReceiver extends BroadcastReceiver {
    @Override
    public void onReceive(Context context, Intent intent) {
        if (intent.getAction().equals(DownloadManager.ACTION_DOWNLOAD_COMPLETE)
            && tv_apk_result != null) {
            long downId = intent.getLongExtra(DownloadManager.EXTRA_DOWNLOAD_ID, -1);
            Log.d(TAG, "download complete! id : " + downId + ", mDownloadId=" +
mDownloadId);

            tv_apk_result.setVisibility(View.VISIBLE);
            tv_apk_result.setText(DateUtil.getNowDateTime(null) + " 编号"
                + downId + "的下载任务已完成");
            sp_apk_url.setEnabled(true);
        }
    }
}

```



```

    }

    // 接收下载通知栏的点击事件，在下载过程中有效，下载完成后失效
    public static class NotificationClickReceiver extends BroadcastReceiver {
        @Override
        public void onReceive(Context context, Intent intent) {
            Log.d(TAG, " NotificationClickReceiver onReceive");
            if (intent.getAction().equals(DownloadManager.ACTION_NOTIFICATION_CLICKED)
                && tv_apk_result != null) {
                long[] downloadIds = intent.getLongArrayExtra(DownloadManager.EXTRA_
NOTIFICATION_CLICK_DOWNLOAD_IDS);
                for (long downloadId : downloadIds) {
                    Log.d(TAG, " notify click! id : " + downloadId + ", mDownloadId=" + mDownloadId);
                    if (downloadId == mDownloadId) {
                        tv_apk_result.setText(DateUtil.getNowDateTime(null) + " 编号"
                            + downloadId + "的下载进度条被点击了一下");
                    }
                }
            }
        }
    }
}

```

AndroidManifest.xml

```

<receiver android:name=".DownloadApkActivity$DownloadCompleteReceiver" >
    <intent-filter>
        <action android:name="android.intent.action.DOWNLOAD_COMPLETE" />
    </intent-filter>
</receiver>

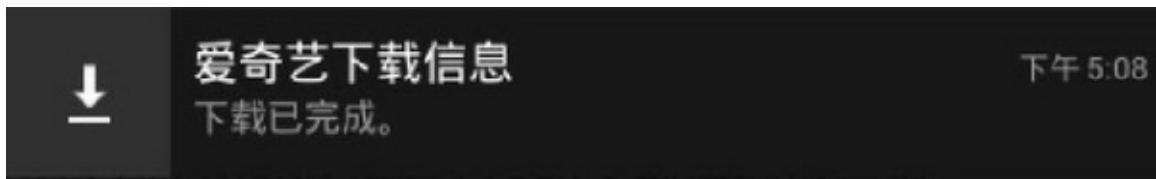
<receiver android:name=".DownloadApkActivity$NotificationClickReceiver" >
    <intent-filter>
        <action android:name="android.intent.action.DOWNLOAD_NOTIFICATION_CLICKED" />
    </intent-filter>
</receiver>

```

APK10-2010-2110-2010-21



10-20



10-21

App

```
private Handler mHandler = new Handler();
private Runnable mRefresh = new Runnable() {
    @Override
    public void run() {
        boolean bFinish = false;
        Query down_query = new Query();
        down_query.setFilterById(mDownloadId);
        Cursor cursor = mDownloadManager.query(down_query);
        if (cursor.moveToFirst()) {
            for (; cursor.moveToNext()) {
                int nameIdx = cursor.getColumnIndex(DownloadManager.COLUMN_LOCAL_
FILENAME);
                int mediaTypeIdx = cursor.getColumnIndex(DownloadManager.COLUMN_
MEDIA_TYPE);
                int totalSizeIdx = cursor.getColumnIndex(DownloadManager.COLUMN_
TOTAL_SIZE_BYTES);
                int nowSizeIdx = cursor.getColumnIndex(DownloadManager.COLUMN_
BYTES_DOWNLOADED_SO_FAR);
                int statusIdx = cursor.getColumnIndex(DownloadManager.COLUMN_STATUS);
                int progress = (int) (100 * cursor.getLong(nowSizeIdx) / cursor.getLong(totalSizeIdx));
                if (cursor.getString(nameIdx) == null) {
                    break;
                }
                tpc_progress.setProgress(progress, 100);
                mImagePath = cursor.getString(nameIdx);
                String desc = "";
                desc = String.format("%s 文件路径 : %s\n", desc, cursor.getString(nameIdx));
                desc = String.format("%s 媒体类型 : %s\n", desc, cursor.getString(mediaTypeIdx));
                desc = String.format("%s 文件总大小 : %d\n", desc, cursor.getLong(totalSizeIdx));
                desc = String.format("%s 已下载大小 : %d\n", desc, cursor.getLong(nowSizeIdx));
                desc = String.format("%s 下载进度 : %d%%\n", desc, progress);
                desc = String.format("%s 下载状态 : %s\n", desc, mStatusMap.get(cursor.
getInt(statusIdx)));
                tv_image_result.setText(desc);
                if (progress == 100) {
                    bFinish = true;
                }
                if (cursor.isLast() == true) {
                    break;
                }
            }
        }
    }
}
```

```
        }  
    }  
    cursor.close();  
    if (bFinish != true) {  
        mHandler.postDelayed(this, 100);  
    } else {  
        sp_image_url.setEnabled(true);  
        tpc_progress.setVisibility(View.INVISIBLE);  
        iv_image_url.setImageURI(Uri.parse(mImagePath));  
    }  
}  
};
```

VISIBILITY_HIDDENAndroidManifest.xml

<!-- -->

<uses-permission

android:name="android.permission.DOWNLOAD_WITHOUT_NOTIFICATION

/> 10-2210-2310.110-224%10-23



10-22



10-23

10.3.2

Android6“6.3”DialogFragment

5“5.3Fragment”DialogFragmentFragmentFragment

FragmentDialogFragment

DialogFragmentonCreateDialog

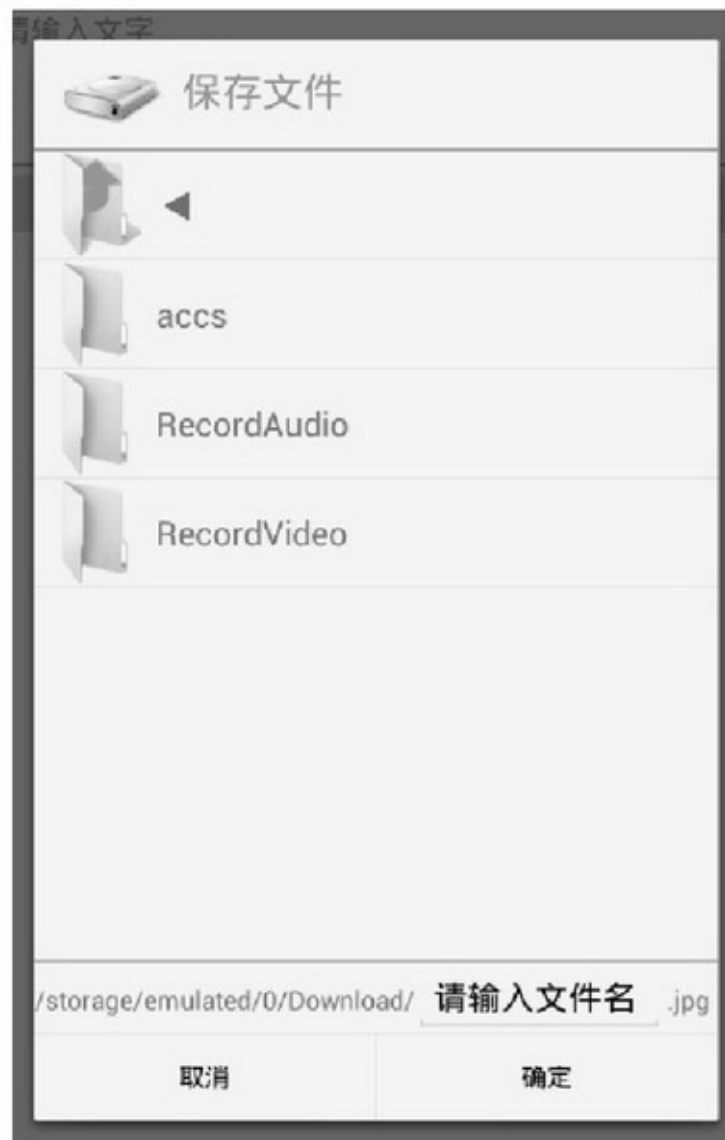
ListFragmentListActivity

PreferenceFragmentPreferenceActivityAndroid“”

PreferenceFragment

WebViewFragment

10-2410-2510-2410-25



10-24



10-24

```
filedialogfiledialogfiledialogbuild.gradle
dependenciesfiledialog
compile project(':filedialog')
```

10.3.3

App(QQ)

AndroidDownloadManagerHTTPPOST

multipart/form-data

URLConnection

```
public class HttpUploadUtil {
    public static String upload(String uploadUrl, String uploadFile) {
        String fileName = "";
        int pos = uploadFile.lastIndexOf("/");
        if (pos >= 0) {
            fileName = uploadFile.substring(pos + 1);
        }
        String end = "\r\n";
        String Hyphens = "--";
        String boundary = "WUm4580jbtwflJhNp7zi1djFEO3wNNm";
        try {
            URL url = new URL(uploadUrl);
            HttpURLConnection conn = (HttpURLConnection) url.openConnection();
            conn.setDoInput(true);
            conn.setDoOutput(true);
            conn.setUseCaches(false);
            conn.setRequestMethod("POST");
            conn.setRequestProperty("Connection", "Keep-Alive");
            conn.setRequestProperty("Charset", "UTF-8");
            conn.setRequestProperty("Content-Type", "multipart/form-data;boundary=" + boundary);

            DataOutputStream ds = new DataOutputStream(conn.getOutputStream());
            ds.writeBytes(Hyphens + boundary + end);
            ds.writeBytes("Content-Disposition: form-data; "
                + "name=\"file1\";filename=\"" + fileName + "\"" + end);
            ds.writeBytes(end);
            FileInputStream fStream = new FileInputStream(uploadFile);
            // 每次写入 1024 字节
            int bufferSize = 1024;
            byte[] buffer = new byte[bufferSize];
            int length = -1;
            // 将文件数据写入缓冲区
            while((length = fStream.read(buffer)) != -1) {
                ds.write(buffer, 0, length);
            }
            ds.writeBytes(end);
            ds.writeBytes(Hyphens + boundary + Hyphens + end);
            fStream.close();
        }
    }
}
```



```
        ds.flush();
        // 获取返回内容
        InputStream is = conn.getInputStream();
        int ch;
        StringBuffer b = new StringBuffer();
        while ((ch = is.read()) != -1) {
            b.append((char) ch);
        }
        ds.close();
        return "SUCC";
    } catch (Exception e) {
        e.printStackTrace();
        return "上传失败:" + e.getMessage();
    }
}
}
```

AsyncTask

```

public class UploadHttpTask extends AsyncTask<String, Void, String> {
    private final static String TAG = "UploadHttpTask";
    private Context mContext;

    public UploadHttpTask(Context context) {
        super();
        mContext = context;
    }

    @Override
    protected String doInBackground(String... params) {
        String uploadUrl = params[0];
        String filePath = params[1];
        Log.d(TAG, "uploadUrl=" + uploadUrl + ", filePath=" + filePath);
        String result = HttpUploadUtil.upload(uploadUrl, filePath);
        return result;
    }

    @Override
    protected void onPostExecute(String result) {
        mListener.onUploadFinish(result);
    }

    private OnUploadHttpListener mListener;
    public void setOnUploadHttpListener(OnUploadHttpListener listener) {
        mListener = listener;
    }

    public static interface OnUploadHttpListener {
        public abstract void onUploadFinish(String result);
    }
}

```

HTTPITJ2EEHTTP

步骤 01 “360WiFi”WIFIWIFI

步骤 02 WindowsWindows FirewallWIFI

步骤 03 WLANWIFI

步骤 04 Eclipsedemo——UploadTestRun As → Run on

Server

步骤

05

ipconfig /allMicrosoft Virtual WiFi Miniport Adapter

IP(10-26)AppIP

```
命令提示符
无线局域网适配器 无线网络连接 2:
连接特定的 DNS 后缀 . . . . . :
描述 . . . . . : Microsoft Virtual WiFi Miniport Adapter
物理地址 . . . . . : FA-16-54-A1-C4-30
DHCP 已启用 . . . . . : 不是
自动配置已启用 . . . . . : 是
本地连接 IPv6 地址 . . . . . : fe80::150c:8e3e:57ba:2229%34<首选>
IPv4 地址 . . . . . : 192.168.173.1<首选>
子网掩码 . . . . . : 255.255.255.0
IPv4 地址 . . . . . : 192.168.253.1<首选>
子网掩码 . . . . . : 255.255.255.0
默认网关 . . . . . :
DHCPv6 IAD . . . . . : 804918868
DHCPv6 客户端 DUID . . . . . : 00-01-00-01-1B-90-8B-D2-54-EE-75-22-40-30

DNS 服务器 . . . . . : fec0:0:0:ffff::1%1
                          fec0:0:0:ffff::2%1
                          fec0:0:0:ffff::3%1

TCP/IP 上的 NetBIOS . . . . . : 已启用
```

10-26WIFIIP

HTTPApp(<http://192.168.253.1:8080/UploadTest/uploadServlet>)

App

10-27

network

http://192.168.0.212:8080/UploadTest/uploadServlet

HTTP上传文件

上传文件的路径为：/storage/emulated/0/

Download/10000.png

上传结果为：SUCC

预计下载地址为：http://192.168.0.212:8080/

UploadTest/10000.png

10.4 Socket SocketInetAddressSocket

10.4.1 InetAddress AppHTTP

InetAddress

getByNameIPInetAddress

getHostAddressIP

getHostName

isReachable

```
public void onClick(View v) {
    if (v.getId() == R.id.btn_host_name) {
        new CheckThread(et_host_name.getText().toString()).start();
    }
}

private Handler mHandler = new Handler() {
    @Override
    public void handleMessage(Message msg) {
        tv_host_name.setText("主机检查结果如下 : \n"+msg.obj);
    }
};

private class CheckThread extends Thread {
    private String mHostName;
    public CheckThread(String host_name) {
        mHostName = host_name;
    }

    @Override
```

```
public void run() {
    Message message = Message.obtain();
    try {
        InetAddress host = InetAddress.getByByName(mHostName);
        boolean isReachable = host.isReachable(5000);
        String desc = (isReachable)?"可以连接":"无法连接";
        if (isReachable == true) {
            desc = String.format("%s\n 主机名为%s\n 主机地址为%s",
                desc, host.getHostName(), host.getHostAddress());
        }
        message.what = 0;
        message.obj = desc;
    } catch (Exception e) {
        e.printStackTrace();
        message.what = -1;
        message.obj = e.getMessage();
    }
    mHandler.sendMessage(message);
}
```

10-2810-2910-2810-29IP





10.4.2Socket

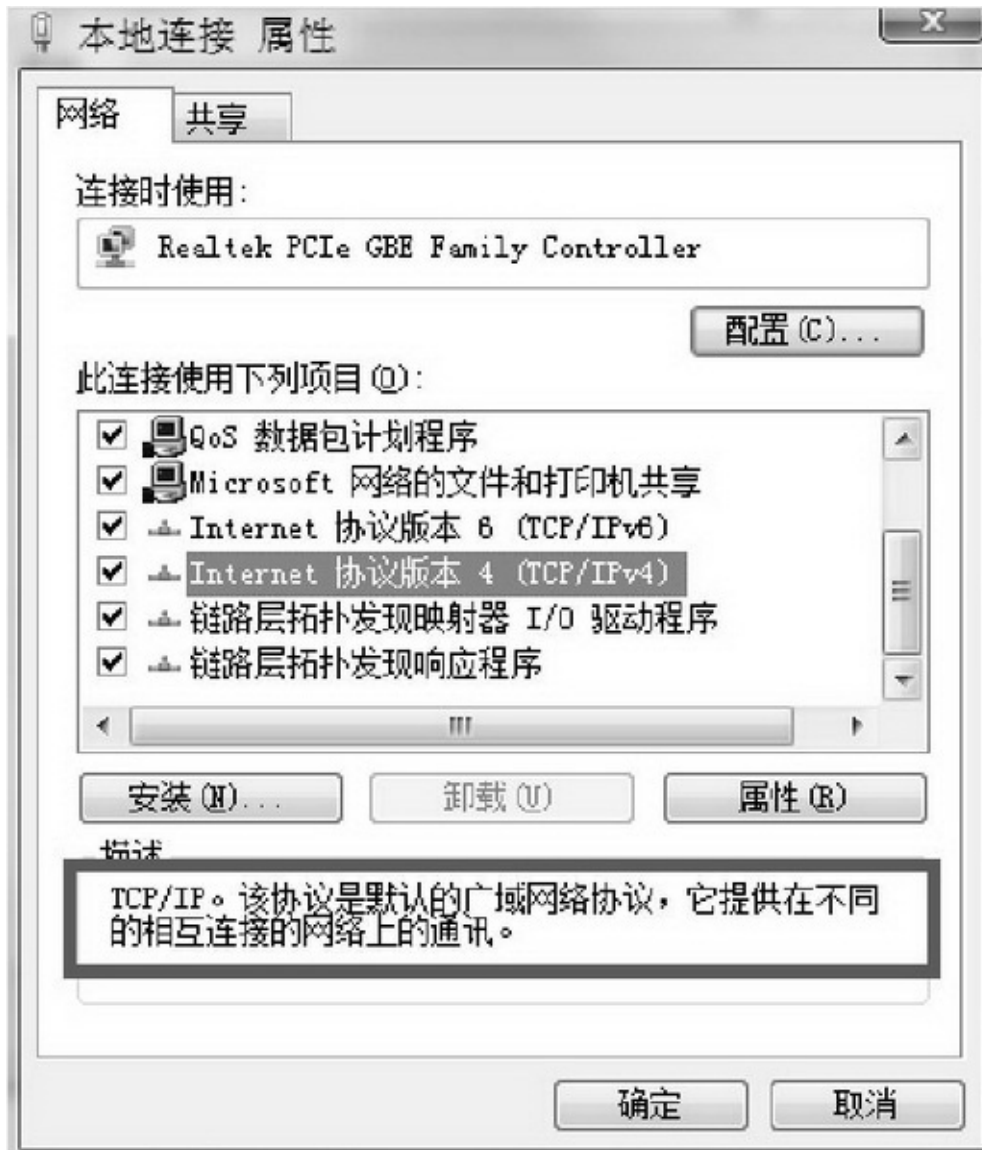
SocketTCP/IPIP10-30TCP/IP

TCP/IP3

IPICMPARPRARPBOOTP

TCPUDP

HTTPFTPTELNETSMTPDNS



10-30

HTTPSocketAPITCPHTTPUDPFTP(QQ)

Socket

AndroidSocketSocketServerSocket

1. Socket

Socket(Socket)Socket

connectIP

getInputStream
getOutputStream
getInetAddressInetAddress
isConnectedsocket
isClosedsocket
closesocket

2. ServerSocket

ServerSocketServerSocket

acceptSocket
getInetAddressInetAddress
isClosedsocket
closesocket
SocketSocket

```
public class MessageTransmit implements Runnable {  
    private static final String TAG = "MessageTransmit";  
    private static final String SOCKET_IP = "192.168.1.5"; // Socket 服务器的 IP, 根据实际情况修改  
    private static final int SOCKET_PORT = 51000; // Socket 服务器的端口, 根据实际情况修改  
    private Socket mSocket;
```

```

private BufferedReader mReader = null;
private OutputStream mWriter = null;

@Override
public void run() {
    mSocket = new Socket();
    try {
        mSocket.connect(new InetSocketAddress(SOCKET_IP, SOCKET_PORT), 3000);
        mReader = new BufferedReader(new InputStreamReader(mSocket.getInputStream()));
        mWriter = mSocket.getOutputStream();
        new RecvThread().start(); // 启动一条子线程读取服务器的返回数据
        Looper.prepare();
        Looper.loop();
    } catch (Exception e) {
        e.printStackTrace();
    }
}

// 定义接收 UI 线程的 Handler 对象，App 向后台服务器发送消息
public Handler mRecvHandler = new Handler() {
    @Override
    public void handleMessage(Message msg) {
        Log.d(TAG, "handleMessage: "+msg.obj);
        String send_msg = msg.obj.toString()+"\n";
        // 换行符相当于回车键，表示“我写好了发出去吧”

        try {
            mWriter.write(send_msg.getBytes("utf8"));
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
};

// 定义消息接收子线程，App 从后台服务器接收消息
private class RecvThread extends Thread {
    @Override
    public void run() {
        try {
            String content = null;
            while ((content = mReader.readLine()) != null) { // 读取来自服务器的数据
                Message msg = Message.obtain();
                msg.obj = content;
                SocketActivity.mHandler.sendMessage(msg);
            }
        }
    }
}

```

```

    } catch (Exception e) {
        e.printStackTrace();
    }
}
}
}
}

```

ActivitySocketSocket

```

public class SocketActivity extends AppCompatActivity implements OnClickListener {
    private EditText et_socket;
    private static TextView tv_socket;
    private MessageTransmit mTransmit;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_socket);
        et_socket = (EditText) findViewById(R.id.et_socket);
        tv_socket = (TextView) findViewById(R.id.tv_socket);
        findViewById(R.id.btn_socket).setOnClickListener(this);
        mTransmit = new MessageTransmit();
        new Thread(mTransmit).start();
    }

    @Override
    public void onClick(View v) {
        if (v.getId() == R.id.btn_socket) {
            Message msg = Message.obtain();
            msg.obj = et_socket.getText().toString();
            // 注意这里是主线程向分线程发送消息，与 IntentService 的情况类似
            mTransmit.mRecvHandler.sendMessage(msg);
        }
    }

    public static Handler mHandler = new Handler() {
        @Override
        public void handleMessage(Message msg) {
            String desc = String.format("%s 收到服务器的应答消息 : %s",
                DateUtil.getNowTime(), msg.obj.toString());
            tv_socket.setText(desc);
        }
    };
}

```

Socket(App)SocketSocketTestServer.java10.3“10.3.3”

Socket10-3110-3210-3110-32AppSocket“hello”

Socket“hi”App



10-31Socket



10-32Socket

10.5QQ

AppQQAppAppAppApp“QQ
”

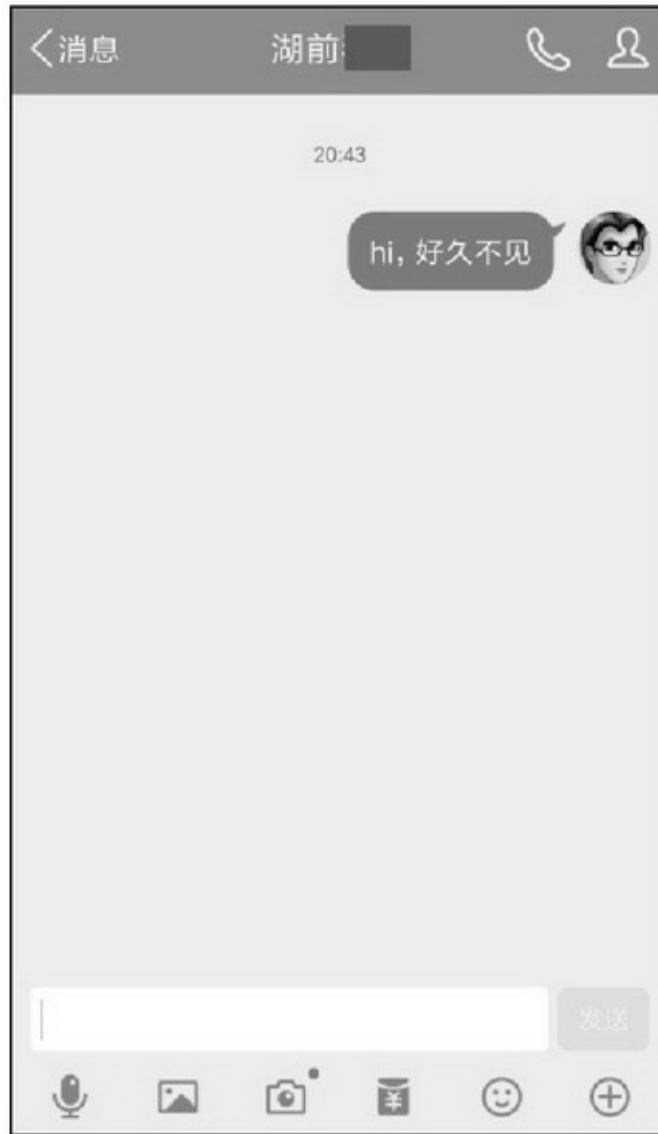
10.5.1

QQApp“”AppQQ“”

QQ10-3310-34

App10-3510-3610-3510-36





10-34





10-36

8

HTTPAppHTTP

HTTPHTTP

SocketSocket

Socket

1.

QQ3Socket3

(1)SocketSocket

(2)SocketSocket

(3)SocketASocketABSocket

2.App

(1)3

(2)SocketAppSocketSocketApplication

(3)Socket

(4)SocketBroadcast

(5)App

3.Socket

Socket

(1)SocketSocket

(2)Socket

(3)Socket

①Socket

②SocketSocket

③Socket

④SocketSocket

4.

()SocketXMLJSON

10.5.2ExpandableListView ()

LinearLayoutListViewGridViewExpandableListView

setAdapterExpandableListAdapter

expandGroup

collapseGroup
isGroupExpanded
setSelectedGroup
setSelectedChild
setGroupIndicator
setChildIndicator
setOnGroupExpandListenerOnGroupExpandListeneronGroupExpand

setOnGroupCollapseListenerOnGroupCollapseListener
onGroupCollapse
setOnGroupClickListenerOnGroupClickListeneronGroupClick

setOnChildClickListenerOnChildClickListeneronChildClick

——ExpandableListAdapter5

getGroupCount
getChildrenCount
getGroupView
getChildView
isChildSelectable
()

```

public class MailExpandAdapter implements
ExpandableListAdapter, OnGroupClickListener, OnChildClickListener {
    private LayoutInflater mInflater;
    private Context mContext;
    private ArrayList<MailBox> mBoxList;

    public MailExpandAdapter(Context context, ArrayList<MailBox> box_list) {
        mInflater = LayoutInflater.from(context);
        mContext = context;
        mBoxList = box_list;
    }

    @Override
    public int getGroupCount() {
        return mBoxList.size();
    }

    @Override
    public int getChildrenCount(int groupPosition) {
        return mBoxList.get(groupPosition).mail_list.size();
    }

    @Override
    public View getGroupView(int groupPosition, boolean isExpanded, View convertView, ViewGroup
parent) {
        ViewHolderBox holder = null;
        if (convertView == null) {
            holder = new ViewHolderBox();
            convertView = mInflater.inflate(R.layout.item_box, null);
            holder.iv_box = (ImageView) convertView.findViewById(R.id.iv_box);
            holder.tv_box = (TextView) convertView.findViewById(R.id.tv_box);
            holder.tv_count = (TextView) convertView.findViewById(R.id.tv_count);
            convertView.setTag(holder);
        } else {
            holder = (ViewHolderBox) convertView.getTag();
        }
        MailBox box = mBoxList.get(groupPosition);

```

```

        holder.iv_box.setImageResource(box.box_icon);
        holder.tv_box.setText(box.box_title);
        holder.tv_count.setText(box.mail_list.size()+"封邮件");
        return convertView;
    }

    @Override
    public View getChildView(final int groupPosition, final int childPosition,
        boolean isLastChild, View convertView, ViewGroup parent) {
        ViewHolderMail holder = null;
        if (convertView == null) {
            holder = new ViewHolderMail();
            convertView = mInflater.inflate(R.layout.item_mail, null);
            holder.ck_mail = (CheckBox) convertView.findViewById(R.id.ck_mail);
            holder.tv_date = (TextView) convertView.findViewById(R.id.tv_date);
            convertView.setTag(holder);
        } else {
            holder = (ViewHolderMail) convertView.getTag();
        }
        MailItem item = mBoxList.get(groupPosition).mail_list.get(childPosition);
        holder.ck_mail.setText(item.mail_title);
        holder.ck_mail.setOnCheckedChangeListener(new OnCheckedChangeListener() {
            @Override
            public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
                MailBox box = mBoxList.get(groupPosition);
                MailItem item = box.mail_list.get(childPosition);
                String desc = String.format("您点击了%s, 标题是%s", box.box_title, item.mail_title);
                Toast.makeText(mContext, desc, Toast.LENGTH_SHORT).show();
            }
        });
        holder.tv_date.setText(item.mail_date);
        return convertView;
    }

    @Override
    public boolean isChildSelectable(int groupPosition, int childPosition) {
        return true; //如果子条目需要响应点击事件, 这里就要返回 true
    }

    public final class ViewHolderBox {
        public ImageView iv_box;
        public TextView tv_box;
        public TextView tv_count;
    }

```

```

    }

    public final class ViewHolderMail {
        public CheckBox ck_mail;
        public TextView tv_date;
    }

    @Override
    public boolean onChildClick(ExpandableListView parent, View v, int groupPosition, int childPosition,
long id) {
        ViewHolderMail holder = (ViewHolderMail) v.getTag();
        holder.ck_mail.setChecked(!(holder.ck_mail.isChecked()));
        return true;
    }

    @Override
    public boolean onGroupClick(ExpandableListView parent, View v, int groupPosition, long id) {
        String desc = String.format("您点击了%s", mBoxList.get(groupPosition).box_title);
        Toast.makeText(mContext, desc, Toast.LENGTH_SHORT).show();
        return false; //如果返回 true, 就不会展示子列表
    }
}
}

```

```

private void initMailBox() {
    ExpandableListView elv_list = (ExpandableListView) findViewById(R.id.elv_list);
    final ArrayList<MailBox> box_list = new ArrayList<MailBox>();
    box_list.add(new MailBox(R.drawable.mail_folder_inbox, "收件箱", getRecvMail()));
    box_list.add(new MailBox(R.drawable.mail_folder_outbox, "发件箱", getSentMail()));
    box_list.add(new MailBox(R.drawable.mail_folder_draft, "草稿箱", getDraftMail()));
    box_list.add(new MailBox(R.drawable.mail_folder_recycle, "废件箱", getRecycleMail()));
    MailExpandAdapter adapter = new MailExpandAdapter(this, box_list);
    elv_list.setAdapter(adapter);
    elv_list.setOnChildClickListener(adapter);
    elv_list.setOnGroupClickListener(adapter);
    elv_list.expandGroup(0); //默认展开第一个邮件夹
}

```

10-3710-3810-3710-38

network		
OO邮箱		
	收件箱	5封邮件
	这里是收件箱呀1	2016年11月15日
	这里是收件箱呀2	2016年11月10日
	这里是收件箱呀3	2016年11月14日
	这里是收件箱呀4	2016年11月11日
	这里是收件箱呀5	2016年11月13日
	发件箱	5封邮件
	草稿箱	5封邮件
	废件箱	5封邮件

network	
OO邮箱	
 收件箱	5封邮件
 发件箱	5封邮件
 邮件发出去了吗1	2016年11月15日
 邮件发出去了吗2	2016年11月14日
 邮件发出去了吗3	2016年11月11日
 邮件发出去了吗4	2016年11月13日
 邮件发出去了吗5	2016年11月10日
 草稿箱	5封邮件
 废件箱	5封邮件

10-38

3

(1)isChildSelectabletrue

(2)setOnChildClickListeneronChildClick

(3)ButtonEditTextsetFocusablesetFocusableInTouchMode

false5“5.2.2ListView”descendantFocusability

android:descendantFocusability="blocksDescendants"

10.5.3

(1)AndroidManifest.xml

```
<!-- -->
<uses-permission android:name="android.permission.INTERNET" /> <!--
-->
<uses-permission
android:name="android.permission.ACCESS_NETWORK_STATE" /> <uses-
permission android:name="android.permission.ACCESS_WIFI_STATE" /> <!--
SD -->
<uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission
android:name="android.permission.READ_EXTERNAL_STORAGE" /> <uses-
permission
android:name="android.permission.MOUNT_UNMOUNT_FILESYSTEMS" />
(2)AndroidManifest.xmlapplicationandroid:name=".MainApplication"
```

```
<receiver android:name=".ChatMainActivity$RecvMsgReceiver" >
  <intent-filter>
    <action android:name="com.example.network.RECV_MSG" />
  </intent-filter>
</receiver>

<receiver android:name=".QQContactActivity$GetListReceiver" >
  <intent-filter>
    <action android:name="com.example.network.GET_LIST" />
  </intent-filter>
</receiver>
```

3

(1)UploadTestUploadServlet.java

(2)UploadTestQueryFriend.java

(3)SocketSocketChatServer.java

请输入您的昵称

登 录

10-39

UploadTestSocketApp(AppURLIP)App10-39

10-4010-4110-40A“”B“”10-41





10-41

10-42~10-4510-4210-43A(“”)10-4410-45B(“”)

network

与大山聊天

大山 2016-11-14 23:38:06
hi, 好久不见

轻舞飞扬 2016-11-15 00:27:12
嘻嘻, 是的呀

大山 2016-11-14 23:39:16
周末我去公园了, 好多鲜花, 你呢?

大山 2016-11-14 23:39:49



输入



发送



10-432

network

与轻舞飞扬聊天

大山 2016-11-14 23:38:06
hi, 好久不见

轻舞飞扬 2016-11-14 23:38:36
嘻嘻, 是的呀

大山 2016-11-14 23:39:15
周末我去公园了, 好多鲜花, 你呢?

大山 2016-11-14 23:39:48



输入



发送

10-441



10-452

QQ3

(1)9

(2)4SQLiteSQLite

(3)9""App

App


```

public class QQContactActivity extends AppCompatActivity implements
    OnClickListener, OnQueryFriendListener {
    private final static String TAG = "QQContactActivity";
    private static Context mContext;
    private static ExpandableListView elv_friend;
    private static ArrayList<FriendGroup> mGroupList;
    private static FriendGroup mGroupOnline = new FriendGroup();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_qq_contact);
        Toolbar tl_head = (Toolbar) findViewById(R.id.tl_head);
        tl_head.setTitle(getResources().getString(R.string.menu_second));
        setSupportActionBar(tl_head);
        tl_head.setNavigationOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View view) {
                finish();
            }
        });
        mContext = getApplicationContext();
        mGroupOnline.title = "在线好友";
        mGroupList = new ArrayList<FriendGroup>();
        elv_friend = (ExpandableListView) findViewById(R.id.elv_friend);
        findViewById(R.id.btn_refresh).setOnClickListener(this);
        QueryFriendTask queryTask = new QueryFriendTask(this);
        queryTask.setOnQueryFriendListener(this);
        queryTask.execute();
    }
}

```

```

    }

    @Override
    public void onQueryFriend(String resp) {
        try {
            JSONObject obj = new JSONObject(resp);
            JSONArray groupArray = obj.getJSONArray("group_list");
            for (int i=0; i<groupArray.length(); i++) {
                JSONObject groupObj = groupArray.getJSONObject(i);
                FriendGroup group = new FriendGroup();
                group.title = groupObj.getString("title");
                JSONArray friendArray = groupObj.getJSONArray("friend_list");
                for (int j=0; j<friendArray.length(); j++) {
                    JSONObject friendObj = friendArray.getJSONObject(j);
                    Friend friend = new Friend("", friendObj.getString("name"), "");
                    group.friend_list.add(friend);
                }
                mGroupList.add(group);
            }
            showAllFriend();
        } catch (JSONException e) {
            e.printStackTrace();
            Toast.makeText(this, "获取全部好友列表出错 : "+e.getMessage(),
                Toast.LENGTH_SHORT).show();
        }
    }

    @Override
    protected void onResume() {
        mHandler.postDelayed(mRefresh, 500);
        super.onResume();
    }

    @Override
    protected void onDestroy() {
        MainApplication.getInstance().sendAction(ClientThread.LOGOUT, "", "");
        super.onDestroy();
    }

    private Handler mHandler = new Handler();
    private Runnable mRefresh = new Runnable() {
        @Override
        public void run() {

```


10.6

App()HTTP(HTTPHTTP)(
)Socket“QQ”App

4

(1)

(2)HTTP

(3)

(4)Socket

第11章 事件



App“——”

11.1

App“”

11.1.1

App3EditTextTextWatcher
setOnKeyListenerOnKeyListeneronKey
11-1OnKeyListener()

11-1

按键编码	KeyEvent 类的按键名称	说明
3	KEYCODE_HOME	主页键（未开放给普通 App）
4	KEYCODE_BACK	返回键（后退键）
24	KEYCODE_VOLUME_UP	加大音量键
25	KEYCODE_VOLUME_DOWN	减小音量键
26	KEYCODE_POWER	电源键（未开放给普通 App）
66	KEYCODE_ENTER	回车键
67	KEYCODE_DEL	删除键（退格键）
82	KEYCODE_MENU	菜单键
84	KEYCODE_SEARCH	搜索键
187	KEYCODE_APP_SWITCH	任务键（未开放给普通 App）

onKey(KeyEvent.ACTION_DOWN)
(KeyEvent.ACTION_UP)

```

public class KeySoftActivity extends AppCompatActivity implements OnKeyListener {
    private EditText et_soft;
    private TextView tv_result;
    private String desc = "";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_key_soft);
        et_soft = (EditText) findViewById(R.id.et_soft);
        tv_result = (TextView) findViewById(R.id.tv_result);
        et_soft.setOnKeyListener(this);
    }

    @Override
    public boolean onKey(View v, int keyCode, KeyEvent event) {
        if (event.getAction() == KeyEvent.ACTION_DOWN) {
            desc = String.format("%s 输入的软按键编码是%d,动作是按下", desc, keyCode);
            if (keyCode == KeyEvent.KEYCODE_ENTER) {
                desc = String.format("%s, 按键为回车键", desc);
            } else if (keyCode == KeyEvent.KEYCODE_DEL) {
                desc = String.format("%s, 按键为删除键", desc);
            } else if (keyCode == KeyEvent.KEYCODE_BACK) {
                desc = String.format("%s, 按键为返回键", desc);
                mHandler.postDelayed(mFinish, 3000);
            } else if (keyCode == KeyEvent.KEYCODE_MENU) {
                desc = String.format("%s, 按键为菜单键", desc);
            } else if (keyCode == KeyEvent.KEYCODE_VOLUME_UP) {
                desc = String.format("%s, 按键为加大音量键", desc);
            } else if (keyCode == KeyEvent.KEYCODE_VOLUME_DOWN) {
                desc = String.format("%s, 按键为减小音量键", desc);
            }
            desc = desc + "\n";
            tv_result.setText(desc);
            return true;
        } else {
            return false; //返回 true 表示处理完了不再输入该字符, 返回 false 表示输入该字符
        }
    }

    private Handler mHandler = new Handler();
    private Runnable mFinish = new Runnable() {
        @Override

```

```

        public void run() {
            finish();
        }
    };
}

```

11-13AppApp

event

hello

输入的软按键编码是66,动作是按下, 按键为回车键
输入的软按键编码是67,动作是按下, 按键为删除键
输入的软按键编码是24,动作是按下, 按键为加大音量键
输入的软按键编码是25,动作是按下, 按键为减小音量键
输入的软按键编码是82,动作是按下, 按键为菜单键
输入的软按键编码是4,动作是按下, 按键为返回键

11-1

11.1.2

ActivityonKeyDownonKeyDownonKeyKeyEvent

(1)onKeyDownActivityonKey

(2)onKeyDown()onKey

(3)onKeyDownonKey


```

@Override
public boolean onKeyDown(int keyCode, KeyEvent event) {
    desc = String.format("%s 物理按键的编码是%d", desc, keyCode);
    if (keyCode == KeyEvent.KEYCODE_BACK) {
        desc = String.format("%s, 按键为返回键", desc);
        mHandler.postDelayed(mFinish, 3000);
    } else if (keyCode == KeyEvent.KEYCODE_MENU) {
        desc = String.format("%s, 按键为菜单键", desc);
    } else if (keyCode == KeyEvent.KEYCODE_VOLUME_UP) {
        desc = String.format("%s, 按键为加大音量键", desc);
    } else if (keyCode == KeyEvent.KEYCODE_VOLUME_DOWN) {
        desc = String.format("%s, 按键为减小音量键", desc);
    }
    desc = desc + "\n";
    tv_result.setText(desc);
    return true; //返回 true 表示不再响应系统动作，返回 false 表示继续响应系统动作
}

private Handler mHandler = new Handler();
private Runnable mFinish = new Runnable() {
    @Override
    public void run() {
        finish();
    }
};

```

event

请按设备上的物理键开始检测

物理按键的编码是24, 按键为加大音量键

物理按键的编码是25, 按键为减小音量键

物理按键的编码是82, 按键为菜单键

物理按键的编码是4, 按键为返回键

11-2

11-2App4

“”AppAppApp

“”onKeyDown

```
private boolean bExit = false;
@Override
public boolean onKeyDown(int keyCode, KeyEvent event) {
    if (keyCode == KeyEvent.KEYCODE_BACK) {
        if (bExit) {
            finish();
        }
        bExit = true;
        Toast.makeText(this, "再按一次返回键退出!", Toast.LENGTH_SHORT).show();
        return true;
    } else {
        return super.onKeyDown(keyCode, event);
    }
}
```

ActivityonBackPressed

```
private boolean bExit = false;
@Override
public void onBackPressed() {
    if (bExit) {
        finish();
        return;
    }
    bExit = true;
    Toast.makeText(this, "再按一次返回键退出!", Toast.LENGTH_SHORT).show();
}
```

11-3

A black rectangular button with white text that reads "再按一次返回键退出!". The button has a slight shadow and is centered on a light gray background.

11-3“”

11.1.3

9Android6App

ActivityonKeyDownOnKeyListener

```
public class VolumeDialog implements OnSeekBarChangeListener, OnKeyListener {
    private Dialog dialog;
    private View view;
    private SeekBar sb_music;
    private AudioManager mAudioMgr;
    private int MUSIC = AudioManager.STREAM_MUSIC;
    private int mMaxVolume;
    private int mNowVolume;

    public VolumeDialog(Context context) {
        mAudioMgr = (AudioManager) context.getSystemService(Context.AUDIO_SERVICE);
        mMaxVolume = mAudioMgr.getStreamMaxVolume(MUSIC);
        mNowVolume = mAudioMgr.getStreamVolume(MUSIC);
        view = LayoutInflater.from(context).inflate(R.layout.dialog_volume, null);
        dialog = new Dialog(context, R.style.VolumeDialog);
        sb_music = (SeekBar) view.findViewById(R.id.sb_music);
        sb_music.setOnSeekBarChangeListener(this);
        sb_music.setProgress(sb_music.getMax() * mNowVolume/mMaxVolume);
    }
}
```

```

}

public void show() {
    dialog.getWindow().setContentView(view);
    dialog.getWindow().setLayout(
        LayoutParams.MATCH_PARENT, LayoutParams.WRAP_CONTENT);
    dialog.show();
    sb_music.setFocusable(true);
    sb_music.setFocusableInTouchMode(true);
    sb_music.setOnKeyListener(this);
}

public void dismiss() {
    if (dialog != null && dialog.isShowing()) {
        dialog.dismiss();
    }
}

public boolean isShowing() {
    if (dialog != null) {
        return dialog.isShowing();
    } else {
        return false;
    }
}

public void adjustVolume(int direction, boolean fromActivity) {
    if (direction == AudioManager.ADJUST_RAISE) {
        mNowVolume += 1;
    } else {
        mNowVolume -= 1;
    }
    sb_music.setProgress(sb_music.getMax() * mNowVolume/mMaxVolume);
    mAudioMgr.adjustStreamVolume(MUSIC, direction, AudioManager.FLAG_PLAY_SOUND);
    if (mListener != null && fromActivity!=true) {
        mListener.onVolumeAdjust(mNowVolume);
    }
    close();
}

private void close() {
    mHandler.removeCallbacks(mClose);
    mHandler.postDelayed(mClose, 2000);
}

```

```

    }

    private Handler mHandler = new Handler();
    private Runnable mClose = new Runnable() {
        @Override
        public void run() {
            dismiss();
        }
    };

    @Override
    public void onProgressChanged(SeekBar seekBar, int progress, boolean fromUser) {
    }

    @Override
    public void onStartTrackingTouch(SeekBar seekBar) {
    }

    @Override
    public void onStopTrackingTouch(SeekBar seekBar) {
        mNowVolume = mMaxVolume * seekBar.getProgress()/seekBar.getMax();
        mAudioMgr.setStreamVolume(MUSIC, mNowVolume, AudioManager.FLAG_PLAY_SOUND);
        if (mListener != null) {
            mListener.onVolumeAdjust(mNowVolume);
        }
        close();
    }

    @Override
    public boolean onKeyDown(View v, int keyCode, KeyEvent event) {
        if (keyCode == KeyEvent.KEYCODE_VOLUME_UP && event.getAction() == KeyEvent.
ACTION_DOWN) {
            adjustVolume(AudioManager.ADJUST_RAISE, false);
            return true;
        } else if (keyCode == KeyEvent.KEYCODE_VOLUME_DOWN && event.getAction() ==
KeyEvent.ACTION_DOWN) {
            adjustVolume(AudioManager.ADJUST_LOWER, false);
            return true;
        } else {
            return false;
        }
    }
}

```

```

private VolumeAdjustListener mListener;
public void setVolumeAdjustListener(VolumeAdjustListener listener) {
    mListener = listener;
}

public static interface VolumeAdjustListener {
    public abstract void onVolumeAdjust(int volume);
}
}

```

```

public class VolumeSetActivity extends AppCompatActivity implements VolumeAdjustListener {
    private TextView tv_volume;
    private VolumeDialog dialog;
    private AudioManager mAudioMgr;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_volume_set);
        tv_volume = (TextView) findViewById(R.id.tv_volume);
        mAudioMgr = (AudioManager) getSystemService(Context.AUDIO_SERVICE);
    }

    @Override
    public boolean onKeyDown(int keyCode, KeyEvent event) {
        if (keyCode == KeyEvent.KEYCODE_VOLUME_UP && event.getAction() == KeyEvent.
ACTION_DOWN) {
            showVolumeDialog(AudioManager.ADJUST_RAISE);
            return true;
        } else if (keyCode == KeyEvent.KEYCODE_VOLUME_DOWN && event.getAction() ==
KeyEvent.ACTION_DOWN) {
            showVolumeDialog(AudioManager.ADJUST_LOWER);
            return true;
        } else if (keyCode == KeyEvent.KEYCODE_BACK) {
            finish();
            return false;
        } else {
            return false;
        }
    }

    private void showVolumeDialog(int direction) {

```

```

        if (dialog == null || dialog.isShowing() != true) {
            dialog = new VolumeDialog(this);
            dialog.setVolumeAdjustListener(this);
            dialog.show();
        }
        dialog.adjustVolume(direction, true);
        onVolumeAdjust(mAudioMgr.getStreamVolume(AudioManager.STREAM_MUSIC));
    }

    @Override
    public void onVolumeAdjust(int volume) {
        tv_volume.setText("调节后的音乐音量大小为 : "+volume);
    }
}

```

11-411-511-411-5



11-4



11-5

11.2

33—

11.2.1

```
()Android(OnClickListener)
(OnLongClickListener)ScrollViewViewPagerApp
3()dispatchTouchEventonInterceptTouchEventonTouchEvent
dispatchTouchEvent>trueonInterceptTouchEvent>false
onTouchEvent
onInterceptTouchEvent>trueonTouchEvent>false

onTouchEvent>trueonTouchEvent>false
onTouchEventonTouchEvent
3()
ActivitydispatchTouchEventonTouchEvent
ViewGroupLayoutListViewGridViewSpinnerViewPager
RecyclerViewToolbardispatchTouchEventonInterceptTouchEvent
onTouchEvent
ViewTextViewImageViewButtondispatchTouchEvent
onTouchEvent
onInterceptTouchEventonInterceptTouchEvent

33
dispatchTouchEvent>trueonTouchEvent
dispatchTouchEventonTouchEventonTouchEvent
setOnTouchListenerOnTouchListeneronTouchonTouchEvent
```


dispatchTouchEventonInterceptTouchEvent
dispatchTouchEventonInterceptTouchEvent
onTouchEvent

3

dispatchTouchEvent
onInterceptTouchEvent
onTouchEvent

```
public class NotDispatchLayout extends LinearLayout {
    public NotDispatchLayout(Context context) {
        super(context);
    }

    public NotDispatchLayout(Context context, AttributeSet attrs) {
        super(context, attrs);
    }

    @Override
    public boolean dispatchTouchEvent(MotionEvent ev) {
        if (mListener != null) {
            mListener.onNotDispatch();
        }
        return false; // 一般容器默认返回 true, 即允许分发给下级
    }

    private NotDispatchListener mListener;
    public void setNotDispatchListener(NotDispatchListener listener) {
        mListener = listener;
    }

    public static interface NotDispatchListener {
        public abstract void onNotDispatch();
    }
}
```

onNotDispatch

```
public void onNotDispatch() {  
    desc_no = String.format("%s%s 触摸动作未分发，按钮点击不了了\n"  
        , desc_no, DateUtil.getNowTime());  
    tv_dispatch_no.setText(desc_no);  
}
```

11-611-711-611-7onNotDispatch



event

这里允许分发给下级

13:57:29 您点击了按钮

这里不允许发给下级

13:58:02 触摸动作未分发，按钮点击不了了

```

public class InterceptLayout extends LinearLayout {
    public InterceptLayout(Context context) {
        super(context);
    }

    public InterceptLayout(Context context, AttributeSet attrs) {
        super(context, attrs);
    }

    @Override
    public boolean onInterceptTouchEvent(MotionEvent ev) {
        if (mListener != null) {
            mListener.onIntercept();
        }
        return true; // 一般容器默认返回 false（不拦截），不过滚动视图 ScrollView 会拦截
    }

    private InterceptListener mListener;
    public void setInterceptListener(InterceptListener listener) {
        mListener = listener;
    }

    public static interface InterceptListener {
        public abstract void onIntercept();
    }
}

```

onIntercept

```

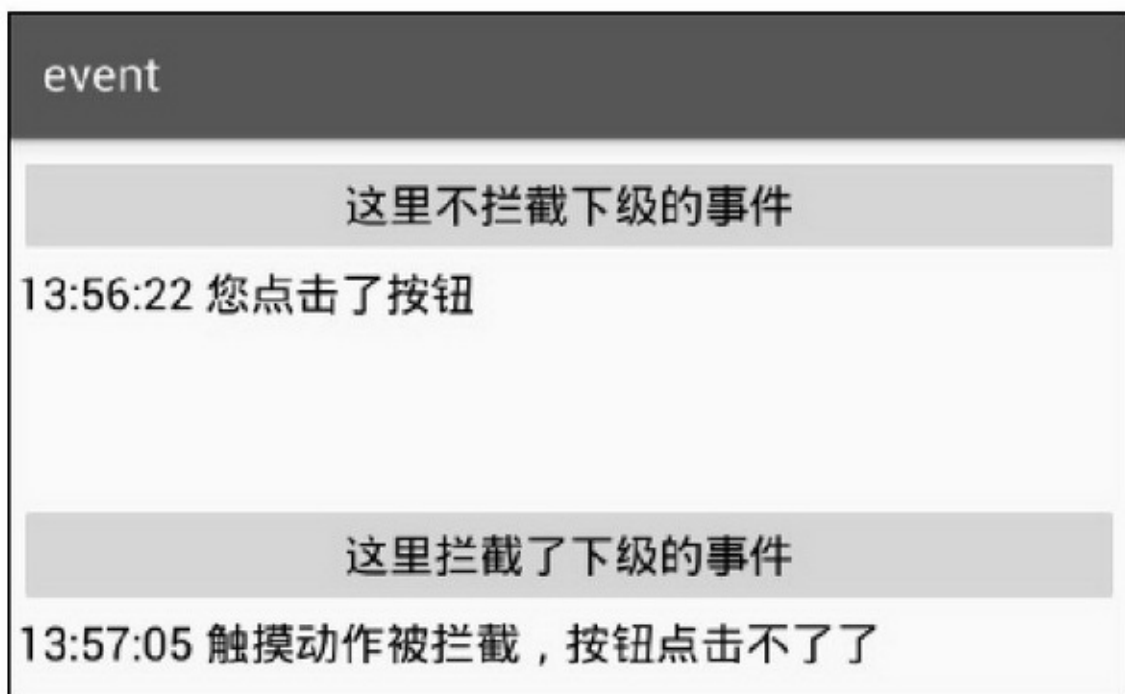
public void onIntercept() {
    desc_yes = String.format("%s%s 触摸动作被拦截，按钮点击不了了\n", desc_yes,
        DateUtil.getNowTime());
    tv_intercept_yes.setText(desc_yes);
}

```

11-811-911-811-9onIntercept



11-8



11-9

11.2.2 MotionEvent dispatchTouchEvent onInterceptTouchEvent onTouchEvent MotionEvent MotionEvent

MotionEvent

getAction11-2

11-2

MotionEvent 类的动作类型	说明
ACTION_DOWN	按下动作
ACTION_UP	提起动作
ACTION_MOVE	移动动作
ACTION_CANCEL	取消动作
ACTION_OUTSIDE	移出边界动作
ACTION_POINTER_DOWN	第二个点的按下动作，用于多点触控的判断
ACTION_POINTER_UP	第二个点的提起动作，用于多点触控的判断
ACTION_MASK	动作掩码，与原动作类型进行“与”(&)操作后获得多点触控信息

getTime()

getX

getY

getRawX

getRawY

getPressure

getPointerCount

```

public class TouchSingleActivity extends AppCompatActivity {
    private TextView tv_touch;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_touch_single);
        tv_touch = (TextView) findViewById(R.id.tv_touch);
    }

    @Override
    public boolean onTouchEvent(MotionEvent event) {
        int seconds = (int) (event.getTime() / 1000); // 从开机到现在的毫秒数
        int hour = seconds / 3600;
        int minute = seconds % 3600 / 60;
        int second = seconds % 60;
        Date date = new Date(event.getTime());
        String desc = String.format("动作发生时间 : 开机距离现在%02d:%02d:%02d",
            hour, minute, second);
        desc = String.format("%s\n 动作名称是 : ", desc);
        int action = event.getAction();
        if (action == event.ACTION_DOWN) {
            desc = String.format("%s 按下", desc);
        } else if (action == event.ACTION_MOVE) {
            desc = String.format("%s 移动", desc);
        } else if (action == event.ACTION_UP) {
            desc = String.format("%s 提起", desc);
        } else if (action == event.ACTION_CANCEL) {
            desc = String.format("%s 取消", desc);
        }
        desc = String.format("%s\n 动作发生位置是 : 横坐标%f, 纵坐标%f",
            desc, event.getX(), event.getY());
        tv_touch.setText(desc);
        return super.onTouchEvent(event);
    }
}

```

11-1011-1111-1211-1011-1111-12

event

动作发生时间：开机距离现在00:23:52
动作名称是：按下
动作发生位置是：横坐标279.514740，纵坐标698.318054

11-10

event

动作发生时间：开机距离现在00:25:08
动作名称是：移动
动作发生位置是：横坐标299.949188，纵坐标472.538544

11-11

event

动作发生时间：开机距离现在00:25:11
动作名称是：提起
动作发生位置是：横坐标299.949188，纵坐标472.538544

11-12


```
public class TouchMultipleActivity extends AppCompatActivity {
    private TextView tv_touch_main;
    private TextView tv_touch_secondary;
    private boolean bSecondaryPressed = false;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_touch_multiple);
        tv_touch_main = (TextView) findViewById(R.id.tv_touch_main);
        tv_touch_secondary = (TextView) findViewById(R.id.tv_touch_secondary);
    }

    @Override
    public boolean onTouchEvent(MotionEvent event) {
        int seconds = (int) (event.getTime() / 1000);
        int hour = seconds / 3600;
        int minute = seconds % 3600 / 60;
        int second = seconds % 60;
        Date date = new Date(event.getTime());
        String desc_main = String.format("主要动作发生时间 : 开机距离现在%02d:%02d:%02d\n%s",
            hour, minute, second, "主要动作名称是 : ");
        String desc_secondary = "";
        int action = event.getAction() & MotionEvent.ACTION_MASK;
        if (action == event.ACTION_DOWN) {
            desc_main = String.format("%s 按下", desc_main);
        } else if (action == event.ACTION_MOVE) {
            desc_main = String.format("%s 移动", desc_main);
            if (bSecondaryPressed == true) {
                desc_secondary = String.format("%s 次要动作名称是 : 移动", desc_secondary);
            }
        }
    }
}
```

```

    } else if (action == event.ACTION_UP) {
        desc_main = String.format("%s 提起", desc_main);
    } else if (action == event.ACTION_CANCEL) {
        desc_main = String.format("%s 取消", desc_main);
    } else if (action == event.ACTION_POINTER_DOWN) {
        bSecondaryPressed = true;
        desc_secondary = String.format("%s 次要动作名称是 : 按下", desc_secondary);
    } else if (action == event.ACTION_POINTER_UP) {
        bSecondaryPressed = false;
        desc_secondary = String.format("%s 次要动作名称是 : 提起", desc_secondary);
    }
    desc_main = String.format("%s\n 主要动作发生位置是 : 横坐标%f, 纵坐标%f",
        desc_main, event.getX(), event.getY());
    tv_touch_main.setText(desc_main);
    if (bSecondaryPressed == true || desc_secondary.length() > 0) {
        desc_secondary = String.format("%s\n 次要动作发生位置是 : 横坐标%f, 纵坐标%f",
            desc_secondary, event.getX(1), event.getY(1));
        tv_touch_secondary.setText(desc_secondary);
    }
    return super.onTouchEvent(event);
}
}
}

```

11-1311-1411-1311-14

event

主要动作发生时间: 开机距离现在39:40:01

主要动作名称是: 按下

主要动作发生位置是: 横坐标447.378662, 纵坐标373.708038

次要动作名称是: 按下

次要动作发生位置是: 横坐标203.717072, 纵坐标635.503540

event

主要动作发生时间：开机距离现在39:41:41

主要动作名称是：提起

主要动作发生位置是：横坐标460.360626，纵坐标497.611237

次要动作名称是：提起

次要动作发生位置是：横坐标220.693481，纵坐标766.401245

11.2.3

PathPathmoveToPathquadToCanvas
drawPath

```

@Override
protected void onDraw(Canvas canvas) {
    super.onDraw(canvas);
    canvas.drawBitmap(cacheBitmap, 0, 0, null);
    canvas.drawPath(path, paint); //这个是需要的，最近一次路径保存在这里
}

private float mLastX, mLastY;
@Override
public boolean onTouchEvent(MotionEvent event) {
    switch (event.getAction()) {
        case MotionEvent.ACTION_DOWN:
            path.moveTo(event.getX(), event.getY());
            pos.firstX = event.getX();
            pos.firstY = event.getY();
            break;
        case MotionEvent.ACTION_MOVE:
            path.quadTo(mLastX, mLastY, event.getX(), event.getY());
            pos.nextX = event.getX();
            pos.nextY = event.getY();
            pathArray.add(pos);
            pos = new PathPosition();
            pos.firstX = event.getX();
            pos.firstY = event.getY();
            break;
        case MotionEvent.ACTION_UP:
            cacheCanvas.drawPath(path, paint);
            path.reset();
            break;
    }
    mLastX = event.getX();
    mLastY = event.getY();
    invalidate();
    return true;
}

```

11-1511-1611-1511-16



11-15



11-16

11.3

11.3.1 GestureDetector Android GestureDetector

GesturDetectorGesturDetector

- onDown
- onShowPress
- onSingleTapUp0.5
- onScroll
- onLongPress0.5
- onFling

OnGestureListener

truefalse

onTouchEvent

OnGestureListener

```

public class GestureDetectorActivity extends AppCompatActivity {
    private TextView tv_gesture;
    private GestureDetector mGesture;
    private String desc = "";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_gesture_detector);
        tv_gesture = (TextView) findViewById(R.id.tv_gesture);
        mGesture = new GestureDetector(this, new MyGestureListener());
    }

    public boolean dispatchTouchEvent(MotionEvent event) {
        mGesture.onTouchEvent(event);
        return true;
    }

    final class MyGestureListener implements GestureDetector.OnGestureListener {
        @Override
        public final boolean onDown(MotionEvent event) {
            return true; //onDown 的返回值没有作用，不影响其他手势的处理
        }

        @Override
        public final boolean onFling(MotionEvent e1, MotionEvent e2, float velocityX, float velocityY) {
            float offsetX = e1.getX() - e2.getX();
            float offsetY = e1.getY() - e2.getY();
            if (Math.abs(offsetX) > Math.abs(offsetY)) {
                if (offsetX > 0) {
                    desc = String.format("%s%s 您向左滑动了一下\n", desc, DateUtil.getNowTime());
                } else {
                    desc = String.format("%s%s 您向右滑动了一下\n", desc, DateUtil.getNowTime());
                }
            } else {
                if (offsetY > 0) {
                    desc = String.format("%s%s 您向上滑动了一下\n", desc, DateUtil.getNowTime());
                } else {
                    desc = String.format("%s%s 您向下滑动了一下\n", desc, DateUtil.getNowTime());
                }
            }
        }
    }
}

```

```

        tv_gesture.setText(desc);
        return true;
    }

    @Override
    public final void onLongPress(MotionEvent event) {
        desc = String.format("%s%s 您长按了一下下\n", desc, DateUtil.getNowTime());
        tv_gesture.setText(desc);
    }

    @Override
    public final boolean onScroll(MotionEvent e1, MotionEvent e2, float distanceX, float distanceY) {
        return false;
    }

    @Override
    public final void onShowPress(MotionEvent event) {
    }

    @Override
    public boolean onSingleTapUp(MotionEvent event) {
        desc = String.format("%s%s 您轻轻点了一下\n", desc, DateUtil.getNowTime());
        tv_gesture.setText(desc);
        return true; //返回 true 表示已经处理了，其他地方不要再处理这个手势
    }
}
}
}

```

11-17

event

14:11:26 您向左滑动了一下
 14:11:29 您向右滑动了一下
 14:11:31 您向上滑动了一下
 14:11:33 您向下滑动了一下
 14:11:38 您轻轻点了一下
 14:11:41 您长按了一下下

11.3.2 ViewFlipper Android ViewFlipper

ViewPager ViewFlipper() ViewPager

ViewFlipper

setFlipInterval

setAutoStart true

startFlipping

stopFlipping

isFlipping

showNext

showPrevious

setDisplayedChild

getDisplayedChild

setInAnimation

getInAnimation

setOutAnimation

getOutAnimation

ViewFlipper

```
public class ViewFlipperActivity extends AppCompatActivity implements OnClickListener {
    private Button btn_control_flipper;
    private RelativeLayout rl_content;
    private ViewFlipper vf_content;
    private RadioGroup rg_indicator;
    private int dip_15;
    private boolean bPlay = true;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_view_flipper);
        btn_control_flipper = (Button) findViewById(R.id.btn_control_flipper);
        rl_content = (RelativeLayout) findViewById(R.id.rl_content);
        vf_content = (ViewFlipper) findViewById(R.id.vf_content);
        rg_indicator = (RadioGroup) findViewById(R.id.rg_indicator);
        btn_control_flipper.setOnClickListener(this);
        findViewById(R.id.btn_pre_flipper).setOnClickListener(this);
        findViewById(R.id.btn_next_flipper).setOnClickListener(this);
        initFlipper();
    }

    private void initFlipper() {
```

```

LayoutParams params = (LayoutParams) rl_content.getLayoutParams();
params.height = (int) (DisplayUtil.getScreenWidth(this) * 250f / 640f);
rl_content.setLayoutParams(params);
ArrayList<Integer> imageList = new ArrayList<Integer>();
imageList.add(Integer.valueOf(R.drawable.banner_1));
imageList.add(Integer.valueOf(R.drawable.banner_2));
imageList.add(Integer.valueOf(R.drawable.banner_3));
imageList.add(Integer.valueOf(R.drawable.banner_4));
imageList.add(Integer.valueOf(R.drawable.banner_5));
for (int i = 0; i < imageList.size(); i++) {
    Integer imageID = ((Integer) imageList.get(i)).intValue();
    ImageView iv_item = new ImageView(this);
    iv_item.setLayoutParams(new LayoutParams(
        LayoutParams.MATCH_PARENT, LayoutParams.MATCH_PARENT));
    iv_item.setScaleType(ImageView.ScaleType.FIT_XY);
    iv_item.setImageResource(imageID);
    vf_content.addView(iv_item);
}
dip_15 = Utils.dip2px(this, 15);
for (int i = 0; i < imageList.size(); i++) {
    RadioButton radio = new RadioButton(this);
    radio.setLayoutParams(new RadioGroup.LayoutParams(dip_15, dip_15));
    radio.setGravity(Gravity.CENTER);
    radio.setButtonDrawable(R.drawable.indicator_selector);
    rg_indicator.addView(radio);
}
vf_content.setDisplayedChild(0);
vf_content.setAutoStart(true);
mHandler.postDelayed(mRefresh, 200);
}

@Override
public void onClick(View v) {
    if (v.getId() == R.id.btn_pre_flipper) {
        vf_content.showPrevious();
    } else if (v.getId() == R.id.btn_next_flipper) {
        vf_content.showNext();
    } else if (v.getId() == R.id.btn_control_flipper) {
        bPlay = !bPlay;
        if (bPlay == true) {
            vf_content.startFlipping();
            btn_control_flipper.setText("停止自动翻页");
        } else {

```

```
        vf_content.stopFlipping();
        btn_control_flipper.setText("开始自动翻页");
    }
}

private Handler mHandler = new Handler();
private Runnable mRefresh = new Runnable() {
    @Override
    public void run() {
        int pos = vf_content.getDisplayedChild();
        ((RadioButton) rg_indicator.getChildAt(pos)).setChecked(true);
        mHandler.postDelayed(this, 200);
    }
};
}
```

11-1811-1911-1811-194setFlipInterval
flipInterval





11-194

11.3.3

- 步骤 01 `dispatchTouchEvent`
- 步骤 02 `onFlingvelocityX`
- 步骤 03

```

public class BannerFlipper extends RelativeLayout {
    private static final String TAG = "BannerFlipper";
    private Context mContext;
    private ViewFlipper mFlipper;
    private RadioGroup mGroup;
    private LayoutInflater mInflater;
    private int dip_15;
    private GestureDetector mGesture;
    private float mFlipGap = 20f;

    public BannerFlipper(Context context) {
        this(context, null);
    }

    public BannerFlipper(Context context, AttributeSet attrs) {
        super(context, attrs);
        mContext = context;
        init();
    }

    public void setImage(ArrayList<Integer> imageList) {
        for (int i = 0; i < imageList.size(); i++) {
            Integer imageID = ((Integer) imageList.get(i)).intValue();
            ImageView iv_item = new ImageView(mContext);
            iv_item.setLayoutParams(new LayoutParams(
                LayoutParams.MATCH_PARENT, LayoutParams.MATCH_PARENT));
            iv_item.setScaleType(ImageView.ScaleType.FIT_XY);
            iv_item.setImageResource(imageID);
            mFlipper.addView(iv_item);
        }
        for (int i = 0; i < imageList.size(); i++) {
            RadioButton radio = new RadioButton(mContext);
            radio.setLayoutParams(new RadioGroup.LayoutParams(dip_15, dip_15));
            radio.setGravity(Gravity.CENTER);
            radio.setButtonDrawable(R.drawable.indicator_selector);
            mGroup.addView(radio);
        }
        mFlipper.setDisplayedChild(imageList.size() - 1);
        startFlip();
    }
}

```

```
private void init() {
    mInflater = ((Activity) mContext).getLayoutInflater();
    View view = mInflater.inflate(R.layout.banner_flipper, null);
    mFlipper = (ViewFlipper) view.findViewById(R.id.banner_flipper);
    mGroup = (RadioGroup) view.findViewById(R.id.rg_indicator);
    addView(view);
    dip_15 = Utils.dip2px(mContext, 15);
    // 该手势的 onSingleTapUp 事件是点击时进入广告页
    mGesture = new GestureDetector(mContext, new BannerGestureListener());
    mHandler.postDelayed(mRefresh, 200);
}

public boolean dispatchTouchEvent(MotionEvent event) {
    mGesture.onTouchEvent(event);
    return true;
}

final class BannerGestureListener implements GestureDetector.OnGestureListener {
    @Override
    public final boolean onDown(MotionEvent event) {
        return true;
    }

    @Override
    public final boolean onFling(MotionEvent e1, MotionEvent e2, float velocityX, float velocityY) {
        if (e1.getX() - e2.getX() > mFlipGap) {
            startFlip();
            return true;
        }
        if (e1.getX() - e2.getX() < -mFlipGap) {
            backFlip();
            return true;
        }
        return false;
    }

    @Override
    public final void onLongPress(MotionEvent event) {
    }

    @Override
    public final boolean onScroll(MotionEvent e1, MotionEvent e2, float distanceX, float distanceY) {
        return false;
    }
}
```

```

    }

    @Override
    public final void onShowPress(MotionEvent event) {
    }

    @Override
    public boolean onSingleTapUp(MotionEvent event) {
        int position = mFlipper.getDisplayedChild();
        mListener.onBannerClick(position);
        return true;
    }
}

private void startFlip() {
    mFlipper.startFlipping();
    mFlipper.showNext();
}

private void backFlip() {
    mFlipper.startFlipping();
    mFlipper.showPrevious();
}

private Handler mHandler = new Handler();
private Runnable mRefresh = new Runnable() {
    @Override
    public void run() {
        int pos = mFlipper.getDisplayedChild();
        ((RadioButton) mGroup.getChildAt(pos)).setChecked(true);
        mHandler.postDelayed(this, 200);
    }
};

private BannerClickListener mListener;
public void setOnBannerListener(BannerClickListener listener) {
    mListener = listener;
}

public static interface BannerClickListener {
    public abstract void onBannerClick(int position);
}
}

```

11-2011-2111-2011-21



11-20



11-21

11.4

11.4.1

AndroidScrollViewViewPagerAB

AppScrollView11-22BannerBanner11-23





11-23

BannerBannerScrollViewScrollViewBanner
ScrollViewScrollView
ScrollView
1.
onInterceptTouchEventonInterceptTouchEvent

```

public class CustomScrollView extends ScrollView {
    private float mOffsetX, mOffsetY;
    private float mLastPosX, mLastPosY;

    public CustomScrollView(Context context) {
        this(context, null);
    }

    public CustomScrollView(Context context, AttributeSet attr) {
        super(context, attr);
    }

    @Override
    public boolean onInterceptTouchEvent(MotionEvent event) {
        boolean result = false;
        switch (event.getAction()) {
            case MotionEvent.ACTION_DOWN:
                mOffsetX = 0.0F;
                mOffsetY = 0.0F;
                mLastPosX = event.getX();
                mLastPosY = event.getY();
                result = super.onInterceptTouchEvent(event); // false 传给子控件
                break;
            default:

```

```

                float thisPosX = event.getX();
                float thisPosY = event.getY();
                mOffsetX += Math.abs(thisPosX - mLastPosX); // x 轴偏差
                mOffsetY += Math.abs(thisPosY - mLastPosY); // y 轴偏差
                mLastPosX = thisPosX;
                mLastPosY = thisPosY;
                if (mOffsetX < 3 && mOffsetY < 3) {
                    result = false; // false 传给子控件 (点击事件)
                } else if (mOffsetX < mOffsetY) {
                    result = true; // true 不传给子控件 (垂直滑动)
                } else {
                    result = false; // false 传给子控件
                }
                break;
        }
        return result;
    }
}

```



11-24

XMLScrollView(com.example.event.widget.CustomScrollView)App

11-24

2.

ScrollViewBannerScrollView

requestDisallowInterceptTouchEvent true Banner
onScroll

```
public final boolean onScroll(MotionEvent e1, MotionEvent e2, float distanceX, float distanceY){
    // 如果外层是普通的 ScrollView, 此处就不允许父容器的拦截动作
    if (Math.abs(distanceY) < Math.abs(distanceX)) {
        BannerFlipper.this.getParent().requestDisallowInterceptTouchEvent(true);
        return true;
    } else {
        return false;
    }
}
```

11-24

11.4.2

ScrollViewViewPagerFragmentScrollView
ScrollViewViewPager
ViewPagerAndroidDrawerLayout
DrawerLayout
setDrawerShadow
▶ onDrawerSlide
▶ onDrawerOpened
▶ onDrawerClosed
▶ onDrawerStateChanged
addDrawerListenerDrawerListener4
removeDrawerListener
closeDrawers
openDrawer
closeDrawer
isDrawerOpen
layout_gravityleftlayout_gravityright
DrawerLayout

```
<android.support.v4.widget.DrawerLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/dl_layout"
    android:layout_width="match_parent"
    android:layout_height="match_parent" >

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical" >

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal" >

            <Button
                android:id="@+id/btn_drawer_left"
                android:layout_width="0dp"
                android:layout_height="wrap_content"
                android:layout_weight="1"
                android:gravity="center"
                android:text="打开左侧侧滑"
                android:textColor="@color/black"
                android:textSize="20sp" />

            <Button
                android:id="@+id/btn_drawer_right"
                android:layout_width="0dp"
                android:layout_height="wrap_content"
                android:layout_weight="1"
                android:gravity="center"
                android:text="打开右侧侧滑"
                android:textColor="@color/black"
                android:textSize="20sp" />

        </LinearLayout>

        <TextView
            android:id="@+id/tv_drawer_center"
            android:layout_width="match_parent"
            android:layout_height="0dp"
            android:layout_weight="1"
            android:gravity="top|center"
            android:paddingTop="30dp"
```

```

        android:text="这里是首页"
        android:textColor="@color/black"
        android:textSize="20sp" />
</LinearLayout>

<ListView
    android:id="@+id/lv_drawer_left"
    android:layout_width="150dp"
    android:layout_height="match_parent"
    android:layout_gravity="left"
    android:background="#ffdd99" />

<ListView
    android:id="@+id/lv_drawer_right"
    android:layout_width="150dp"
    android:layout_height="match_parent"
    android:layout_gravity="right"
    android:background="#99ffdd" />
</android.support.v4.widget.DrawerLayout>

```

```

public class DrawerLayoutActivity extends AppCompatActivity implements OnClickListener {
    private final static String TAG = "DrawerLayoutActivity";
    private DrawerLayout dl_layout;
    private Button btn_drawer_left;
    private Button btn_drawer_right;
    private TextView tv_drawer_center;
    private ListView lv_drawer_left;
    private ListView lv_drawer_right;
    private String[] titleArray = { "首页", "新闻", "娱乐", "博客", "论坛" };
    private String[] settingArray = { "我的", "设置", "关于" };

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_drawer_layout);
        dl_layout = (DrawerLayout) findViewById(R.id.dl_layout);
        dl_layout.addDrawerListener(new SlidingListener());
        btn_drawer_left = (Button) findViewById(R.id.btn_drawer_left);
        btn_drawer_right = (Button) findViewById(R.id.btn_drawer_right);
        tv_drawer_center = (TextView) findViewById(R.id.tv_drawer_center);
        btn_drawer_left.setOnClickListener(this);
        btn_drawer_right.setOnClickListener(this);
    }
}

```



```

lv_drawer_left = (ListView) findViewById(R.id.lv_drawer_left);
ArrayAdapter<String> left_adapter = new ArrayAdapter<String>(this,
    R.layout.item_select, titleArray);
lv_drawer_left.setAdapter(left_adapter);
lv_drawer_left.setOnItemClickListener(new LeftClickListener());
lv_drawer_right = (ListView) findViewById(R.id.lv_drawer_right);
ArrayAdapter<String> right_adapter = new ArrayAdapter<String>(this,
    R.layout.item_select, settingArray);
lv_drawer_right.setAdapter(right_adapter);
lv_drawer_right.setOnItemClickListener(new RightClickListener());
}

@Override
public void onClick(View v) {
    if (v.getId() == R.id.btn_drawer_left) {
        if (dl_layout.isDrawerOpen(lv_drawer_left)) {
            dl_layout.closeDrawer(lv_drawer_left);
        } else {
            dl_layout.openDrawer(lv_drawer_left);
        }
    } else if (v.getId() == R.id.btn_drawer_right) {
        if (dl_layout.isDrawerOpen(lv_drawer_right)) {
            dl_layout.closeDrawer(lv_drawer_right);
        } else {
            dl_layout.openDrawer(lv_drawer_right);
        }
    }
}

private class LeftClickListener implements OnItemClickListener {
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
        String text = titleArray[position];
        tv_drawer_center.setText("这里是" + text + "页面");
        dl_layout.closeDrawers();
    }
}

private class RightClickListener implements OnItemClickListener {
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
        String text = settingArray[position];

```

```
        tv_drawer_center.setText("这里是" + text + "页面");
        dl_layout.closeDrawers();
    }
}

private class SlidingListener implements DrawerListener {
    @Override
    public void onDrawerSlide(View paramView, float paramFloat) {
    }

    @Override
    public void onDrawerOpened(View paramView) {
        if (paramView.getId() == R.id.lv_drawer_left) {
            btn_drawer_left.setText("关闭左侧侧滑");
        } else {
            btn_drawer_right.setText("关闭右侧侧滑");
        }
    }

    @Override
    public void onDrawerClosed(View paramView) {
        if (paramView.getId() == R.id.lv_drawer_left) {
            btn_drawer_left.setText("打开左侧侧滑");
        } else {
            btn_drawer_right.setText("打开右侧侧滑");
        }
    }

    @Override
    public void onDrawerStateChanged(int paramInt) {
    }
}
}
```

11-2511-2611-2711-2511-2611-27



11-25



11-26

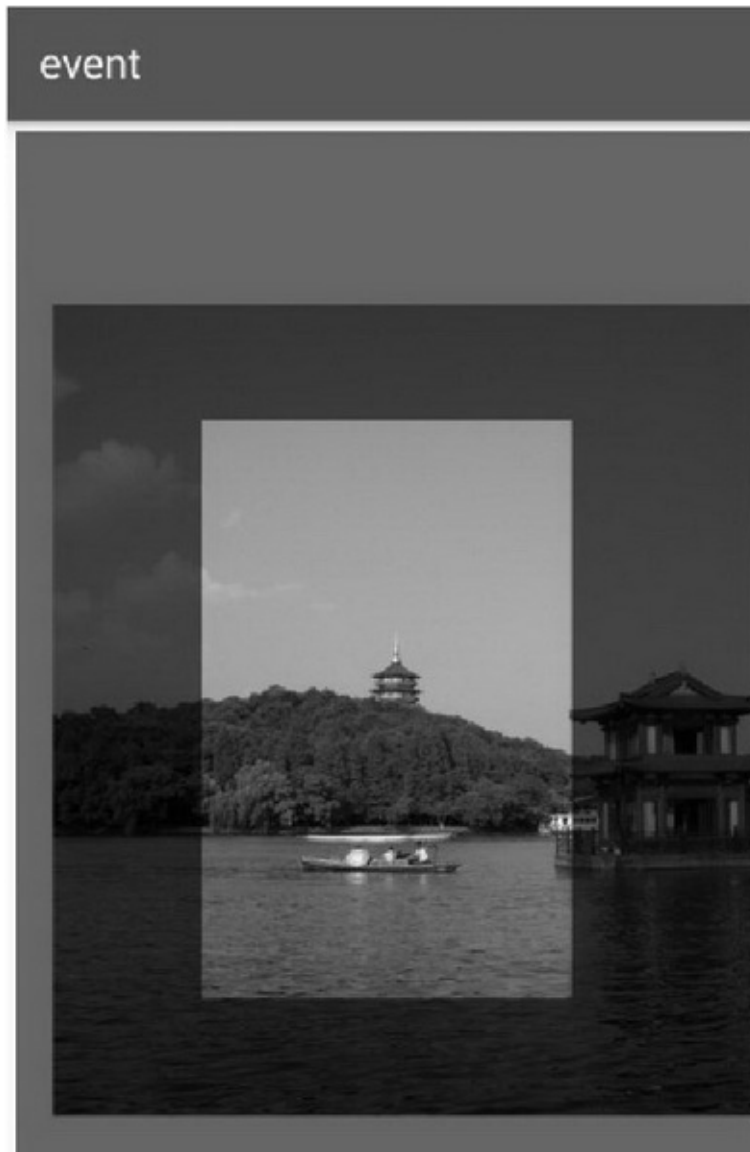


11-27

11.5

OPPOvivo

11.5.1



11-28

6

0.5

0

4

11.5.2

AndroidDrawableBitmapDrawableBitmap

BitmapDrawable

BitmapDrawable

Drawable drawable = new BitmapDrawable(getResources(), bitmap);

DrawableBitmap

Bitmap bitmap = ((BitmapDrawable)drawable).getBitmap(); Bitmap

createBitmap

createScaledBitmap

compress

recycle

getByteCount

getWidth

getHeight

(1)BitmapcreateBitmap

(2)CanvasdrawBitmap

(3)BitmapcreateScaledBitmap

(4)MatrixMatrixpostRotatecreateBitmap

```
public static Bitmap getRotateBitmap(Bitmap b, float rotateDegree) {  
    Matrix matrix = new Matrix();  
    matrix.postRotate((float) rotateDegree);  
    Bitmap rotaBitmap = Bitmap.createBitmap(b, 0, 0, b.getWidth(), b.getHeight(), matrix, false);  
    return rotaBitmap;  
}
```

11-2911-3011-3111-2911-3011-3190



event

打开图片文件

保存图片文件

缩放比率: 2.0 ▾ 旋转角度: 0 ▾





11-3190

11.5.3

3

(1)AndroidManifest.xml

```
<!-- SD 卡 -->  
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />  
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />  
<uses-permission android:name="android.permission.MOUNT_UNMOUNT_FILESYSTEMS" />
```

(2)res/menumeu_meitu.xml

(3)



11-32

11-32

“11-3311-3411-3311-34

11-35

event



11-33

event



11-34



11-35

“”

```
public class MeituView extends View {
    private Context mContext;
    private Paint mPaintShade;

    public MeituView(Context context) {
        this(context, null);
    }

    public MeituView(Context context, AttributeSet attrs) {
        super(context, attrs);
        mContext = context;
        mPaintShade = new Paint();
        mPaintShade.setStyle(Style.FILL);
        mPaintShade.setColor(0x99000000);
    }

    private Bitmap mOrigBitmap = null;
    private Bitmap mCropBitmap = null;
    private Rect mRect = new Rect(0,0,0,0);
    public void setOrigBitmap(Bitmap orig) {
        mOrigBitmap = orig;
    }

    public Bitmap getCropBitmap() {
        return mCropBitmap;
    }

    public boolean setBitmapRect(Rect rect) {
```

```

        if (mOrigBitmap == null) {
            return false;
        } else if (rect.left < 0 || rect.left > mOrigBitmap.getWidth()) {
            return false;
        } else if (rect.top < 0 || rect.top > mOrigBitmap.getHeight()) {
            return false;
        } else if (rect.right <= 0 || rect.left + rect.right > mOrigBitmap.getWidth()) {
            return false;
        } else if (rect.bottom <= 0 || rect.top + rect.bottom > mOrigBitmap.getHeight()) {
            return false;
        }
        mRect = rect;
        mCropBitmap = Bitmap.createBitmap(mOrigBitmap,
            mRect.left, mRect.top, mRect.right, mRect.bottom);
        postInvalidate();
        return true;
    }

    public Rect getBitmapRect() {
        return mRect;
    }

    private boolean bReset = false;
    private float mLastOffsetX, mLastOffsetY;
    private float mLastOffsetXTwo, mLastOffsetYTwo;
    private long mOriginTime;

    @Override
    protected void dispatchDraw(Canvas canvas) {
        super.dispatchDraw(canvas);
        if (mOrigBitmap == null) {
            return;
        }
        Rect rectShade = new Rect(0, 0, getMeasuredWidth(), getMeasuredHeight());
        canvas.drawRect(rectShade, mPaintShade); // 画外圈阴影
        canvas.drawBitmap(mCropBitmap, mRect.left, mRect.top, new Paint()); // 画高亮处的图像
    }

    @Override
    public boolean onTouchEvent(MotionEvent event) {
        int action = event.getAction() & MotionEvent.ACTION_MASK;
        switch (action) {
            case MotionEvent.ACTION_DOWN:
                mOriginTime = event.getTime();
                mOriginX = event.getX();

```

```

mOriginY = event.getY();
mOriginRect = mRect;
mDragMode = getDragMode(mOriginX, mOriginY);
bReset = true;
break;
case MotionEvent.ACTION_UP:
    if (mListener != null && Math.abs(event.getX()-mOriginX)<10 &&
        Math.abs(event.getY()-mOriginY)<10) {
        if (event.getEventTime() - mOriginTime < 500) { // 判断点击还是长按
            mListener.onImageClick();
        } else {
            mListener.onImageLongClick();
        }
    }
    break;
case MotionEvent.ACTION_POINTER_DOWN:
    mDragMode = IMAGE_SCALE_OR_ROTATE; // 另需判断缩放还是旋转
    bReset = true;
    break;
case MotionEvent.ACTION_POINTER_UP:
    mDragMode = DRAG_NONE;
    break;
case MotionEvent.ACTION_MOVE:
    int offsetX = (int) (event.getX()-mOriginX);
    int offsetY = (int) (event.getY()-mOriginY);
    Rect rect = null;
    int left = mOriginRect.left;
    int top = mOriginRect.top;
    int right = mOriginRect.right;
    int bottom = mOriginRect.bottom;
    if (mDragMode == DRAG_NONE) {
        return true;
    } else if (mDragMode == DRAG_WHOLE) {
        rect = new Rect(left+offsetX, top+offsetY, right, bottom);
    } else if (mDragMode == DRAG_LEFT) {
        rect = new Rect(left+offsetX, top, right-offsetX, bottom);
    } else if (mDragMode == DRAG_RIGHT) {
        rect = new Rect(left, top, right+offsetX, bottom);
    } else if (mDragMode == DRAG_TOP) {
        rect = new Rect(left, top+offsetY, right, bottom-offsetY);
    } else if (mDragMode == DRAG_BOTTOM) {
        rect = new Rect(left, top, right, bottom+offsetY);
    } else if (mDragMode == DRAG_LEFT_TOP) {

```



```

        rect = new Rect(left+offsetX, top+offsetY, right-offsetX, bottom-offsetY);
    } else if (mDragMode == DRAG_RIGHT_TOP) {
        rect = new Rect(left, top+offsetY, right+offsetX, bottom-offsetY);
    } else if (mDragMode == DRAG_LEFT_BOTTOM) {
        rect = new Rect(left+offsetX, top, right-offsetX, bottom+offsetY);
    } else if (mDragMode == DRAG_RIGHT_BOTTOM) {
        rect = new Rect(left, top, right+offsetX, bottom+offsetY);
    } else if (mDragMode == IMAGE_TRANSLATE) {
        if (mListener != null) {
            mListener.onImageTraslate(offsetX, offsetY, bReset);
            bReset = false;
        }
    } else if (mDragMode == IMAGE_SCALE_OR_ROTATE) {
        if (mListener != null) {
            float nowWholeDistance = distance(event.getX(), event.getY(),
                event.getX(1), event.getY(1));
            float preWholeDistance = distance(mLastOffsetX, mLastOffsetY,
                mLastOffsetXTwo, mLastOffsetYTwo);
            float primaryDistance = distance(event.getX(), event.getY(),
                mLastOffsetX, mLastOffsetY);
            float secondaryDistance = distance(event.getX(1), event.getY(1),
                mLastOffsetXTwo, mLastOffsetYTwo);
            if (Math.abs(nowWholeDistance-preWholeDistance) >
                (float) Math.sqrt(2)/2.0*(primaryDistance+secondaryDistance)) { //缩放
                mListener.onImageScale(nowWholeDistance / preWholeDistance);
            } else { // 旋转
                int preDegree = degree(mLastOffsetX, mLastOffsetY,
                    mLastOffsetXTwo, mLastOffsetYTwo);
                int nowDegree = degree(event.getX(), event.getY(),
                    event.getX(1), event.getY(1));
                mListener.onImageRotate(nowDegree - preDegree);
            }
        }
    }
}
if (mDragMode!=IMAGE_TRANSLATE && mDragMode!=IMAGE_SCALE_OR_ROTATE) {
    setBitmapRect(rect);
}
break;
default:
    break;
}
mLastOffsetX = event.getX();
mLastOffsetY = event.getY();

```

```

        if (event.getPointerCount() >= 2) {
            mLastOffsetXTwo = event.getX(1);
            mLastOffsetYTwo = event.getY(1);
        }
        return true;
    }

    private float distance(float x1, float y1, float x2, float y2) {
        float offsetX = x2 - x1;
        float offsetY = y2 - y1;
        return (float) Math.sqrt(offsetX*offsetX + offsetY*offsetY);
    }

    private int degree(float x1, float y1, float x2, float y2) {
        return (int) (Math.atan((y2-y1) / (x2-x1)) / Math.PI * 180);
    }

    private int DRAG_NONE = 0, DRAG_WHOLE = 1, DRAG_LEFT = 2, DRAG_RIGHT = 3;
    private int DRAG_TOP = 4, DRAG_BOTTOM = 5, DRAG_LEFT_TOP = 6, DRAG_RIGHT_TOP = 7;
    private int DRAG_LEFT_BOTTOM = 8, DRAG_RIGHT_BOTTOM = 9;
    private int IMAGE_TRANSLATE = 10, IMAGE_SCALE_OR_ROTATE = 11;
    private int mDragMode = DRAG_NONE;
    private int mInterval = 15;
    private float mOriginX, mOriginY;
    private Rect mOriginRect;

    private int getDragMode(float f, float g) {
        int left = mRect.left;
        int top = mRect.top;
        int right = mRect.left + mRect.right;
        int bottom = mRect.top + mRect.bottom;
        if (Math.abs(f-left)<=mInterval && Math.abs(g-top)<=mInterval) {
            return DRAG_LEFT_TOP;
        } else if (Math.abs(f-right)<=mInterval && Math.abs(g-top)<=mInterval) {
            return DRAG_RIGHT_TOP;
        } else if (Math.abs(f-left)<=mInterval && Math.abs(g-bottom)<=mInterval) {
            return DRAG_LEFT_BOTTOM;
        } else if (Math.abs(f-right)<=mInterval && Math.abs(g-bottom)<=mInterval) {
            return DRAG_RIGHT_BOTTOM;
        } else if (Math.abs(f-left)<=mInterval && g>top+mInterval && g<bottom-mInterval) {
            return DRAG_LEFT;
        } else if (Math.abs(f-right)<=mInterval && g>top+mInterval && g<bottom-mInterval) {
            return DRAG_RIGHT;
        } else if (Math.abs(f-left)<=mInterval && g>top+mInterval && g<bottom-mInterval) {
            return DRAG_LEFT;
        }
    }

```

```

    } else if (Math.abs(g-top)<=mInterval && f>left+mInterval && f<right-mInterval) {
        return DRAG_TOP;
    } else if (Math.abs(g-bottom)<=mInterval && f>left+mInterval && f<right-mInterval) {
        return DRAG_BOTTOM;
    } else if (f>left+mInterval && f<right-mInterval && g>top+mInterval && g<bottom-mInterval) {
        return DRAG_WHOLE;
    } else if (f+mInterval<left || f-mInterval>right || g+mInterval<top || g-mInterval>bottom) {
        return IMAGE_TRANSLATE;
    } else {
        return DRAG_NONE;
    }
}

private ImageChangeListener mListener;
public void setImageChangeListener(ImageChangeListener listener) {
    mListener = listener;
}

public static interface ImageChangeListener {
    public abstract void onImageClick();
    public abstract void onImageLongClick();
    public abstract void onImageTraslate(int offsetX, int offsetY, boolean bReset);
    public abstract void onImageScale(float ratio);
    public abstract void onImageRotate(int degree);
}
}

```

11.6

App()(MotionEvent)(
)"——"App

4

(1)

(2)

(3)

(4)

第12章 动画



12.1

GIF

12.1.1

Android

AnimationDrawableAnimationDrawable

addFrame()

setOneShottruefalse

start

stop

isRunning

ImageViewAnimationDrawableImageViewsetImageDrawable

```
private void showFrameAnimByCode() {  
    //帧动画需要把每帧图片加入 AnimationDrawable 队列  
    ad_frame = new AnimationDrawable();  
    ad_frame.addFrame(getResources().getDrawable(R.drawable.flow_p1), 50);  
    ad_frame.addFrame(getResources().getDrawable(R.drawable.flow_p2), 50);  
    ad_frame.addFrame(getResources().getDrawable(R.drawable.flow_p3), 50);  
    ad_frame.addFrame(getResources().getDrawable(R.drawable.flow_p4), 50);  
    ad_frame.addFrame(getResources().getDrawable(R.drawable.flow_p5), 50);  
    ad_frame.addFrame(getResources().getDrawable(R.drawable.flow_p6), 50);  
    ad_frame.addFrame(getResources().getDrawable(R.drawable.flow_p7), 50);  
    ad_frame.addFrame(getResources().getDrawable(R.drawable.flow_p8), 50);  
    ad_frame.setOneShot(false);  
    iv_frame_anim.setImageDrawable(ad_frame);  
    ad_frame.start();  
}
```

animation



12-11

animation



12-22

animation



12-33

XMLImageViewsetImageResourceImageViewgetDrawable

```
<animation-list xmlns:android="http://schemas.android.com/apk/res/android" android:oneshot="false" >
  <item android:drawable="@drawable/flow_p1" android:duration="50"/>
  <item android:drawable="@drawable/flow_p2" android:duration="50"/>
  <item android:drawable="@drawable/flow_p3" android:duration="50"/>
  <item android:drawable="@drawable/flow_p4" android:duration="50"/>
  <item android:drawable="@drawable/flow_p5" android:duration="50"/>
  <item android:drawable="@drawable/flow_p6" android:duration="50"/>
  <item android:drawable="@drawable/flow_p7" android:duration="50"/>
  <item android:drawable="@drawable/flow_p8" android:duration="50"/>
</animation-list>
```

```
private void showFrameAnimByXml() {
    iv_frame_anim.setImageResource(R.drawable.frame_anim);
    ad_frame = (AnimationDrawable) iv_frame_anim.getDrawable();
    ad_frame.start();
}
```

12.1.2GIF

GIFWindowsAndroidPNGJPGGIFGIFGIFGIF

GIF

(1)GIF

(2)GIFGIF

GIFGIF

```

private void showGifAnimation() {
    ImageView iv_gif = (ImageView) findViewById(R.id.iv_gif);
    InputStream is = getResources().openRawResource(R.raw.welcome);
    GifImage gifImage = new GifImage();
    int code = gifImage.read(is);
    if (code == GifImage.STATUS_OK) {
        GifImage.GifFrame[] frameList = gifImage.getFrames();
        AnimationDrawable ad_gif = new AnimationDrawable();
        for (int i=0; i<frameList.length; i++) {
            //BitmapDrawable 用于把 Bitmap 格式转换为 Drawable 格式
            BitmapDrawable bd = new BitmapDrawable(getResources(), frameList[i].image);
            ad_gif.addFrame(bd, frameList[i].delay);
        }
        ad_gif.setOneShot(false);
        iv_gif.setImageDrawable(ad_gif);
        ad_gif.start();
    } else if (code == GifImage.STATUS_FORMAT_ERROR) {
        Toast.makeText(this, "该图片不是 gif 格式", Toast.LENGTH_LONG).show();
    } else {
        Toast.makeText(this, "gif 图片读取失败:" + code, Toast.LENGTH_LONG).show();
    }
}
}

```

GIF12-412-512-4GIF12-5GIF

animation



12-4GIF



12-5GIF

12.1.3

(0.5)AndroidTransitionDrawable

ImageViewsetImageDrawableTransitionDrawable

2

startTransition

resetTransition

reverseTransition

```
private void showFadeAnimation() {  
    //淡入淡出动画需要先设置一个 Drawable 数组，用于变换图片  
    Drawable[] drawableArray = {  
        getResources().getDrawable(R.drawable.fade_begin),  
        getResources().getDrawable(R.drawable.fade_end)  
    };  
    TransitionDrawable td_fade = new TransitionDrawable(drawableArray);  
    iv_fade_anim.setImageDrawable(td_fade);  
    td_fade.startTransition(3000);  
}
```

12-612-712-612-7

animation



12-6



12-7

Android12-1

12-1Android

Drawable 图形类	说明	XML 节点名称	参考章节
ColorDrawable	颜色图形	color	第 2 章的“2.1.2 颜色”
StateListDrawable	状态列表图形	selector	第 2 章的“2.4.2 状态列表图形”
ShapeDrawable	形状图形	shape	第 2 章的“2.4.3 形状图形”
GradientDrawable	渐变图形	gradient	第 2 章的“2.4.3 形状图形”
NinePatchDrawable	点九图形	nine-patch	第 2 章的“2.4.4 九宫格图片”
LayerDrawable	层次图形	layer-list	第 6 章的“6.4.2 进度条 ProgressBar”
ClipDrawable	裁剪图形	clip	第 6 章的“6.4.2 进度条 ProgressBar”
BitmapDrawable	位图图形	bitmap	第 11 章的“11.5.2 小知识：图像的基本加工”
AnimationDrawable	动画图形	animation-list	第 12 章的“12.1.1 帧动画的实现”
TransitionDrawable	过渡图形	transition	第 12 章的“12.1.3 淡入淡出动画”

12.2

411

12.2.1

12.1 TransitionDrawable Android 4.4
“”4(AlphaAnimation TranslateAnimation
ScaleAnimation RotateAnimation) Animation Animation Animation
setFillAfter true false
setRepeatMode Animation.RESTART Animation.REVERSE
Animation.RESTART
setRepeatCount 0
setDuration
setInterpolator
➤ onAnimationStart
➤ onAnimationEnd
➤ onAnimationRepeat
setAnimationListener AnimationListener3
ImageView View startAnimation
clearAnimation

(1) 0.0~1.001

(2)

(3) 0.52

(4) 12-2

Animation 类的坐标类型	说明
ABSOLUTE	绝对位置
RELATIVE_TO_SELF	相对自身位置
RELATIVE_TO_PARENT	相对上级位置

4

```

public class TweenAnimActivity extends AppCompatActivity implements AnimationListener {
    private ImageView iv_tween_anim;
    private Animation alphaAnim, translateAnim, scaleAnim, rotateAnim;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_tween_anim);
        iv_tween_anim = (ImageView) findViewById(R.id.iv_tween_anim);
        initAnim();
        initTweenSpinner();
    }

    private void initAnim() {
        // 从完全透明变为即将不透明
        alphaAnim = new AlphaAnimation(1.0f, 0.1f);
        alphaAnim.setDuration(3000);
        alphaAnim.setFillAfter(true);
        // 向左平移 200
        translateAnim = new TranslateAnimation(1.0f, -200f, 1.0f, 1.0f);
        translateAnim.setDuration(3000);
        translateAnim.setFillAfter(true);
        // 宽度不变，高度变为原来的二分之一
        scaleAnim = new ScaleAnimation(1.0f, 1.0f, 1.0f, 0.5f);
        scaleAnim.setDuration(3000);
        scaleAnim.setFillAfter(true);
        // 围绕着圆心顺时针旋转 360 度
        rotateAnim = new RotateAnimation(0f, 360f, Animation.RELATIVE_TO_SELF,
            0.5f, Animation.RELATIVE_TO_SELF, 0.5f);
        rotateAnim.setDuration(3000);
    }
}

```

```

        rotateAnim.setFillAfter(true);
    }

    private void initTweenSpinner() {
        ArrayAdapter<String> tweenAdapter = new ArrayAdapter<String>(this,
            R.layout.item_select, tweenArray);
        Spinner sp_tween = (Spinner) findViewById(R.id.sp_tween);
        sp_tween.setPrompt("请选择补间动画类型");
        sp_tween.setAdapter(tweenAdapter);
        sp_tween.setOnItemSelectedListener(new TweenSelectedListener());
        sp_tween.setSelection(0);
    }

    private String[] tweenArray= {"灰度动画", "平移动画", "缩放动画", "旋转动画"};
    class TweenSelectedListener implements OnItemSelectedListener {
        public void onItemSelected(AdapterView<?> arg0, View arg1, int arg2, long arg3) {
            if (arg2 == 0) {
                iv_tween_anim.startAnimation(alphaAnim);
                alphaAnim.setAnimationListener(TweenAnimActivity.this);
            } else if (arg2 == 1) {
                iv_tween_anim.startAnimation(translateAnim);
                translateAnim.setAnimationListener(TweenAnimActivity.this);
            } else if (arg2 == 2) {
                iv_tween_anim.startAnimation(scaleAnim);
                scaleAnim.setAnimationListener(TweenAnimActivity.this);
            } else if (arg2 == 3) {
                iv_tween_anim.startAnimation(rotateAnim);
                rotateAnim.setAnimationListener(TweenAnimActivity.this);
            }
        }

        public void onNothingSelected(AdapterView<?> arg0) {
        }
    }

    @Override
    public void onAnimationStart(Animation animation) {
    }

    @Override
    public void onAnimationEnd(Animation animation) {
        if (animation.equals(alphaAnim)) {
            Animation alphaAnim2 = new AlphaAnimation(0.1f, 1.0f);

```

```

        alphaAnim2.setDuration(3000);
        alphaAnim2.setFillAfter(true);
        iv_tween_anim.startAnimation(alphaAnim2);
    } else if (animation.equals(translateAnim)) {
        Animation translateAnim2 = new TranslateAnimation(-200f, 1.0f, 1.0f, 1.0f);
        translateAnim2.setDuration(3000);
        translateAnim2.setFillAfter(true);
        iv_tween_anim.startAnimation(translateAnim2);
    } else if (animation.equals(scaleAnim)) {
        Animation scaleAnim2 = new ScaleAnimation(1.0f, 1.0f, 0.5f, 1.0f);
        scaleAnim2.setDuration(3000);
        scaleAnim2.setFillAfter(true);
        iv_tween_anim.startAnimation(scaleAnim2);
    } else if (animation.equals(rotateAnim)) {
        Animation rotateAnim2 = new RotateAnimation(0f, -360f,
            Animation.RELATIVE_TO_SELF, 0.5f, Animation.RELATIVE_TO_SELF, 0.5f);
        rotateAnim2.setDuration(3000);
        rotateAnim2.setFillAfter(true);
        iv_tween_anim.startAnimation(rotateAnim2);
    }
}

@Override
public void onAnimationRepeat(Animation animation) {
}
}

```

12-8~12-1512-812-912-1012-1112-1212-1312-1412-15



12-8

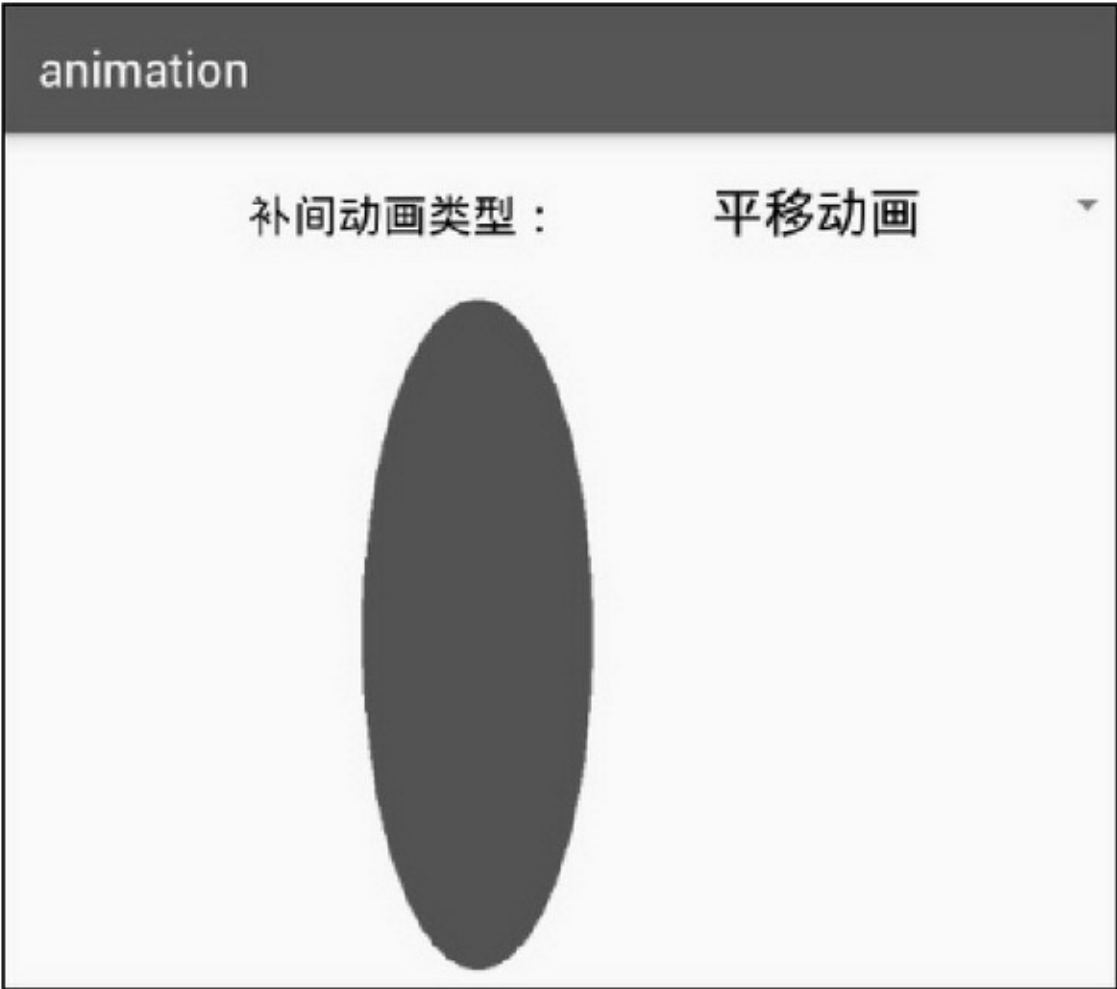
animation

补间动画类型：

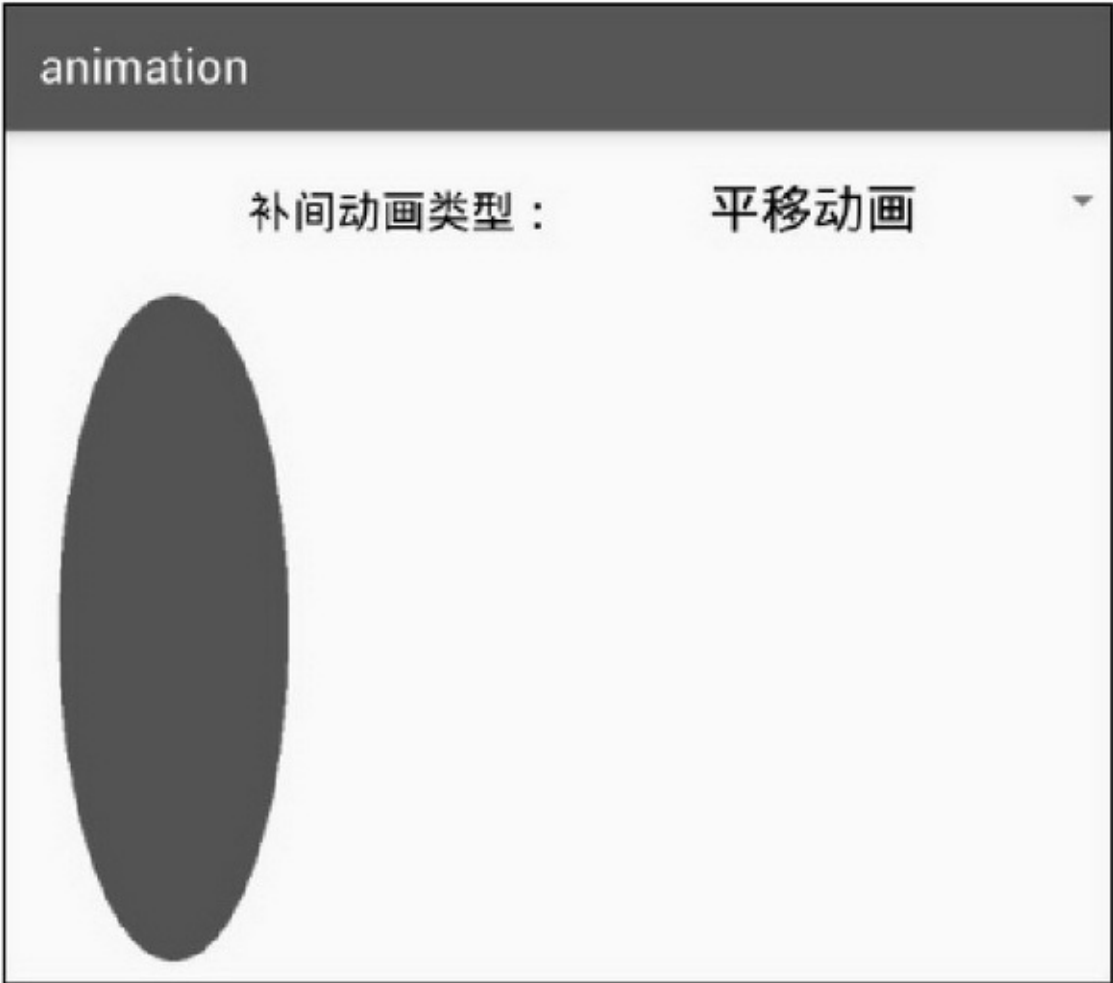
灰度动画



12-9



12-10



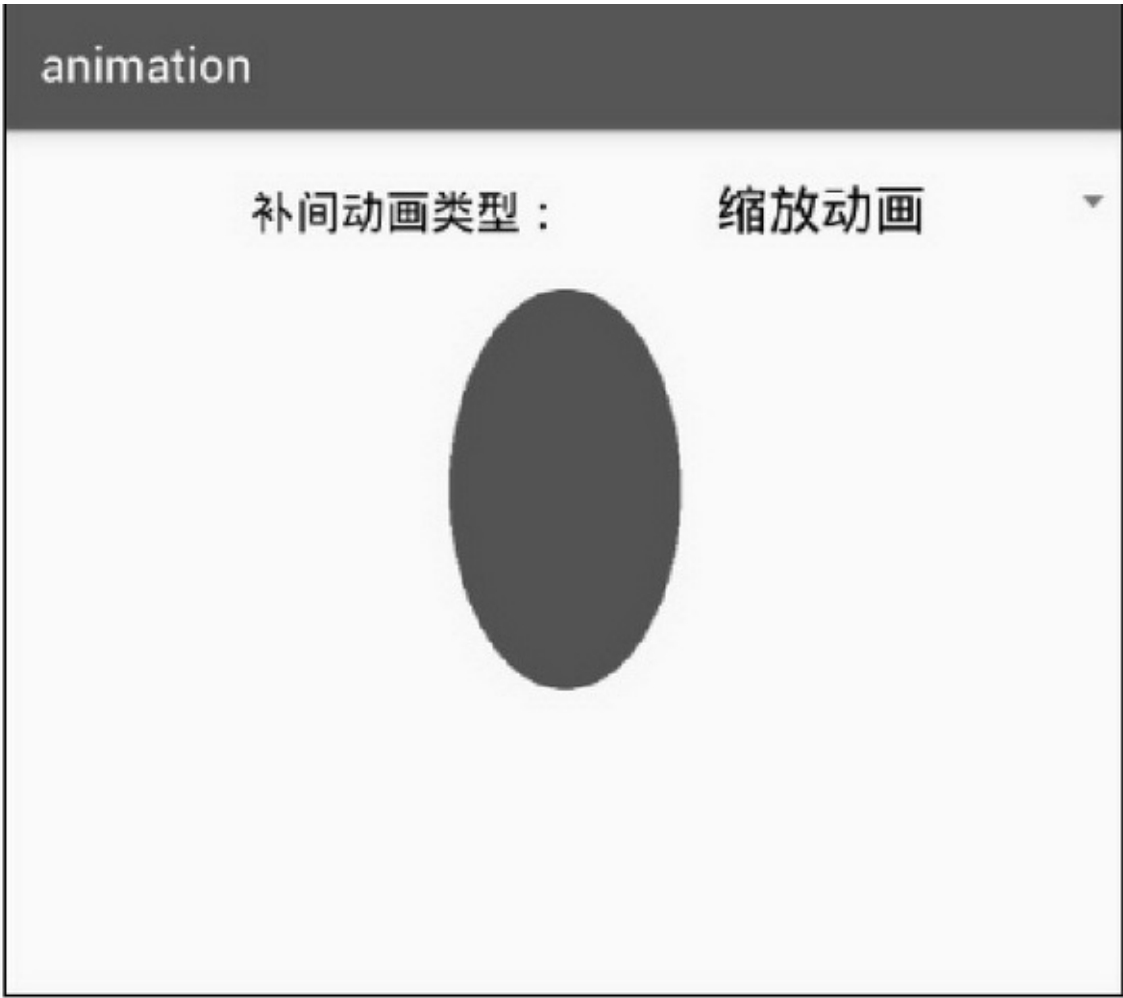
animation

补间动画类型：

缩放动画



12-12



animation

补间动画类型： 旋转动画



12-14



12-15

12.2.2

RotateAnimationRotateAnimation3

```

private void initializePivotPoint() {
    if (mPivotXType == ABSOLUTE) {
        mPivotX = mPivotXValue;
    }
    if (mPivotYType == ABSOLUTE) {
        mPivotY = mPivotYValue;
    }
}

@Override
protected void applyTransformation(float interpolatedTime, Transformation t) {
    float degrees = mFromDegrees + ((mToDegrees - mFromDegrees) * interpolatedTime);
    float scale = getScaleFactor();

    if (mPivotX == 0.0f && mPivotY == 0.0f) {
        t.getMatrix().setRotate(degrees);
    } else {
        t.getMatrix().setRotate(degrees, mPivotX * scale, mPivotY * scale);
    }
}

@Override
public void initialize(int width, int height, int parentWidth, int parentHeight) {
    super.initialize(width, height, parentWidth, parentHeight);
    mPivotX = resolveSize(mPivotXType, mPivotXValue, width, parentWidth);
    mPivotY = resolveSize(mPivotYType, mPivotYValue, height, parentHeight);
}

```

applyTransformationdegrees

RotateAnimation

步骤 01 ()

步骤 02

3

(1)(60)

(2)(12060)

(3)

(1)30

(2)3(3)40~11~33~4

```
protected void applyTransformation(float interpolatedTime, Transformation t) {
    float degrees;
    float leftPos = (float) (1.0 / 4.0);
    float rightPos = (float) (3.0 / 4.0);
    if (interpolatedTime <= leftPos) {
        degrees = mMiddleDegrees + ((mLeftDegrees - mMiddleDegrees) * interpolatedTime * 4);
    } else if (interpolatedTime > leftPos && interpolatedTime < rightPos) {
        degrees = mLeftDegrees + ((mRightDegrees - mLeftDegrees) * (interpolatedTime - leftPos) * 2);
    } else {
        degrees = mRightDegrees + ((mMiddleDegrees - mRightDegrees) * (interpolatedTime - rightPos) * 4);
    }

    float scale = getScaleFactor();
    if (mPivotX == 0.0f && mPivotY == 0.0f) {
        t.getMatrix().setRotate(degrees);
    } else {
        t.getMatrix().setRotate(degrees, mPivotX * scale, mPivotY * scale);
    }
}
```

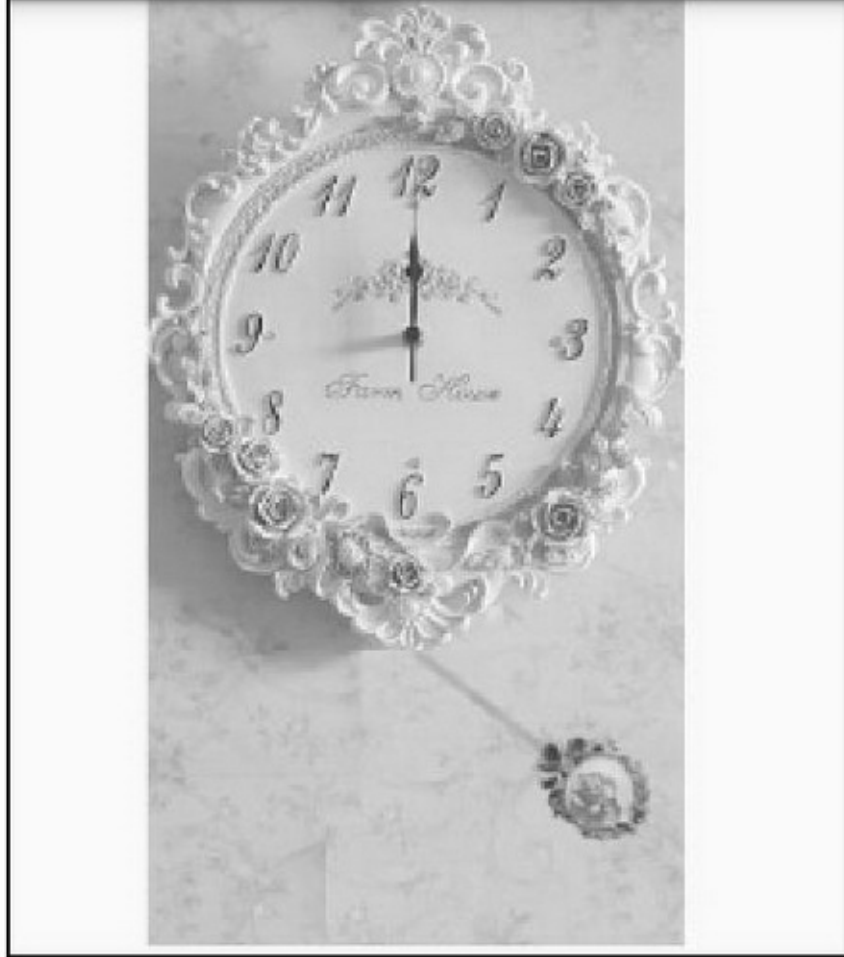
12-1612-1712-1612-17

animation



12-16

animation



12-17

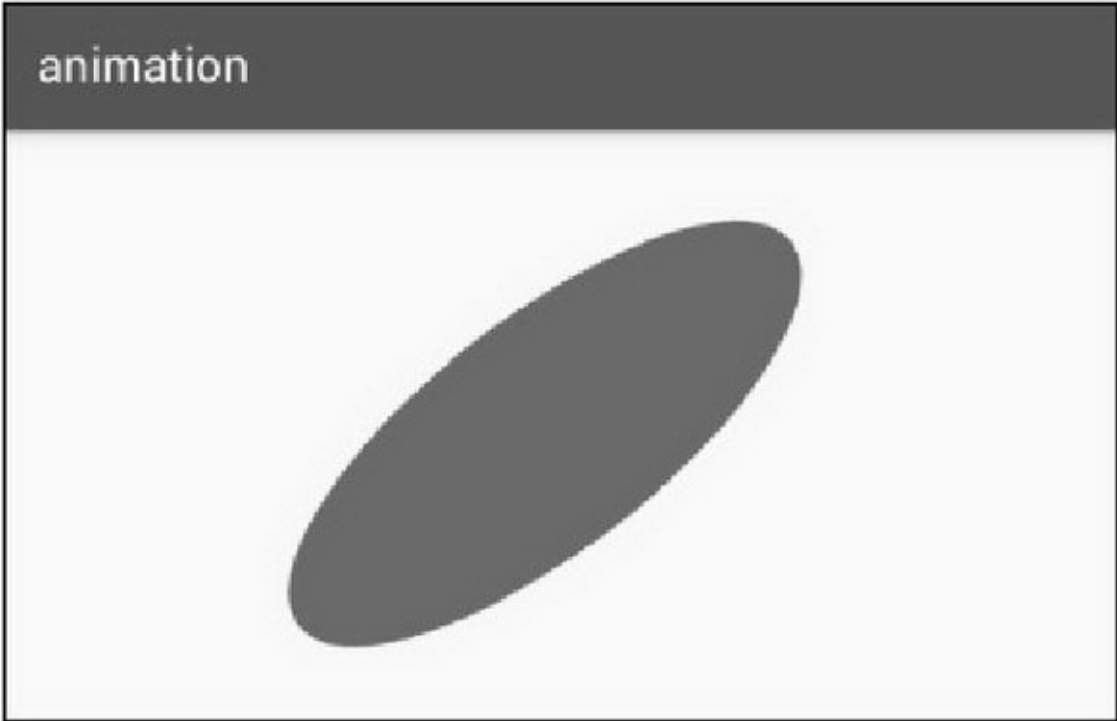
12.2.3

AnimationSet

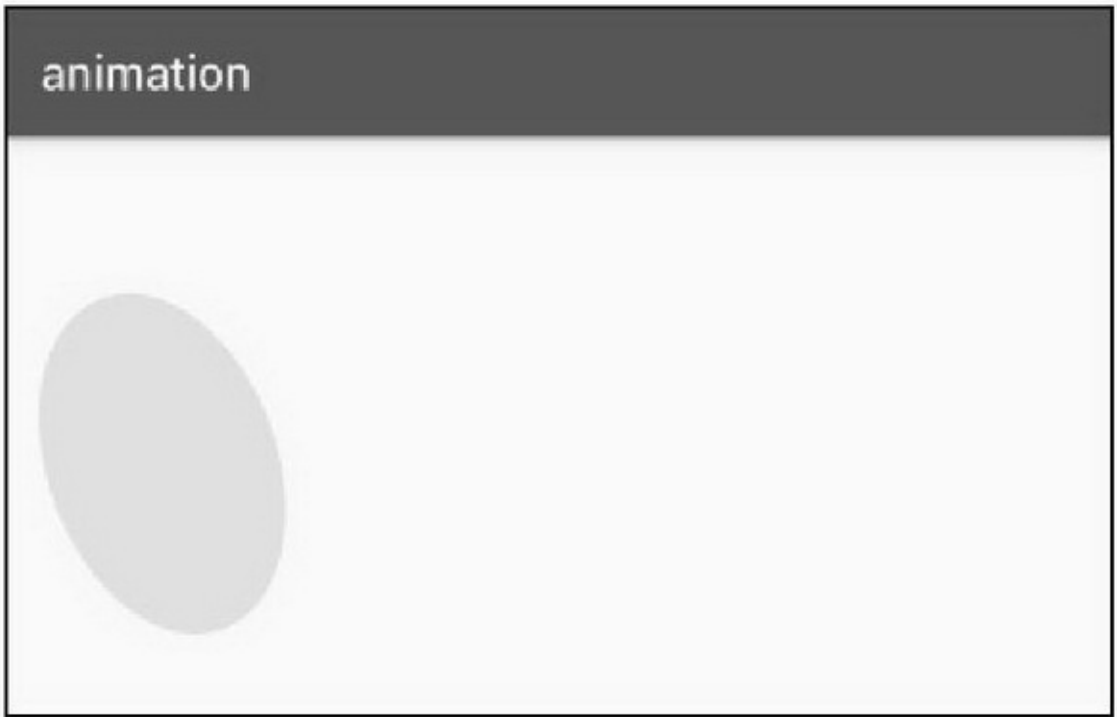
AnimationaddAnimation

```
private void initAnim() {
    alphaAnim = new AlphaAnimation(1.0f, 0.1f); // 灰度动画
    alphaAnim.setDuration(3000);
    alphaAnim.setFillAfter(true);
    translateAnim = new TranslateAnimation(1.0f, -200f, 1.0f, 1.0f); // 平移动画
    translateAnim.setDuration(3000);
    translateAnim.setFillAfter(true);
    scaleAnim = new ScaleAnimation(1.0f, 1.0f, 1.0f, 0.5f); // 缩放动画
    scaleAnim.setDuration(3000);
    scaleAnim.setFillAfter(true);
    rotateAnim = new RotateAnimation(0f, 360f, Animation.RELATIVE_TO_SELF,
        0.5f, Animation.RELATIVE_TO_SELF, 0.5f); // 旋转动画
    rotateAnim.setDuration(3000);
    rotateAnim.setFillAfter(true);
    setAnim = new AnimationSet(true); // 集合动画
    setAnim.addAnimation(translateAnim);
    setAnim.addAnimation(alphaAnim);
    setAnim.addAnimation(scaleAnim);
    setAnim.addAnimation(rotateAnim);
    setAnim.setFillAfter(true);
    iv_anim_set.startAnimation(setAnim);
    setAnim.setAnimationListener(this);
}
```

12-1812-1912-1812-19



12-18



12-19

XMLXML4

```
<set xmlns:android="http://schemas.android.com/apk/res/android">
  <alpha android:duration="3000" android:fromAlpha="1.0" android:toAlpha="0.1" />
  <translate android:duration="3000" android:fromXDelta="1.0" android:toXDelta="-200"
    android:fromYDelta="1.0" android:toYDelta="1.0" />
  <scale android:duration="3000" android:fromXScale="1.0" android:toXScale="1.0"
    android:fromYScale="1.0" android:toYScale="0.5" />
  <rotate android:duration="3000" android:fromDegrees="0" android:toDegrees="360"
    android:pivotX="50%" android:pivotY="50%" />
</set>
```

AnimationUtils.loadAnimation4

setAnim.addAnimation(AnimationUtils.loadAnimation(this, R.anim.anim_set)); XML12-1812-19

12.2.4

11ViewFlipperGestureDetectorBannerViewPager

11ViewFlipper4

setInAnimation

getInAnimation

setOutAnimation

getOutAnimation

4

44

(1)Bannerpush_left_in.xml

```
<set xmlns:android="http://schemas.android.com/apk/res/android">
  <translate android:duration="1500" android:fromXDelta="100.0%p" android:toXDelta="0.0" />
  <alpha android:duration="1500" android:fromAlpha="0.1" android:toAlpha="1.0" />
</set>
```

(2)Bannerpush_left_out.xml

```
<set xmlns:android="http://schemas.android.com/apk/res/android">
  <translate android:duration="1500" android:fromXDelta="0.0" android:toXDelta="-100.0%p" />
  <alpha android:duration="1500" android:fromAlpha="1.0" android:toAlpha="0.1" />
</set>
```

(3)Bannerpush_right_in.xml

```
<set xmlns:android="http://schemas.android.com/apk/res/android">
  <translate android:duration="1500" android:fromXDelta="-100.0%p" android:toXDelta="0.0" />
  <alpha android:duration="1500" android:fromAlpha="0.1" android:toAlpha="1.0" />
</set>
```

(4)Bannerpush_right_out.xml

```
<set xmlns:android="http://schemas.android.com/apk/res/android">
  <translate android:duration="1500" android:fromXDelta="0.0" android:toXDelta="100.0%p" />
  <alpha android:duration="1500" android:fromAlpha="1.0" android:toAlpha="0.1" />
</set>
```

11BannerFlipper

```

private void startFlip() {
    mFlipper.startFlipping();
    mFlipper.setInAnimation(AnimationUtils.loadAnimation(mContext, R.anim.push_left_in));
    mFlipper.setOutAnimation(AnimationUtils.loadAnimation(mContext, R.anim.push_left_out));
    mFlipper.getOutAnimation().setAnimationListener(new BannerAnimationListener(this));
    mFlipper.showNext();
}

private void backFlip() {
    mFlipper.startFlipping();
    mFlipper.setInAnimation(AnimationUtils.loadAnimation(mContext, R.anim.push_right_in));
    mFlipper.setOutAnimation(AnimationUtils.loadAnimation(mContext, R.anim.push_right_out));
    mFlipper.getOutAnimation().setAnimationListener(new BannerAnimationListener(this));
    mFlipper.showPrevious();
    mFlipper.setInAnimation(AnimationUtils.loadAnimation(mContext, R.anim.push_left_in));
    mFlipper.setOutAnimation(AnimationUtils.loadAnimation(mContext, R.anim.push_left_out));
    mFlipper.getOutAnimation().setAnimationListener(new BannerAnimationListener(this));
}

private class BannerAnimationListener implements Animation.AnimationListener {
    private BannerAnimationListener(BannerFlipper bannerFlipper) {
    }

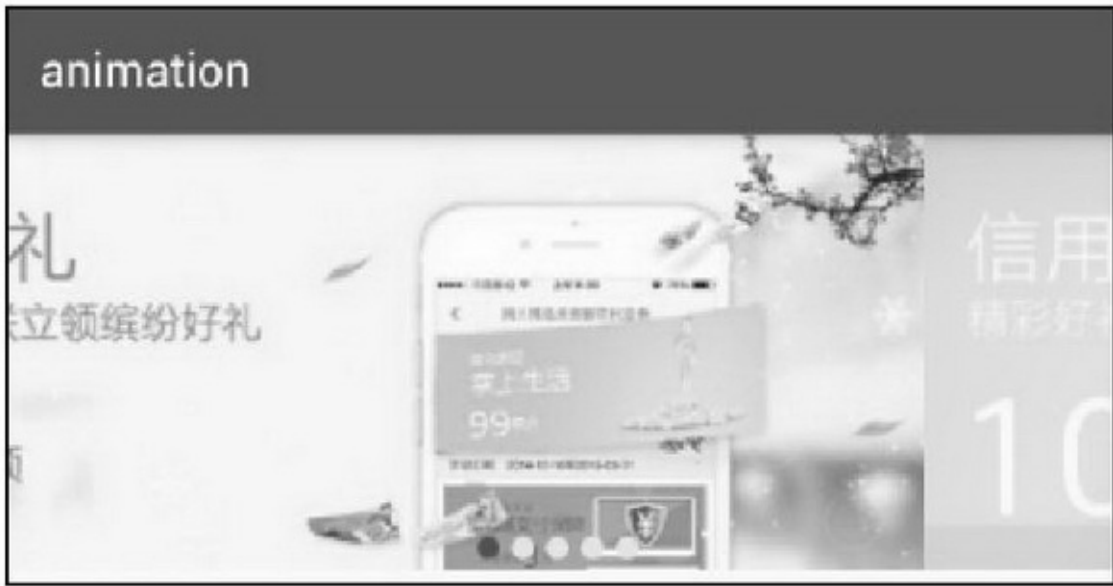
    @Override
    public final void onAnimationEnd(Animation paramAnimation) {
        int position = mFlipper.getDisplayedChild();
        ((RadioButton) mGroup.getChildAt(position)).setChecked(true);
    }

    @Override
    public final void onAnimationRepeat(Animation paramAnimation) {
    }

    @Override
    public final void onAnimationStart(Animation paramAnimation) {
    }
}

```

12-2012-2112-2012-21



12-20



12-21

7BannerAndroid

12.3

12.3.1

View46612-3

12-3

View 类的属性名称	属性说明	属性设置方法	对应的补间动画
alpha	透明度	setAlpha	灰度动画
rotation	旋转角度	setRotation	旋转动画
scaleX	横坐标的缩放比例	setScaleX	缩放动画
scaleY	纵坐标的缩放比例	setScaleY	缩放动画
translationX	横坐标的平移距离	setTranslationX	平移动画
translationY	纵坐标的平移距离	setTranslationY	平移动画

6Android3.0ObjectAnimator

ObjectAnimator

ofInt

ofFloat

ofArgb

ofObjectRect4ofof3A4BAB3A4

B5CABBC

setRepeatModeValueAnimator.RESTART

ValueAnimator.REVERSEValueAnimator.RESTART

setRepeatCount0

setDuration

setInterpolator

setEvaluator

start
cancel
end
pause
resume
reverse
isRunningisRunningtrue
isPaused
isStarted
 onAnimationStart
 onAnimationEnd
 onAnimationCancel
 onAnimationRepeat
addListenerAnimatorListener4
removeListener
removeAllListeners

```
public class ObjectAnimActivity extends AppCompatActivity {
    private ImageView iv_object_anim;
    private ObjectAnimator alphaAnim, translateAnim, scaleAnim, rotateAnim, clipAnim;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_object_anim);
        iv_object_anim = (ImageView) findViewById(R.id.iv_object_anim);
        initAnim();
        initObjectSpinner();
    }

    private void initAnim() {
        alphaAnim = ObjectAnimator.ofFloat(iv_object_anim, "alpha", 1f, 0.1f, 1f);
        translateAnim = ObjectAnimator.ofFloat(iv_object_anim, "translationX", 0f, -200f, 0f, 200f, 0f);
        scaleAnim = ObjectAnimator.ofFloat(iv_object_anim, "scaleY", 1f, 0.5f, 1f);
        rotateAnim = ObjectAnimator.ofFloat(iv_object_anim, "rotation", 0f, 360f, 0f);
    }
}
```

```

private void initObjectSpinner() {
    ArrayAdapter<String> objectAdapter = new ArrayAdapter<String>(this,
        R.layout.item_select, objectArray);
    Spinner sp_object = (Spinner) findViewById(R.id.sp_object);
    sp_object.setPrompt("请选择属性动画类型");
    sp_object.setAdapter(objectAdapter);
    sp_object.setOnItemSelectedListener(new ObjectSelectedListener());
    sp_object.setSelection(0);
}

private String[] objectArray={"灰度动画", "平移动画", "缩放动画", "旋转动画", "裁剪动画"};
class ObjectSelectedListener implements OnItemSelectedListener {
    public void onItemSelected(AdapterView<?> arg0, View arg1, int arg2, long arg3) {
        showAnimation(arg2);
    }

    public void onNothingSelected(AdapterView<?> arg0) {
    }
}

@TargetApi(Build.VERSION_CODES.JELLY_BEAN_MR2)
private void showAnimation(int type) {
    ObjectAnimator anim = null;
    if (type == 0) {
        anim = alphaAnim;
    } else if (type == 1) {
        anim = translateAnim;
    } else if (type == 2) {
        anim = scaleAnim;
    } else if (type == 3) {
        anim = rotateAnim;
    } else if (type == 4) {
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.JELLY_BEAN_MR2) {
            int width = iv_object_anim.getWidth();
            int height = iv_object_anim.getHeight();
            clipAnim = ObjectAnimator.ofObject(iv_object_anim, "clipBounds",
                new RectEvaluator(), new Rect(0,0,width,height),
                new Rect(width/3,height/3,width/3*2,height/3*2),
                new Rect(0,0,width,height));
            anim = clipAnim;
        } else {
            Toast.makeText(this,
                "裁剪动画要求 Android 为 4.3 以上版本", Toast.LENGTH_SHORT).show();
            return;
        }
    }
}

```

```

        if (anim != null) {
            anim.setDuration(3000);
            anim.start();
        }
    }
}

```

12-2212-2312-2212-23





12-23

12.3.2

AnimationSetAnimatorSetAnimatorSetObjectAnimator
AnimatorAnimatorSet
setDuration
setInterpolator
➤ with

 before

 after

playAnimatorSet.BuilderBuilder

start

pause

resume

cancel

end

isRunning

isStarted

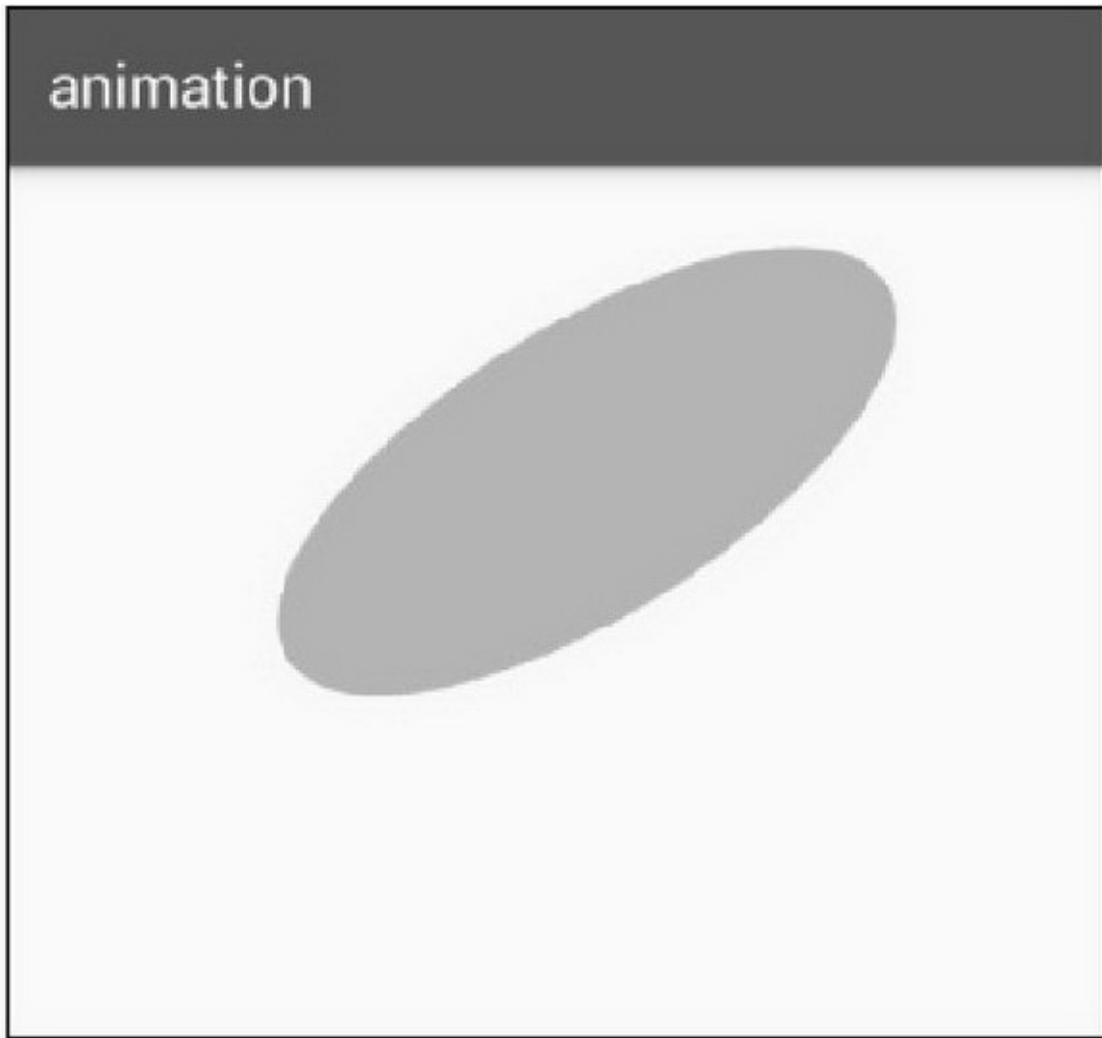
```
private void initAnim() {  
    ObjectAnimator anim1 = ObjectAnimator.ofFloat(iv_object_group, "translationX", 0f, 100f);  
    ObjectAnimator anim2 = ObjectAnimator.ofFloat(iv_object_group, "alpha", 1f, 0.1f, 1f, 0.5f, 1f);  
    ObjectAnimator anim3 = ObjectAnimator.ofFloat(iv_object_group, "rotation", 0f, 360f);  
    ObjectAnimator anim4 = ObjectAnimator.ofFloat(iv_object_group, "scaleY", 1f, 0.5f, 1f);  
    ObjectAnimator anim5 = ObjectAnimator.ofFloat(iv_object_group, "translationX", 100f, 0f);  
    animSet = new AnimatorSet();  
    AnimatorSet.Builder builder = animSet.play(anim2);  
    // anim1 先执行，然后再同步执行 anim2、anim3、anim3，最后执行 anim5  
    builder.with(anim3).with(anim4).after(anim1).before(anim5);  
    animSet.setDuration(4500);  
    animSet.start();  
}
```

12-2412-2512-2412-25

animation



12-24



12-25

12.3.3

setInterpolator12-4

12-4

插值器的实现类	说明
LinearInterpolator	匀速插值器
AccelerateInterpolator	加速插值器
DecelerateInterpolator	减速插值器
AccelerateDecelerateInterpolator	落水插值器，即前半段加速、后半段减速
AnticipateInterpolator	射箭插值器，后退几步再往前冲
OvershootInterpolator	回旋插值器，冲过头再归位
AnticipateOvershootInterpolator	射箭回旋插值器，后退几步再往前冲，冲过头再归位
BounceInterpolator	震荡插值器，类似皮球落地（落地后会弹起几次）
CycleInterpolator	钟摆插值器，以开始位置为中线而晃动（类似摇摆动画，开始位置与结束位置的距离就是摇摆的幅度）

ARGBRectsetEvaluator12-5

12-5

估值器的实现类	说明
IntEvaluator	整型估值器
FloatEvaluator	浮点型估值器
ArgbEvaluator	颜色估值器
RectEvaluator	矩形估值器

intfloatargbObject12-6

12-6

属性动画的构造方法	估值器	对应的属性说明
ofInt	IntEvaluator	整型类型的属性
ofFloat	FloatEvaluator	大部分状态属性，如 alpha、rotation、scaleY、translationX、textSize 等
ofArgb	ArgbEvaluator	颜色，如 backgroundColor、textColor 等
ofObject	RectEvaluator	裁剪范围，如 clipBounds

```
public class InterpolatorActivity extends AppCompatActivity implements AnimatorListener {
    private TextView tv_interpolator;
```

```

private ObjectAnimator animAcce, animDece, animLinear, animBounce;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_interpolator);
    tv_interpolator = (TextView) findViewById(R.id.tv_interpolator);
    initAnimator();
    initInterpolatorSpinner();
}

private void initInterpolatorSpinner() {
    ArrayAdapter<String> interpolatorAdapter = new ArrayAdapter<String>(this,
        R.layout.item_select, interpolatorArray);
    Spinner sp_interpolator = (Spinner) findViewById(R.id.sp_interpolator);
    sp_interpolator.setPrompt("请选择插值器类型");
    sp_interpolator.setAdapter(interpolatorAdapter);
    sp_interpolator.setOnItemSelectedListener(new InterpolatorSelectedListener());
    sp_interpolator.setSelection(0);
}

private String[] interpolatorArray={
    "背景色+加速插值器+颜色估值器", "旋转+减速插值器+浮点型估值器",
    "裁剪+匀速插值器+矩形估值器", "文字大小+震荡插值器+浮点型估值器");
class InterpolatorSelectedListener implements OnItemSelectedListener {
    public void onItemSelected(AdapterView<?> arg0, View arg1, int arg2, long arg3) {
        showInterpolator(arg2);
    }

    public void onNothingSelected(AdapterView<?> arg0) {
    }
}

private void initAnimator() {
    animAcce = ObjectAnimator.ofArgb(tv_interpolator, "backgroundColor", Color.RED, Color.LTGRAY);
    animAcce.setInterpolator(new AccelerateInterpolator());
    animAcce.setEvaluator(new ArgbEvaluator());

    animDece = ObjectAnimator.ofFloat(tv_interpolator, "rotation", 0f, 360f);
    animDece.setInterpolator(new DecelerateInterpolator());
    animDece.setEvaluator(new FloatEvaluator());

    animBounce = ObjectAnimator.ofFloat(tv_interpolator, "textSize", 20f, 60f);
    animBounce.setInterpolator(new BounceInterpolator());
    animBounce.setEvaluator(new FloatEvaluator());
}

```

```

@TargetApi(Build.VERSION_CODES.JELLY_BEAN_MR2)
private void showInterpolator(int type) {
    ObjectAnimator anim = null;
    if (type == 0) {
        anim = animAcce;
    } else if (type == 1) {
        anim = animDece;
    } else if (type == 2) {
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.JELLY_BEAN_MR2) {
            int width = tv_interpolator.getWidth();
            int height = tv_interpolator.getHeight();
            animLinear = ObjectAnimator.ofObject(tv_interpolator, "clipBounds",
                new RectEvaluator(), new Rect(0,0,width,height),
                new Rect(width/3,height/3,width/3*2,height/3*2),
                new Rect(0,0,width,height));
            animLinear.setInterpolator(new LinearInterpolator());
            anim = animLinear;
        } else {
            Toast.makeText(this,
                "矩形估值器要求 Android 为 4.3 以上版本", Toast.LENGTH_SHORT).show();
            return;
        }
    } else if (type == 3) {
        anim = animBounce;
    }
    if (anim != null) {
        anim.setDuration(2000);
        anim.start();
    }
}
}

```

12-2612-2712-2612-27

animation

插值器类型：文字大小+震荡插值器+浮点型估 ▾

看看插值器的效果是什么

12-26

animation

插值器类型：文字大小+震荡插值器+浮点型估 ▾

看看插值器的效果是什么

12.4

3

12.4.1

67BanneronDrawdispatchDraw

```
@Override
protected void onDraw(Canvas canvas) {
    super.onDraw(canvas);
    if (mRunning == true) {
        int width = getMeasuredWidth();
        int height = getMeasuredHeight();
        int diameter = Math.min(width, height);
        RectF rectf = new RectF((width - diameter) / 2, (height - diameter) / 2,
            (width + diameter) / 2, (height + diameter) / 2);
        canvas.drawArc(rectf, 0, mDrawingAngle, true, mPaint);
    }
}

private Runnable mRefresh = new Runnable() {
    @Override
    public void run() {
        mDrawingAngle += mIncrease;
        if (mDrawingAngle <= mEndAngle) {
            postInvalidate();
            mHandler.postDelayed(this, mInterval);
        } else {
            mRunning = false;
        }
    }
};
```

12-2812-2912-2812-29

animation





12-29

12.4.2


```

@Override
public void onClick(View v) {
    if (v.getId() == R.id.ll_content) {
        bSelected = !bSelected;
        tv_content.clearAnimation();
        final int deltaValue;
        final int startValue = tv_content.getHeight();
        if (bSelected) {
            deltaValue = tv_content.getLineHeight() * tv_content.getLineCount() - startValue;
        } else {
            deltaValue = tv_content.getLineHeight() * mNormalLines - startValue;
        }
        Animation animation = new Animation() {
            protected void applyTransformation(float interpolatedTime, Transformation t) {
                tv_content.setHeight((int) (startValue + deltaValue * interpolatedTime));
            }
        };
        animation.setDuration(500);
        tv_content.startAnimation(animation);
    }
}

```

12-3012-3112-3012-31

animation

遇到爱你的人，学会感恩。遇到你爱的人，学会付出；遇到你恨的人，学会原谅。遇到恨你的人，学会道歉；遇到欣赏你

animation

遇到爱你的人，学会感恩。遇到你爱的人，学会付出；遇到你恨的人，学会原谅。遇到恨你的人，学会道歉；遇到欣赏你的人，学会笑纳。遇到你欣赏的人，学会赞赏。遇到嫉妒你的人，学会低调。遇到你嫉

12-31

12.4.3 Scroller 112020

Android Scroller Scroller Scroller

startScroll

computeScrollOffset

getCurrX

getCurrY

getDuration

forceFinished

isFinished

computeScrollOffset

(1) computeScrollOffset is Finished

(2) computeScrollOffset is Finished true

Scroller scrollTo scrollBy

scrollTo

scrollBy scrollBy scrollTo

```

public class ScrollTextView extends TextView {
    private Scroller mScroller;

    public ScrollTextView(Context context) {
        this(context, null);
    }

    public ScrollTextView(Context context, AttributeSet attrs) {
        super(context, attrs);
        mScroller = new Scroller(context);
    }

    public void smoothScrollTo(int fx, int fy) {
        int dx = fx - mScroller.getFinalX();
        int dy = fy - mScroller.getFinalY();
        smoothScrollBy(dx, dy);
    }

    public void smoothScrollBy(int dx, int dy) {
        // 设置滚动偏移量，注意正数是往左滚、往上滚，负数才是往右滚、往下滚
        mScroller.startScroll(mScroller.getFinalX(), mScroller.getFinalY(), -dx, -dy);
        // 调用 invalidate 方法才能保证 computeScroll 方法被调用
        invalidate();
    }

    @Override
    public void computeScroll() {
        if (mScroller.computeScrollOffset()) { // 先判断 mScroller 滚动是否完成
            scrollTo(mScroller.getCurrX(), mScroller.getCurrY()); //调用 scrollTo 方法完成实际滚动
            postInvalidate(); // 刷新页面
        }
        super.computeScroll();
    }
}

```

12-3212-3312-3212-33



12-32



12-33

12.5QQ

QQ“QQ”

12.5.1



12-34

12-34

- (1)
 - (2)
 - (3)
 - (4)
 - (5)
 - (6)
 - (7)
- App

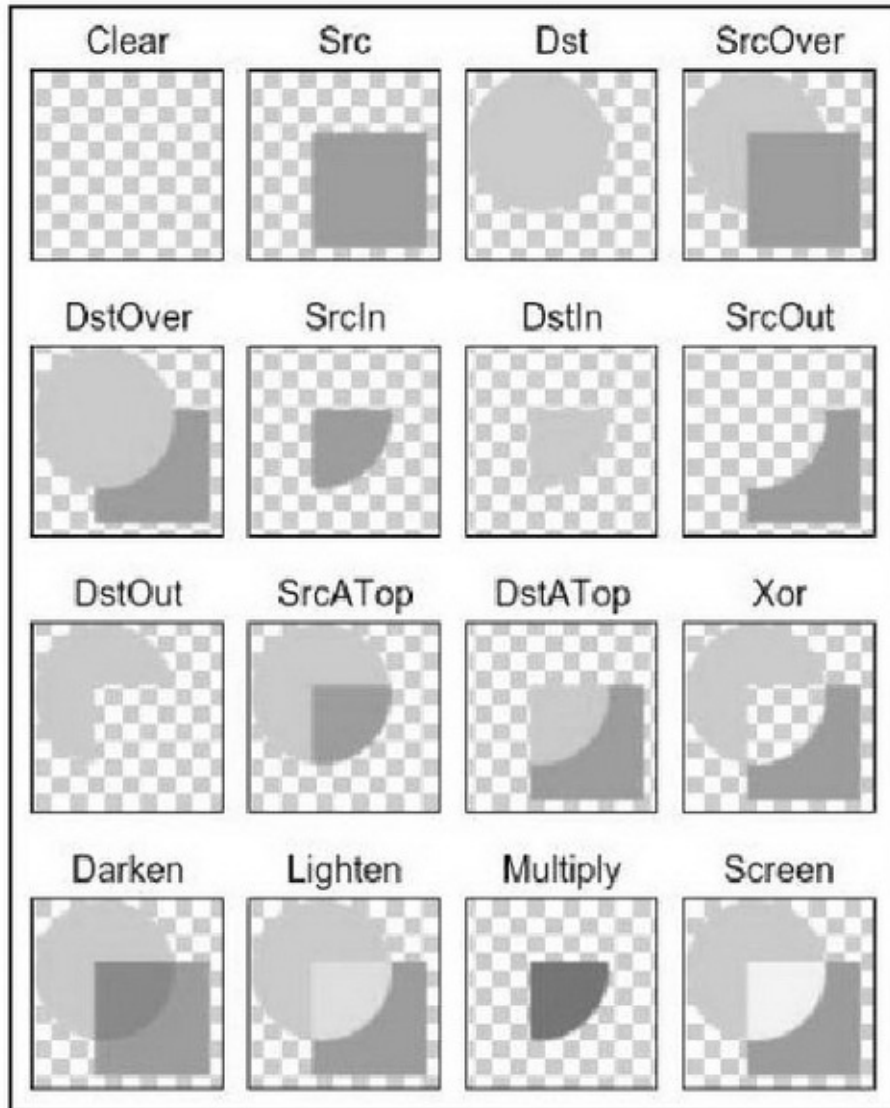
12.5.2

Canvas

Android12-7SrcDst

12-7

PorterDuff.Mode 类的图层模式	说明
CLEAR	不显示任何图形
SRC	只显示上层图形
DST	只显示下层图形
SRC_OVER	按通常情况显示，即重叠部分由上层遮盖下层
DST_OVER	重叠部分由下层遮盖上层，其余部分正常显示
SRC_IN	只显示重叠部分的上层图形
DST_IN	只显示重叠部分的下层图形
SRC_OUT	只显示上层图形的未重叠部分
DST_OUT	只显示下层图形的未重叠部分
SRC_ATOP	只显示上层图形区域，但重叠部分显示下层图形
DST_ATOP	只显示下层图形区域，但重叠部分显示上层图形
XOR	不显示重叠部分，其余部分正常显示
DARKEN	重叠部分按颜料混合方式加深，其余部分正常显示
LIGHTEN	重叠部分按光照重合方式加亮，其余部分正常显示
MULTIPLY	只显示重叠部分，且重叠部分的颜色混合加深
SCREEN	过滤重叠部分的深色，其余部分正常显示



12-35

12-35

setXfermodeDST_IN

```
public class ShutterView extends View {
    private final static String TAG = "ShutterView";
    private Context mContext;
    private Paint mPaint;
    private int mOrientation = LinearLayout.HORIZONTAL;
    private int mLeafCount = 10;
    private PorterDuff.Mode mMode = PorterDuff.Mode.DST_IN;
    private Bitmap mBitmap;
    private int mRatio = 0;

    public ShutterView(Context context) {
        this(context, null);
    }

    public ShutterView(Context context, AttributeSet attrs) {
        super(context, attrs);
        mContext = context;
        mPaint = new Paint();
    }

    public void setOrientation(int orientation) {
        mOrientation = orientation;
    }

    public void setLeafCount(int leaf_count) {
        mLeafCount = leaf_count;
    }

    public void setMode(PorterDuff.Mode mode) {
        mMode = mode;
    }

    public void setImageBitmap(Bitmap bitmap) {
        mBitmap = bitmap;
    }

    public void setRatio(int ratio) {
        mRatio = ratio;
        invalidate();
    }

    @SuppressWarnings("DrawAllocation")
    @Override
    protected void onDraw(Canvas canvas) {
        super.onDraw(canvas);
        if (mBitmap == null) {
```



```

        return;
    }
    int width = getMeasuredWidth();
    int height = getMeasuredHeight();
    canvas.drawColor(Color.TRANSPARENT);
    // 创建图片的遮罩
    Bitmap mask = Bitmap.createBitmap(width, height, mBitmap.getConfig());
    Canvas canvasMask = new Canvas(mask);
    for (int i = 0; i < mLeafCount; i++) {
        if (mOrientation == LinearLayout.HORIZONTAL) {
            int column_width = (int) Math.ceil(width*1f / mLeafCount);
            int left = column_width * i;
            int right = left + column_width * mRatio / 100;
            canvasMask.drawRect(left, 0, right, height, mPaint);
        } else {
            int row_height = (int) Math.ceil(height*1f / mLeafCount);
            int top = row_height * i;
            int bottom = top + row_height * mRatio / 100;
            canvasMask.drawRect(0, top, width, bottom, mPaint);
        }
    }
    // 设置离屏缓存
    int saveLayer = canvas.saveLayer(0, 0, width, height, null, Canvas.ALL_SAVE_FLAG);
    // 绘制目标图像
    Rect src = new Rect(0, 0, mBitmap.getWidth(), mBitmap.getHeight());
    Rect dst = new Rect(0, 0, width, width*mBitmap.getHeight()/mBitmap.getWidth());
    canvas.drawBitmap(mBitmap, src, dst, mPaint);
    // 设置混合模式 （只在源图像和目标图像相交的地方绘制目标图像）
    mPaint.setXfermode(new PorterDuffXfermode(mMode));
    // 再绘制 src 源图
    canvas.drawBitmap(mask, 0, 0, mPaint);
    // 还原混合模式
    mPaint.setXfermode(null);
    // 还原画布
    canvas.restoreToCount(saveLayer);
}
}

```

ShutterViewratio

```

public class ShutterActivity extends AppCompatActivity {
    private ShutterView sv_shutter;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_shutter);
        sv_shutter = (ShutterView) findViewById(R.id.sv_shutter);
        sv_shutter.setImageBitmap(BitmapFactory.decodeResource(getResources(), R.drawable.bdg03));
        initShutterSpinner();
    }

    private void initShutterSpinner() {
        ArrayAdapter<String> shutterAdapter = new ArrayAdapter<String>(this,
            R.layout.item_select, shutterArray);
        Spinner sp_shutter = (Spinner) findViewById(R.id.sp_shutter);
        sp_shutter.setPrompt("请选择百叶窗动画类型");
        sp_shutter.setAdapter(shutterAdapter);
        sp_shutter.setOnItemSelectedListener(new ShutterSelectedListener());
        sp_shutter.setSelection(0);
    }

    private String[] shutterArray={"水平五叶", "水平十叶", "水平二十叶",
        "垂直五叶", "垂直十叶", "垂直二十叶"};

    class ShutterSelectedListener implements OnItemSelectedListener {
        public void onItemSelected(AdapterView<?> arg0, View arg1, int arg2, long arg3) {
            sv_shutter.setOrientation((arg2<3)?LinearLayout.HORIZONTAL:LinearLayout.VERTICAL);
            if (arg2 == 0 || arg2 == 3) {
                sv_shutter.setLeafCount(5);
            } else if (arg2 == 1 || arg2 == 4) {
                sv_shutter.setLeafCount(10);
            } else if (arg2 == 2 || arg2 == 5) {
                sv_shutter.setLeafCount(20);
            }
            ObjectAnimator anim = ObjectAnimator.ofInt(sv_shutter, "ratio", 0, 100);
            anim.setDuration(3000);
            anim.start();
        }

        public void onNothingSelected(AdapterView<?> arg0) {
        }
    }
}

```

12-3612-3712-3612-37

animation

百叶窗动画样式：

水平五叶



animation

百叶窗动画样式：

水平五叶



12-37

MosaicView12-3812-3912-3812-39

animation

马赛克动画样式：

水平三十格





12-39

12.5.3

(1)XMLres/anim

(2)AndroidAndroid 4.3RectEvaluatorAndroid 4.3

12-40~12-4512-4012-4112-4212-4312-

4412-45

animation

正在播放灰度动画



animation

正在播放裁剪动画



12-41

animation

正在播放百叶窗动画



animation

正在播放马赛克动画



12-43

animation

正在播放淡入淡出动画



12-44

animation

正在播放平移动画



12-45

```

public class YingjiActivity extends AppCompatActivity implements
    AnimatorListener, AnimationListener, OnClickListener {
    private RelativeLayout rl_yingji;
    private TextView tv_anim_title;
    private LayoutParams mParams;
    private ImageView view1, view4, view5, view6;
    private ShutterView view2;
    private MosaicView view3;
    private int[] mImageArray = {
        R.drawable.bdg01, R.drawable.bdg02, R.drawable.bdg03, R.drawable.bdg04, R.drawable.bdg05,
        R.drawable.bdg06, R.drawable.bdg07, R.drawable.bdg08, R.drawable.bdg09, R.drawable.bdg10
    };
    private ObjectAnimator anim1, anim2, anim3, anim4;
    private Animation translateAnim, setAnim;
    private int mDuration = 5000;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_yingji);
        rl_yingji = (RelativeLayout) findViewById(R.id.rl_yingji);
        tv_anim_title = (TextView) findViewById(R.id.tv_anim_title);
        playYingji();
    }

    private void initView() {
        mParams = new LayoutParams(LayoutParams.MATCH_PARENT, LayoutParams.MATCH_PARENT);
        view1 = new ImageView(this); //灰度动画和裁剪动画
        view1.setLayoutParams(mParams);
        view1.setImageResource(mImageArray[0]);
        view1.setScaleType(ScaleType.FIT_START);
        view1.setAlpha(0f); //百叶窗动画
        view2 = new ShutterView(this);
        view2.setLayoutParams(mParams);
        view2.setImageBitmap(BitmapFactory.decodeResource(getResources(), mImageArray[1]));
        view2.setMode(PorterDuff.Mode.DST_OUT);
        view3 = new MosaicView(this); //马赛克动画
        view3.setLayoutParams(mParams);
        view3.setImageBitmap(BitmapFactory.decodeResource(getResources(), mImageArray[2]));
        view3.setMode(PorterDuff.Mode.DST_OUT);
        view3.setRatio(-5);
        view4 = new ImageView(this); //淡入淡出动画
    }
}

```

```

view4.setLayoutParams(mParams);
view4.setImageResource(mImageArray[3]);
view4.setScaleType(ScaleType.FIT_START);
view5 = new ImageView(this); //平移动画
view5.setLayoutParams(mParams);
view5.setImageResource(mImageArray[5]);
view5.setScaleType(ScaleType.FIT_START);
view6 = new ImageView(this); //集合动画
view6.setLayoutParams(mParams);
view6.setImageResource(mImageArray[6]);
view6.setScaleType(ScaleType.FIT_START);
}

private void playYingji() {
    rl_yingji.removeAllViews();
    initView();
    rl_yingji.addView(view1);
    anim1 = ObjectAnimator.ofFloat(view1, "alpha", 0f, 1f);
    anim1.setDuration(mDuration);
    anim1.addListener(this);
    anim1.start();
}

@Override
public void onAnimationStart(Animator animation) {
    if (animation.equals(anim1)) {
        tv_anim_title.setText("正在播放灰度动画");
    } else if (animation.equals(anim2)) {
        tv_anim_title.setText("正在播放裁剪动画");
    } else if (animation.equals(anim3)) {
        tv_anim_title.setText("正在播放百叶窗动画");
    } else if (animation.equals(anim4)) {
        tv_anim_title.setText("正在播放马赛克动画");
    }
}

@TargetApi(Build.VERSION_CODES.JELLY_BEAN_MR2)
@Override
public void onAnimationEnd(Animator animation) {
    if (animation.equals(anim1)) {
        rl_yingji.addView(view2, 0);
        Bitmap bitmap = BitmapFactory.decodeResource(getResources(), mImageArray[0]);
        int width = view1.getWidth();
        int height = bitmap.getHeight()*width/bitmap.getWidth();

```

```

        anim2 = ObjectAnimator.ofObject(view1, "clipBounds",
            new RectEvaluator(), new Rect(0,0,width,height),
            new Rect(width/2,height/2,width/2,height/2));
        anim2.setDuration(mDuration);
        anim2.addListener(this);
        anim2.start();
    } else if (animation.equals(anim2)) {
        rl_yingji.removeView(view1);
        rl_yingji.addView(view3, 0);
        anim3 = ObjectAnimator.ofInt(view2, "ratio", 0, 100);
        anim3.setDuration(mDuration);
        anim3.addListener(this);
        anim3.start();
    } else if (animation.equals(anim3)) {
        rl_yingji.removeView(view2);
        rl_yingji.addView(view4, 0);
        int offset = 5;
        view3.setOffset(offset);
        anim4 = ObjectAnimator.ofInt(view3, "ratio", 0-offset, 101+offset);
        anim4.setDuration(mDuration);
        anim4.addListener(this);
        anim4.start();
    } else if (animation.equals(anim4)) {
        rl_yingji.removeView(view3);
        Drawable[] drawableArray = { getResources().getDrawable(mImageArray[3]),
            getResources().getDrawable(mImageArray[4]) };
        TransitionDrawable td_fade = new TransitionDrawable(drawableArray);
        td_fade.setCrossFadeEnabled(false);
        view4.setImageDrawable(td_fade);
        int delay = mDuration;
        td_fade.startTransition(delay);
        tv_anim_title.setText("正在播放淡入淡出动画");
        mHandler.postDelayed(mTransitionEnd, delay);
    }
}

private Handler mHandler = new Handler();
private Runnable mTransitionEnd = new Runnable() {
    @Override
    public void run() {
        rl_yingji.addView(view5, 0);
        translateAnim = new TranslateAnimation(0f, -view4.getWidth(), 0f, 0f);
        translateAnim.setDuration(mDuration);
    }
}

```

```

        translateAnim.setFillAfter(true);
        view4.startAnimation(translateAnim);
        translateAnim.setAnimationListener(YingjiActivity.this);
    }
};

private void startSetAnim() {
    Animation alpha = new AlphaAnimation(1.0f, 0.1f);
    alpha.setDuration(mDuration);
    alpha.setFillAfter(true);
    Animation translate = new TranslateAnimation(1.0f, -200f, 1.0f, 1.0f);
    translate.setDuration(mDuration);
    translate.setFillAfter(true);
    Animation scale = new ScaleAnimation(1.0f, 1.0f, 1.0f, 0.5f);
    scale.setDuration(mDuration);
    scale.setFillAfter(true);
    Animation rotate = new RotateAnimation(0f, 360f, Animation.RELATIVE_TO_SELF,
        0.5f, Animation.RELATIVE_TO_SELF, 0.5f);
    rotate.setDuration(mDuration);
    rotate.setFillAfter(true);
    setAnim = new AnimationSet(true);
    ((AnimationSet) setAnim).addAnimation(alpha);
    ((AnimationSet) setAnim).addAnimation(translate);
    ((AnimationSet) setAnim).addAnimation(scale);
    ((AnimationSet) setAnim).addAnimation(rotate);
    setAnim.setFillAfter(true);
    view5.startAnimation(setAnim);
    setAnim.setAnimationListener(this);
}

@Override
public void onAnimationCancel(Animator animation) {
}

@Override
public void onAnimationRepeat(Animator animation) {
}

@Override
public void onAnimationStart(Animation animation) {
    if (animation.equals(translateAnim)) {
        tv_anim_title.setText("正在播放平移动画");
    } else if (animation.equals(setAnim)) {
        tv_anim_title.setText("正在播放集合动画");
    }
}

```



```
}  
  
@Override  
public void onAnimationEnd(Animation animation) {  
    if (animation.equals(translateAnim)) {  
        rl_yingji.removeView(view4);  
        rl_yingji.addView(view6, 0);  
        startSetAnim();  
    } else if (animation.equals(setAnim)) {  
        rl_yingji.removeView(view5);  
        tv_anim_title.setText("动感影集播放结束，谢谢观看");  
        view6.setOnClickListener(this);  
    }  
}  
  
@Override  
public void onAnimationRepeat(Animation animation) {  
}  
  
@Override  
public void onClick(View v) {  
    if (v.equals(view6)) {  
        playYingji();  
    }  
}  
}
```

12.6

App(GIF)()(
Scroller)“QQ”App

4

(1)

(2)

(3)

(4)

第13章 多媒体



AppContentProvider“——”

13.1

13.1.1 Gallery

GalleryAndroidGalleryDeprecation()

HorizontalScrollViewViewPagerGalleryGallery

Gallery

setSpacingXMLspacing

setUnselectedAlphaXMLunselectedAlpha0.0~1.00.01.0

setAdapter

getSelectedItemId

setSelection

setOnItemClickListener

GalleryImageViewFrameLayoutImageViewGallery

GalleryImageView

GalleryImageView

```

public class GalleryActivity extends AppCompatActivity implements OnItemClickListener {
    private ImageView iv_gallery;
    private Gallery gl_gallery;
    private int[] mImageRes = { R.drawable.scene1, R.drawable.scene2, R.drawable.scene3,
        R.drawable.scene4, R.drawable.scene5, R.drawable.scene6 };

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_gallery);
        iv_gallery = (ImageView) findViewById(R.id.iv_gallery);
        iv_gallery.setImageResource(mImageRes[0]);
        int dip_pad = Utils.dip2px(this, 20);
        gl_gallery = (Gallery) findViewById(R.id.gl_gallery);
        gl_gallery.setPadding(0, dip_pad, 0, dip_pad);
        gl_gallery.setSpacing(dip_pad);
        gl_gallery.setUnselectedAlpha(0.5f);
        gl_gallery.setAdapter(new GalleryAdapter(this, mImageRes));
        gl_gallery.setOnItemClickListener(this);
    }

    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
        iv_gallery.setImageResource(mImageRes[position]);
    }
}

```

Gallery13-113-213-113-2

media



13-1



13-2

GalleryRecyclerView
RecyclerViewImageView

```

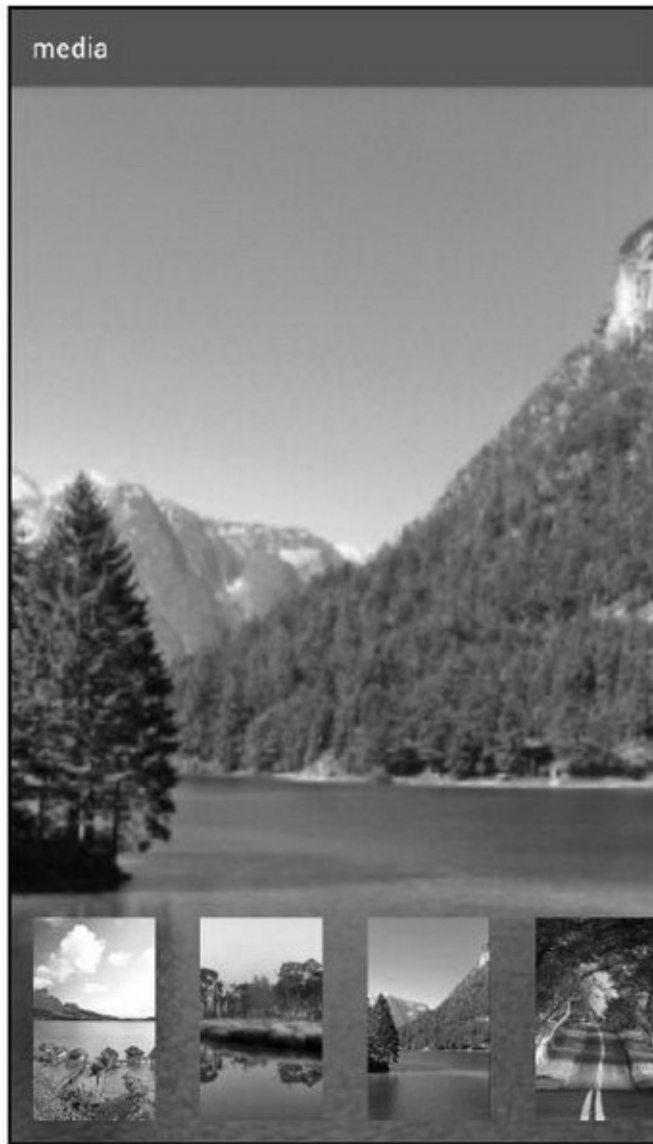
public class RecyclerViewActivity extends AppCompatActivity implements OnItemClickListener {
    private ImageView iv_photo;
    private RecyclerView rv_photo;
    private int[] mImageRes = { R.drawable.scene1, R.drawable.scene2, R.drawable.scene3,
        R.drawable.scene4, R.drawable.scene5, R.drawable.scene6 };

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_recycler_view);
        iv_photo = (ImageView) findViewById(R.id.iv_photo);
        iv_photo.setImageResource(mImageRes[0]);
        rv_photo = (RecyclerView) findViewById(R.id.rv_photo);
        LinearLayoutManager manager = new LinearLayoutManager(this);
        manager.setOrientation(LinearLayout.HORIZONTAL);
        rv_photo.setLayoutManager(manager);
        PhotoAdapter adapter = new PhotoAdapter(this, mImageRes);
        adapter.setOnItemClickListener(this);
        rv_photo.setAdapter(adapter);
        rv_photo.setItemAnimator(new DefaultItemAnimator());
        rv_photo.addItemDecoration(new SpacesItemDecoration(20));
    }

    @Override
    public void onItemClick(View view, int position) {
        iv_photo.setImageResource(mImageRes[position]);
        rv_photo.scrollToPosition(position);
    }
}

```

RecyclerView13-313-413-3313-444



13-33



13-44

13.1.2 ImageSwitcher Gallery

ImageView ImageSwitcher ImageSwitcher

ImageSwitcher ViewAnimator TextSwitcher 11 ViewFlipper

ViewAnimator

ImageSwitcher

setFactory ViewFactory makeView ImageSwitcher

ImageView

setImageResourceID setImageDrawable setImageURI

setImageDrawableDrawable

setImageURIURI

setInAnimation

setOutAnimation

1112setInAnimationsetOutAnimationGallery

GestureDetector

ImageSwitcher

```
public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
    is_switcher.setInAnimation(AnimationUtils.loadAnimation(this, R.anim.fade_in));
    is_switcher.setOutAnimation(AnimationUtils.loadAnimation(this, R.anim.fade_out));
    is_switcher.setImageResource(mImageRes[position]);
}

public class ViewFactoryImpl implements ViewFactory {
    @Override
    public View makeView() {
        ImageView iv = new ImageView(ImageSwitcherActivity.this);
        iv.setBackgroundColor(0xFFFFFFFF);
        iv.setScaleType(ScaleType.FIT_XY);
        iv.setLayoutParams(new ImageSwitcher.LayoutParams(
            LayoutParams.MATCH_PARENT, LayoutParams.MATCH_PARENT));
        return iv;
    }
}

@Override
```

```
public boolean onTouch(View v, MotionEvent event) {
    mGesture.onTouchEvent(event);
    return true;
}

@Override
public void gotoNext() {
    is_switcher.setInAnimation(AnimationUtils.loadAnimation(this, R.anim.push_left_in));
    is_switcher.setOutAnimation(AnimationUtils.loadAnimation(this, R.anim.push_left_out));
    int next_pos = (int) (gl_switcher.getSelectedItemId() + 1);
    if (next_pos >= mImageRes.length) {
        next_pos = 0;
    }
    is_switcher.setImageResource(mImageRes[next_pos]);
    gl_switcher.setSelection(next_pos);
}

@Override
public void gotoPre() {
    is_switcher.setInAnimation(AnimationUtils.loadAnimation(this, R.anim.push_right_in));
    is_switcher.setOutAnimation(AnimationUtils.loadAnimation(this, R.anim.push_right_out));
    int pre_pos = (int) (gl_switcher.getSelectedItemId() - 1);
    if (pre_pos < 0) {
        pre_pos = mImageRes.length - 1;
    }
    is_switcher.setImageResource(mImageRes[pre_pos]);
    gl_switcher.setSelection(pre_pos);
}
```

13-513-613-513-6

media



13-5



13-6

13.1.3

GalleryImageSwitcher

CardViewAndroid5.0CardViewCardView

FrameLayout

CardView5.0Androidbuild.gradledependenciescardview

compile 'com.android.support:cardview-v7:25.0.0'

CardView13-1

13-1CardView

CardView 的属性名称	CardView 的设置方法	说明
cardBackgroundColor	setCardBackgroundColor	设置卡片边框的背景颜色
cardCornerRadius	setRadius	设置卡片边框的圆角半径
cardElevation	setCardElevation	设置卡片边缘的阴影高程，即宽度
contentPadding	setContentPadding	设置卡片边框的间隔

CardView

(1)cardviewappres-auto“app:”“android:”

(2)marginwrap_content

CardView

```

public class CardViewActivity extends AppCompatActivity {
    private CardView cv_card;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_card_view);
        cv_card = (CardView) findViewById(R.id.cv_card);
        initCardSpinner();
    }

    private void initCardSpinner() {
        ArrayAdapter<String> cardAdapter = new ArrayAdapter<String>(this,
            R.layout.item_select, cardArray);
        Spinner sp_card = (Spinner) findViewById(R.id.sp_card);
        sp_card.setPrompt("请选择卡片视图类型");
        sp_card.setAdapter(cardAdapter);
        sp_card.setOnItemSelectedListener(new CardSelectedListener());
        sp_card.setSelection(0);
    }

    private String[] cardArray={"圆角与阴影均为 3", "圆角与阴影均为 6", "圆角与阴影均为 10",
        "圆角与阴影均为 15", "圆角与阴影均为 20", "圆角与阴影均为 30", "圆角与阴影均为 50"};
    private int[] radiusArray = {3, 6, 10, 15, 20, 30, 50};
    class CardSelectedListener implements OnItemSelectedListener {
        public void onItemSelected(AdapterView<?> arg0, View arg1, int arg2, long arg3) {
            int interval = radiusArray[arg2];
            cv_card.setRadius(interval);
            cv_card.setCardElevation(interval);
            MarginLayoutParams params = (MarginLayoutParams) cv_card.getLayoutParams();
            params.setMargins(interval, interval, interval, interval);
            cv_card.setLayoutParams(params);
        }

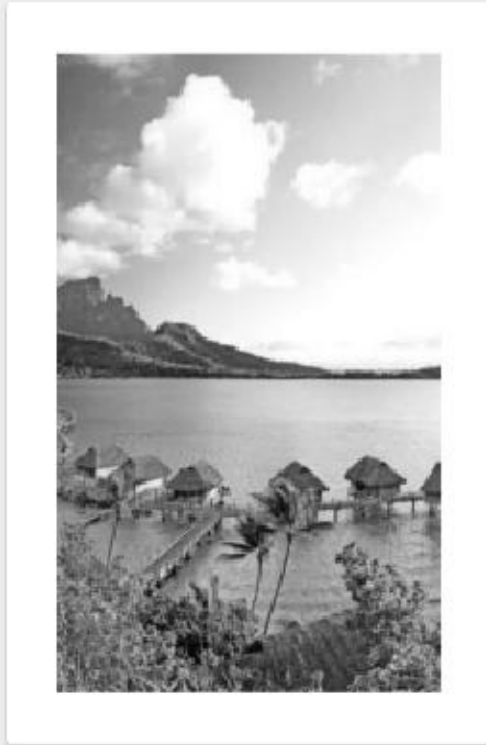
        public void onNothingSelected(AdapterView<?> arg0) {
        }
    }
}

```

13-713-813-7613-815

media

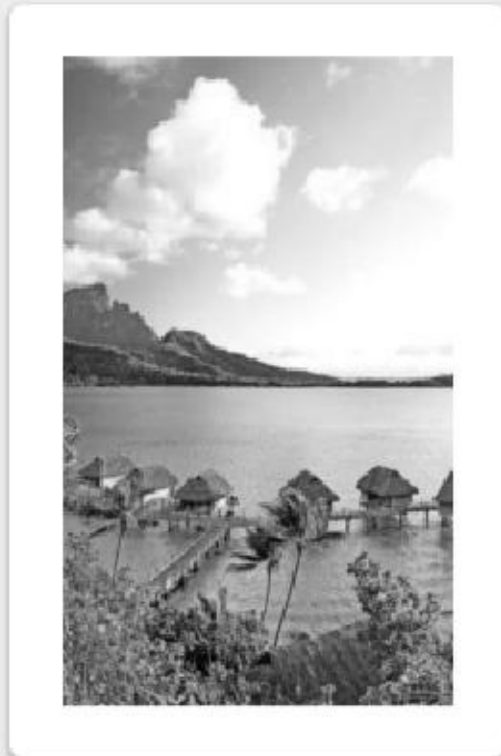
卡片视图样式: 圆角与阴影均为6 ▼



13-76

media

卡片视图样式: 圆角与阴影均为15 ▾



13-815

Android5.0——PaletteApp

Palette

```
Palette5.0build.gradledependenciespalette
```

```
compile 'com.android.support:palette-v7:25.0.0'
```

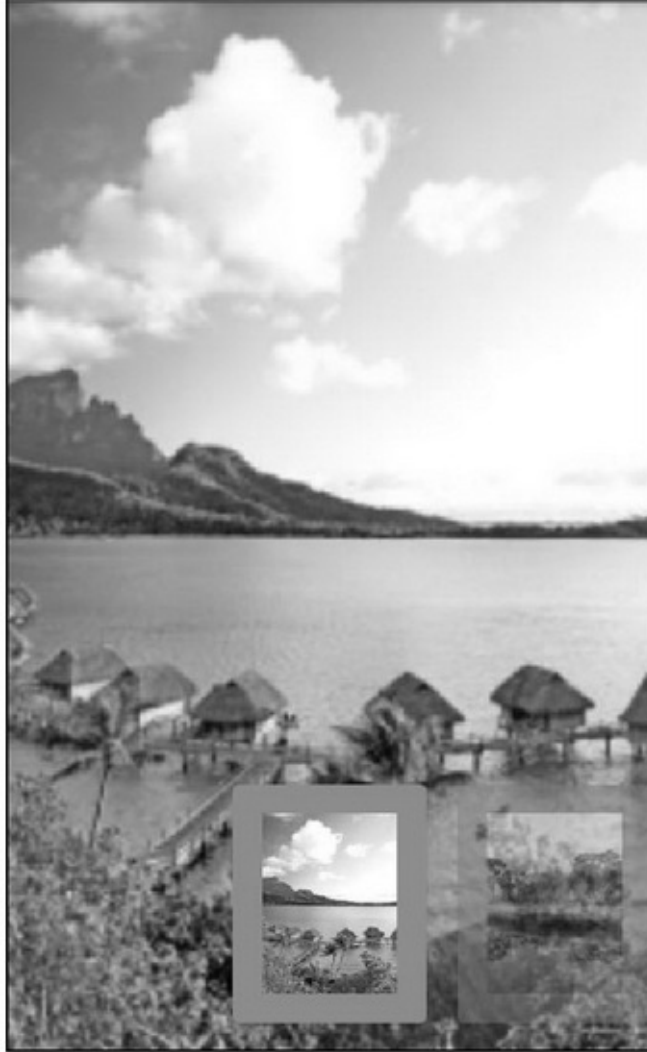
Palette

from

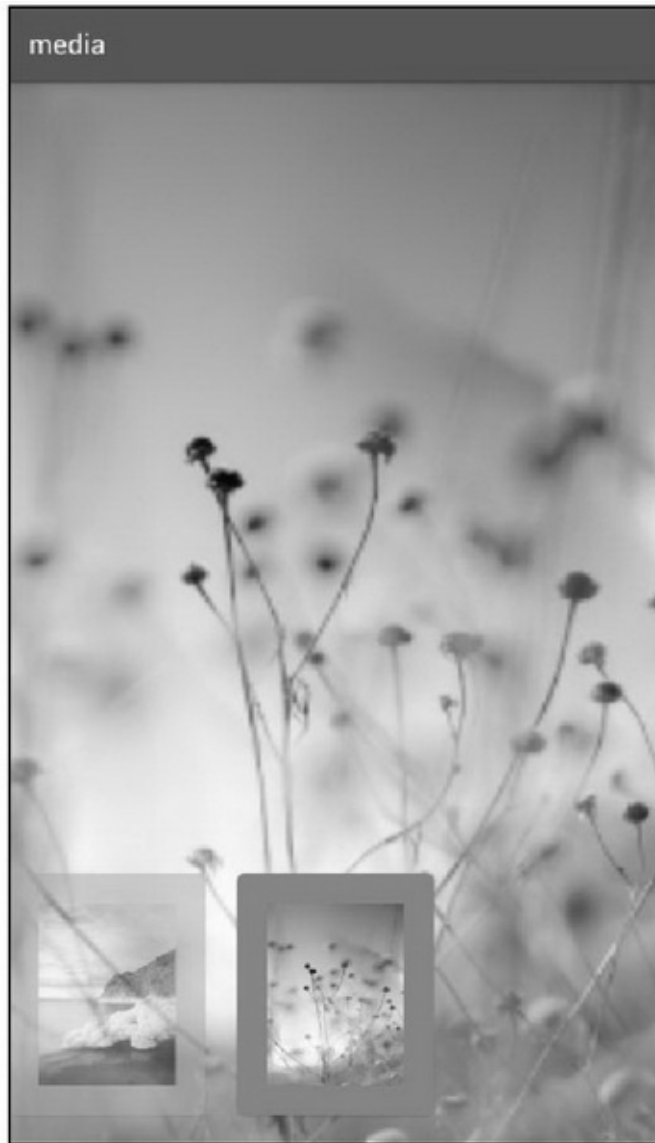
```
Builder.generateAndroidPaletteAsyncListener
```

onGenerated

media



13-9



13-10

App“”

13.2

“_____”

13.2.1 VideoView 9MediaPlayerSurfaceViewSurfaceView MediaPlayerAndroidVideoViewSurfaceViewMediaPlayer

VideoView

setVideoPath

setMediaController

setOnPreparedListenerOnPreparedListeneronPrepared

setOnCompletionListenerOnCompletionListeneronCompletion

setOnErrorListenerOnErrorListeneronError

setOnInfoListenerOnInfoListeneronInfo/

requestFocusstart

start

pause

resume

suspend

seekTo

getDuration

getCurrentPositiongetDuration

isPlaying

getBufferPercentage01

VideoViewSeekBar

VideoViewSeekBargetDurationgetCurrentPositionseekTo

SeekBarVideoViewSeekBar

VideoViewSeekBar9

```

public class VideoViewActivity extends AppCompatActivity implements
    OnClickListener, FileSelectCallbacks, OnSeekBarChangeListener {
    private VideoView vv_play;
    private SeekBar sb_play;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_video_view);
        findViewById(R.id.btn_open).setOnClickListener(this);
        vv_play = (VideoView) findViewById(R.id.vv_play);
        sb_play = (SeekBar) findViewById(R.id.sb_play);
        sb_play.setOnSeekBarChangeListener(this);
        sb_play.setEnabled(false);
    }

    @Override
    public void onClick(View v) {
        if (v.getId() == R.id.btn_open) {
            FileSelectFragment.show(this, new String[] { "mp4" }, null);
        }
    }

    @Override
    public void onConfirmSelect(String absolutePath, String fileName, Map<String, Object> map_param) {
        String file_path = absolutePath + "/" + fileName;
        vv_play.setVideoPath(file_path);
        vv_play.requestFocus();
        vv_play.start();
        sb_play.setEnabled(true);
        mHandler.post(mRefresh);
    }

    @Override
    public boolean isValid(String absolutePath, String fileName, Map<String, Object> map_param) {
        return true;
    }

    private Handler mHandler = new Handler();
    private Runnable mRefresh = new Runnable() {
        @Override
        public void run() {
            sb_play.setProgress(100 * vv_play.getCurrentPosition() / vv_play.getDuration());
            mHandler.postDelayed(this, 500);
        }
    }
}

```

```

};

@Override
public void onProgressChanged(SeekBar seekBar, int progress, boolean fromUser) {
}

@Override
public void onStartTrackingTouch(SeekBar seekBar) {
}

@Override
public void onStopTrackingTouch(SeekBar seekBar) {
    int pos = seekBar.getProgress() * vv_play.getDuration() / 100;
    vv_play.seekTo(pos);
}
}

```





13-12

13.2.2MediaController

(1)

(2)

(3)

```
VideoViewsetMediaControllerMediaControllerWindows
```

```
MediaController
```

```
setMediaPlayerVideoViewsetAnchorView
```

```
setAnchorViewVideoViewsetMediaPlayer
```

```
show
```

hide

isShowing

setPrevNextListeners(OnClickListener)

VideoViewSurfaceViewMediaControllerFrameLayout

MediaControllerVideoViewVideoViewVideoView

MediaControllerVideoViewVideoView

1.VideoView

setVideoPathsetMediaControllerstartsetMediaPlayer

```
public void onConfirmSelect(String absolutePath, String fileName, Map<String, Object> map_param) {  
    String file_path = absolutePath + "/" + fileName;  
    vv_play.setVideoPath(file_path);  
    vv_play.requestFocus();  
    //媒体控制条代码开始  
    MediaController mc_play = new MediaController(this);  
    vv_play.setMediaController(mc_play);  
    mc_play.setMediaPlayer(vv_play);  
    //媒体控制条代码结束  
    vv_play.start();  
}
```

2.VideoView

setMediaPlayersetAnchorViewVideoView

VideoView

```
public void onConfirmSelect(String absolutePath, String fileName, Map<String, Object> map_param) {  
    String file_path = absolutePath + "/" + fileName;  
    VideoView vv_play = new VideoView(this);  
    vv_play.setVideoPath(file_path);  
    vv_play.requestFocus();  
    MediaController mc_play = new MediaController(this);  
    mc_play.setAnchorView(vv_play);  
    mc_play.setKeepScreenOn(true);  
    vv_play.setMediaController(mc_play);  
    ll_play.addView(vv_play);  
    vv_play.start();  
}
```

13-1313-1413-1313-14

media

打开视频文件





13-14

13.2.3——

VideoViewMediaController

- (1)
- (2)
- (3)
- (4)
- (5)

(6)

VideoViewMediaController

1.VideoView

MovieView

(1)onMeasure

(2)onTouch

(3)onKeyDown

(4)

2.MediaController

VideoController

(1)

(2)

(3)

MovieView

(1)

(2)App

(3)()

activityandroid:themeandroid:windowFullscreen

android:windowNoTitleAppandroid:windowContentOverlay

```
<style name="FullScreenTheme" parent="AppBaseTheme">
    <item name="android:windowFullscreen">true</item>
    <item name="android:windowNoTitle">true</item>
    <item name="android:windowContentOverlay">@null</item>
</style>
```

3activityandroid:configChangesAppActivityonDestroy

onCreate(3Activity)configChanges

onConfigurationChanged13-2

configChanges 属性的取值	说明
touchscreen	触摸屏发生改变，一般不会发生
keyboard	键盘发生改变，例如使用了外部键盘
keyboardHidden	软键盘弹出或隐藏
navigation	导航发生改变，一般不会发生
screenLayout	屏幕的显示发生改变，例如使用了外部显示器
fontScale	字体比例发生改变，例如在系统设置中调整默认字体
orientation	屏幕发生横屏与竖屏的切换
screenSize	屏幕大小发生改变

screenOrientationsensor13-3

13-3

screenOrientation 属性的取值	说明
portrait	只允许垂直方向
landscape	只允许水平方向
sensor	由传感器控制方向
unspecified	默认值，由系统选择方向
user	使用用户当前首选的方向
fullSensor	显示的 4 个方向由传感器决定，即 4 个方向都允许倒转

activity

```
<activity
    android:name=".MoviePlayerActivity"
    android:configChanges="orientation|keyboardHidden|screenSize"
    android:screenOrientation="sensor"
    android:theme="@style/FullScreenTheme" >
</activity>
```

MovieView

```

//支持以下功能：自动全屏、调节音量、收缩控制栏、设置背景
@TargetApi(Build.VERSION_CODES.JELLY_BEAN) //setBackground 需要
public class MovieView extends VideoView implements OnTouchListener, OnKeyListener, VolumeAdjustListener {
    private static final String TAG = "MovieView";
    private Context mContext;
    private int screenWidth, screenHeight;
    private int videoWidth, videoHeight;
    private int realWidth, realHeight;
    private int mXpos, mYpos, mOffset;
    // 自动隐藏顶部和底部 View 的时间
    public static final int HIDE_TIME = 5000;
    private View mTopView;
    private View mBottomView;
    private AudioManager mAudioMgr;
    private VolumeDialog dialog;
    private Handler mHandler = new Handler();

    public MovieView(Context context) {
        this(context, null);
    }

    public MovieView(Context context, AttributeSet attrs) {
        this(context, attrs, 0);
    }

    public MovieView(Context context, AttributeSet attrs, int defStyle) {
        super(context, attrs, defStyle);
        mContext = context;
        screenWidth = DisplayUtil.getSreenWidth(mContext);
        screenHeight = DisplayUtil.getSreenHeight(mContext);
        mOffset = Utils.dip2px(mContext, 10);
        mAudioMgr = (AudioManager) mContext.getSystemService(Context.AUDIO_SERVICE);
    }

    @Override
    protected void onMeasure(int widthMeasureSpec, int heightMeasureSpec) {
        int width = getDefaultSize(realWidth, widthMeasureSpec);

```



```

int height = getDefaultSize(realHeight, heightMeasureSpec);
if (realWidth > 0 && realHeight > 0) {
    if (realWidth * height > width * realHeight) {
        height = width * realHeight / realWidth;
    } else if (realWidth * height < width * realHeight) {
        width = height * realWidth / realHeight;
    }
}
setMeasuredDimension(width, height);
}

@Override
public boolean onTouch(View v, MotionEvent event) {
    switch (event.getAction()) {
        case MotionEvent.ACTION_DOWN:
            mXpos = (int) event.getX();
            mYpos = (int) event.getY();
            break;
        case MotionEvent.ACTION_UP:
            if (Math.abs(event.getX()-mXpos) < mOffset && Math.abs(event.getY()-mYpos) < mOffset) {
                showOrHide();
            }
            break;
        default:
            break;
    }
    return true;
}

public void prepare(View topTiew, View bottomView) {
    mTopView = topTiew;
    mBottomView = bottomView;
    setBackgroundResource(R.drawable.video_bg1);
}

public void begin(MediaPlayer mp) {
    setBackground(null);
    if (mp != null) {
        videoWidth = mp.getVideoWidth();
        videoHeight = mp.getVideoHeight();
    }
    realWidth = videoWidth;
    realHeight = videoHeight;
}

```

```

        start();
    }

    public void end(MediaPlayer mp) {
        setBackgroundResource(R.drawable.video_bg3);
        realWidth = screenWidth;
        realHeight = screenHeight;
    }

    public void showOrHide() {
        if (mTopView==null || mBottomView==null) {
            return;
        }
        if (mTopView.getVisibility() == View.VISIBLE) {
            mTopView.clearAnimation();
            Animation animTop = AnimationUtils.loadAnimation(mContext, R.anim.leave_from_top);
            animTop.setAnimationListener(new AnimationImp() {
                @Override
                public void onAnimationEnd(Animation animation) {
                    mTopView.setVisibility(View.GONE);
                }
            });
            mTopView.startAnimation(animTop);
            mBottomView.clearAnimation();
            Animation animBottom = AnimationUtils.loadAnimation(mContext, R.anim.leave_from_bottom);
            animBottom.setAnimationListener(new AnimationImp() {
                @Override
                public void onAnimationEnd(Animation animation) {
                    mBottomView.setVisibility(View.GONE);
                }
            });
            mBottomView.startAnimation(animBottom);
        } else {
            mTopView.setVisibility(View.VISIBLE);
            mTopView.clearAnimation();
            Animation animTop = AnimationUtils.loadAnimation(mContext, R.anim.entry_from_top);
            mTopView.startAnimation(animTop);
            mBottomView.setVisibility(View.VISIBLE);
            mBottomView.clearAnimation();
            Animation animBottom = AnimationUtils.loadAnimation(mContext, R.anim.entry_from_bottom);
            mBottomView.startAnimation(animBottom);
            mHandler.removeCallbacks(mHide);
            mHandler.postDelayed(mHide, HIDE_TIME);
        }
    }

```

```

    }
}

private Runnable mHide = new Runnable() {
    @Override
    public void run() {
        showOrHide();
    }
};

private class AnimationImp implements AnimationListener {
    @Override
    public void onAnimationEnd(Animation animation) {
    }

    @Override
    public void onAnimationRepeat(Animation animation) {
    }

    @Override
    public void onAnimationStart(Animation animation) {
    }
}

@Override
public boolean onKeyDown(View v, int keyCode, KeyEvent event) {
    if (keyCode == KeyEvent.KEYCODE_VOLUME_UP) {
        showVolumeDialog(AudioManager.ADJUST_RAISE);
        return true;
    } else if (keyCode == KeyEvent.KEYCODE_VOLUME_DOWN) {
        showVolumeDialog(AudioManager.ADJUST_LOWER);
        return true;
    }
    return false;
}

private void showVolumeDialog(int direction) {
    if (dialog == null || dialog.isShowing() != true) {
        dialog = new VolumeDialog(mContext);
        dialog.setVolumeAdjustListener(this);
        dialog.show();
    }
    dialog.adjustVolume(direction, true);
}

```

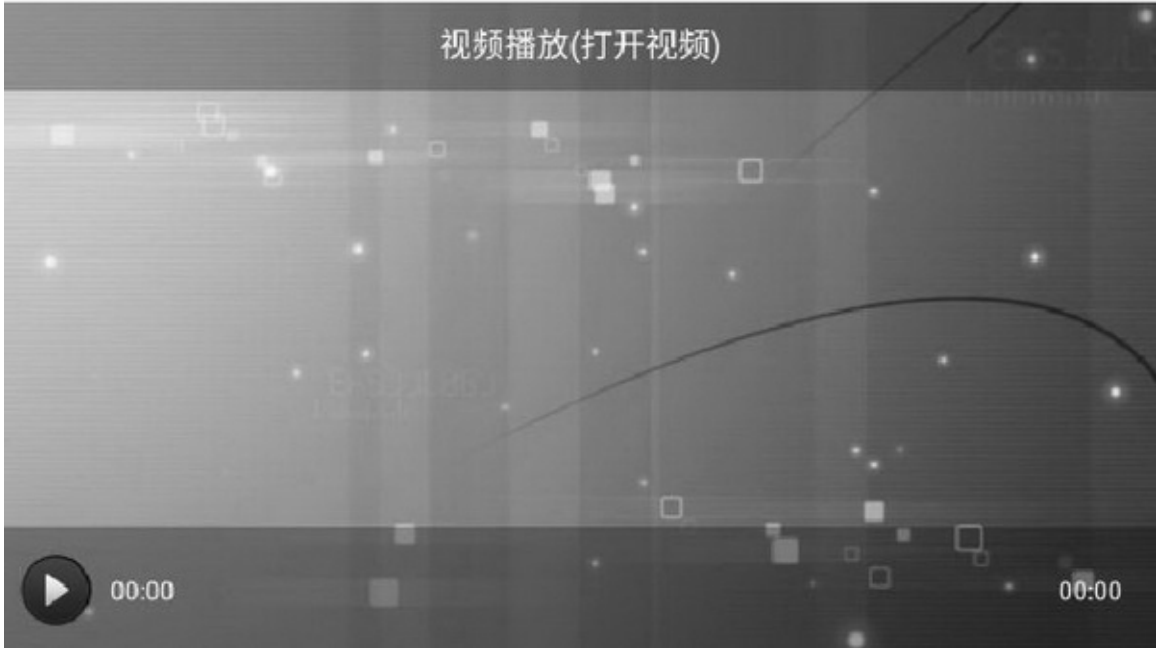
```

        onVolumeAdjust(mAudioMgr.getStreamVolume(AudioManager.STREAM_MUSIC));
    }

    @Override
    public void onVolumeAdjust(int volume) {
    }
}

```

13-15~13-1813-1513-16

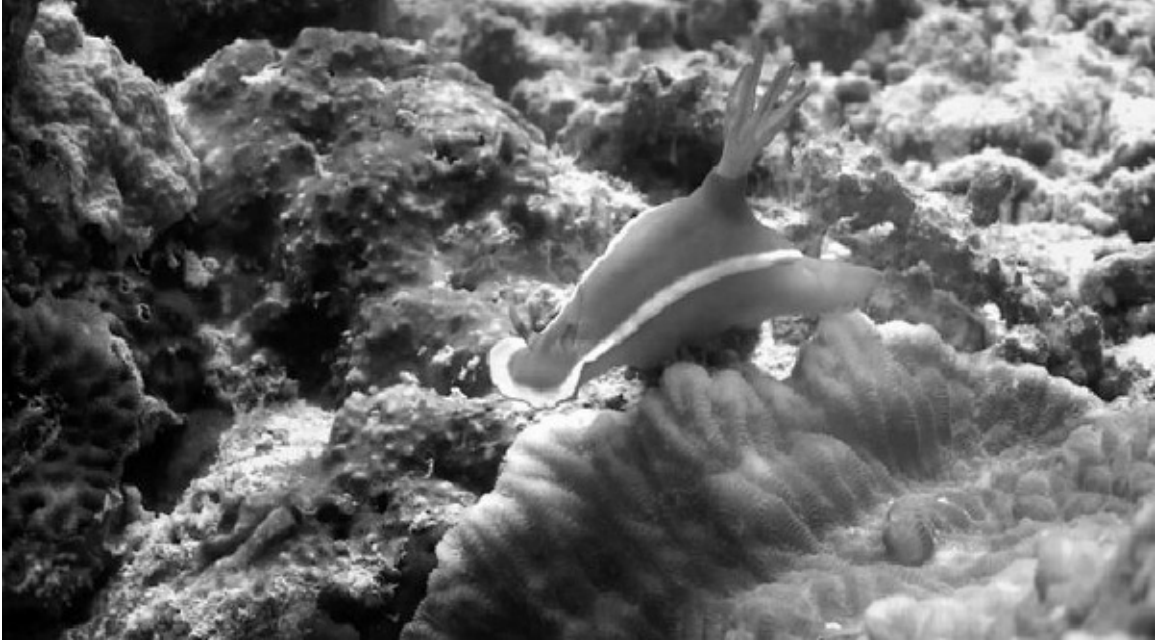


13-15

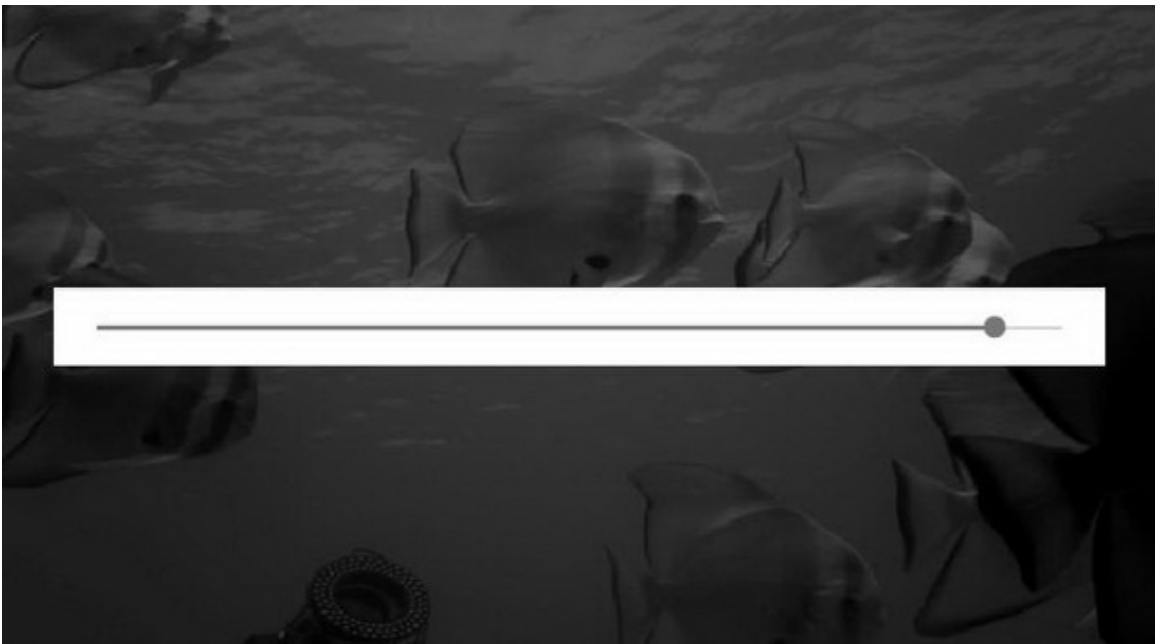


13-16

13-1713-189



13-17



13-18

“”

13.3

AndroidContentProvider

13.3.1 ContentProvider AndroidActivityBroadcastService

ContentProvider3356ContentProvider

ContentProviderAppSQLiteContentProvider

ContentProvider

onCreate

query

insert

update

delete

getType

SQLiteContentProviderSQLiteOpenHelperSQLiteDatabase

SQLiteContentProviderSQLiteContentProvider

ContentProvider

```
public class UserInfoProvider extends ContentProvider {
    public static final int USER_INFO = 1;
    public static final UriMatcher uriMatcher = new UriMatcher(UriMatcher.NO_MATCH);
    // 这里的 AUTHORITIES 取值必须与 AndroidManifest.xml 里的 android:authorities 保持一致
    static {
        uriMatcher.addURI(UserInfoContent.AUTHORITIES, "/user", USER_INFO);
    }
    private UserDBHelper userDB;

    //删除数据
    @Override
    public int delete(Uri uri, String selection, String[] selectionArgs) {
        int count = 0;
```

```

        if (uriMatcher.match(uri) == USER_INFO) {
            SQLiteDatabase db = userDB.getWritableDatabase();
            count = db.delete(UserInfoContent.TABLE_NAME, selection, selectionArgs);
            db.close();
        }
        return count;
    }

    //插入数据
    @Override
    public Uri insert(Uri uri, ContentValues values) {
        Uri newUri = uri;
        if (uriMatcher.match(uri) == USER_INFO) {
            SQLiteDatabase db = userDB.getWritableDatabase();
            // 向指定的表插入数据, 得到返回的 id
            long rowId = db.insert(UserInfoContent.TABLE_NAME, null, values);
            if (rowId > 0) { // 判断插入是否执行成功
                // 如果添加成功, 就利用新添加的 id 和新生成的地址
                newUri = ContentUris.withAppendedId(UserInfoContent.CONTENT_URI, rowId);
                // 通知监听器, 数据已经改变
                getContext().getContentResolver().notifyChange(newUri, null);
            }
            db.close();
        }
        return uri;
    }

    //创建 ContentProvider 时调用的回调函数
    @Override
    public boolean onCreate() {
        userDB = UserDBHelper.getInstance(getContext(), 1);
        return false;
    }

    //查询数据库
    @Override
    public Cursor query(Uri uri, String[] projection, String selection, String[] selectionArgs, String sortOrder) {
        Cursor cursor = null;
        if (uriMatcher.match(uri) == USER_INFO) {
            SQLiteDatabase db = userDB.getReadableDatabase();
            // 执行查询
            cursor = db.query(UserInfoContent.TABLE_NAME,
                projection, selection, selectionArgs, null, null, sortOrder);
        }
    }

```

```

        // 设置监听
        cursor.setNotificationUri(getContext().getContentResolver(), uri);
    }
    return cursor;
}

//数据访问类型，暂未实现
@Override
public String getType(Uri uri) {
    throw new UnsupportedOperationException("Not yet implemented");
}

//更新数据，暂未实现
@Override
public int update(Uri uri, ContentValues values, String selection, String[] selectionArgs) {
    throw new UnsupportedOperationException("Not yet implemented");
}
}

```

AndroidManifest.xml

```

<provider
    android:name=".provider.UserInfoProvider"
    android:authorities="com.example.media.provider.UserInfoProvider"
    android:enabled="true"
    android:exported="true" />

```

AppApp

13.3.2ContentResolver ContentProviderAppApp

ContentResolverAppContentResolverActivity

ContentResolverContentProviderqueryinsertupdatedelete
 getContentResolver
 getTypequeryCursorSQLite

query()

uriUri

projection

selection

selectionArgs

sortOrder

UserInfoProviderContentResolver

```
private void addUser(ContentResolver resolver, UserInfo user) {
    ContentValues name = new ContentValues();
    name.put("name", user.name);
    name.put("age", user.age);
    name.put("height", user.height);
    name.put("weight", user.weight);
    name.put("married", false);
    name.put("update_time", Utils.getNowDateTime());
    // insert 的第一个参数指明了要访问的数据源
    resolver.insert(UserInfoContent.CONTENT_URI, name);
}
```

ContentResolver

```
private String readAllUser(ContentResolver resolver) {
    ArrayList<UserInfo> userArray = new ArrayList<UserInfo>();
    // query 的第一个参数指明了要访问的数据源
    Cursor cursor = resolver.query(UserInfoContent.CONTENT_URI, null, null, null, null);
    while (cursor.moveToNext()) {
        UserInfo user = new UserInfo();
        user.name = cursor.getString(cursor.getColumnIndex(UserInfoContent.USER_NAME));
        user.age = cursor.getInt(cursor.getColumnIndex(UserInfoContent.USER_AGE));
        user.height = cursor.getInt(cursor.getColumnIndex(UserInfoContent.USER_HEIGHT));
        user.weight = cursor.getFloat(cursor.getColumnIndex(UserInfoContent.USER_WEIGHT));
        userArray.add(user);
    }
    cursor.close();
    String result = "";
    for (int i=0; i<userArray.size(); i++) {
        UserInfo user = userArray.get(i);
        result = String.format("%s%s    年龄%d    身高%d    体重%f\n", result,
            user.name, user.age, user.height, user.weight);
    }
    return result;
}
```

13-19ContentResolver113-20

media

阿四

25

165

50

添加用户信息

当前共找到1位用户信息



13-20

AppContentProviderApp

ContentResolvewrwrap_contacts

```

public static void addContacts(ContentResolver resolver, Contact contact) {
    // 往 raw_contacts 中添加数据, 并获取添加的 id 号
    Uri raw_uri = Uri.parse("content://com.android.contacts/raw_contacts");
    ContentValues values = new ContentValues();
    long contactId = ContentUris.parseId(resolver.insert(raw_uri, values));

    // 往 data 中添加数据 (要根据前面获取的 id 号)
    Uri uri = Uri.parse("content://com.android.contacts/data");
    ContentValues name = new ContentValues();
    name.put("raw_contact_id", contactId);
    name.put("mimetype", "vnd.android.cursor.item/name");
    name.put("data2", contact.name);
    resolver.insert(uri, name);

    ContentValues phone = new ContentValues();
    phone.put("raw_contact_id", contactId);
    phone.put("mimetype", "vnd.android.cursor.item/phone_v2");
    phone.put("data2", "2");
    phone.put("data1", contact.phone);
    resolver.insert(uri, phone);

    ContentValues email = new ContentValues();
    email.put("raw_contact_id", contactId);
    email.put("mimetype", "vnd.android.cursor.item/email_v2");
    email.put("data2", "2");
    email.put("data1", contact.email);
    resolver.insert(uri, email);
}

```

4insert4insertinsert

AndroidContentProviderOperation

ContentProviderOperation

ContentProviderOperation

```

public static void addFullContacts(ContentResolver resolver, Contact contact) {
    Uri raw_uri = Uri.parse("content://com.android.contacts/raw_contacts");
    Uri uri = Uri.parse("content://com.android.contacts/data");
    ContentProviderOperation op_main = ContentProviderOperation
        .newInsert(raw_uri).withValue("account_name", null).build();
    ContentProviderOperation op_name = ContentProviderOperation
        .newInsert(uri).withValueBackReference("raw_contact_id", 0)
        .withValue("mimetype", "vnd.android.cursor.item/name")
        .withValue("data2", contact.name).build();
    ContentProviderOperation op_phone = ContentProviderOperation
        .newInsert(uri).withValueBackReference("raw_contact_id", 0)
        .withValue("mimetype", "vnd.android.cursor.item/phone_v2")
        .withValue("data2", "2").withValue("data1", contact.phone).build();
    ContentProviderOperation op_email = ContentProviderOperation
        .newInsert(uri).withValueBackReference("raw_contact_id", 0)
        .withValue("mimetype", "vnd.android.cursor.item/email_v2")
        .withValue("data2", "2").withValue("data1", contact.email).build();
    ArrayList<ContentProviderOperation> operations = new ArrayList<ContentProviderOperation>();
    operations.add(op_main);
    operations.add(op_name);
    operations.add(op_phone);
    operations.add(op_email);
    try {
        resolver.applyBatch("com.android.contacts", operations);
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

13-2113-2213-2115713-22158

media

阿四

15960238696

aaa@163.com

添加联系人

当前共找到157位联系人

media

阿四

15960238696

aaa@163.com

添加联系人

当前共找到158位联系人

13-22

13.3.3 ContentObserver ContentResolver

AppAppContentObserver

ContentObserverContentResolverContentResolver

registerContentObserver

unregisterContentObserver

notifyChange

6“(1810086)10086

ContentObserver

```
private Handler mHandler = new Handler();
private SmsGetObserver mObserver;
private static Uri mSmsUri;
private static String[] mSmsColumn;
```

```
@TargetApi(Build.VERSION_CODES.KITKAT)
```

```
private void initSmsObserver() {
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.KITKAT) {
        mSmsUri = Telephony.Sms.Inbox.CONTENT_URI;
        mSmsColumn = new String[] {
            Telephony.Sms.ADDRESS, Telephony.Sms.BODY, Telephony.Sms.DATE };
    } else {
        mSmsUri = Uri.parse("content://sms/inbox");
        mSmsColumn = new String[] { "address", "body", "date" };
    }
    mObserver = new SmsGetObserver(this, mHandler);
    getContentResolver().registerContentObserver(mSmsUri, true, mObserver);
}

@Override
protected void onDestroy() {
    getContentResolver().unregisterContentObserver(mObserver);
    super.onDestroy();
};

private static class SmsGetObserver extends ContentObserver {
    private Context mContext;
    public SmsGetObserver(Context context, Handler handler) {
        super(handler);
        mContext = context;
    }

    @Override
    public void onChange(boolean selfChange) {
        String sender = "";
        String content = "";
        String selection = String.format("address='10086' and date > %d",
            System.currentTimeMillis() - 1000 * 60 * 60);
        Cursor cursor = mContext.getContentResolver().query(
            mSmsUri, mSmsColumn, selection, null, "date desc");
        while(cursor.moveToNext()){
            sender = cursor.getString(0);
            content = cursor.getString(1);
            break;
        }
        cursor.close();
        if (pd != null && pd.isShowing() == true) {
            pd.dismiss();
        }
    }
}
```



```

//回调短信监听方法
mCheckResult = String.format("发送号码 : %s\n 短信内容 : %s", sender, content);
Log.d(TAG, "result="+mCheckResult);
String flow = String.format("流量校准结果如下 : \n\t 总流量为 : %s\n\t 已使用 : %s\n\t
剩余 : %s",
        findFlow(content, "总流量为", "MB"),
        findFlow(content, "已使用", "MB"), findFlow(content, "剩余", "MB"));
tv_check_flow.setText(flow);
super.onChange(selfChange);
    }
}

private static String findFlow(String sms, String begin, String end) {
    int begin_pos = sms.indexOf(begin);
    if (begin_pos < 0) {
        return "未获取";
    }
    String sub_sms = sms.substring(begin_pos);
    int end_pos = sub_sms.indexOf(end);
    if (end_pos < 0) {
        return "未获取";
    }
    String flow_desc = sub_sms.substring(begin.length(), end_pos+end.length());
    return flow_desc;
}
}

```

13-2313-2413-2313-24App

10086

中国移动10086

您好！您上月结转至本月的国内流量为599.89MB。截止2016年12月06日08时10分，您办理的套餐内含移动数据总流量为1GB175.89MB，已使用32.87MB，剩余1GB143.02MB（其中您已使用32.87MB，副号1（155555100000）已使用0.00MB）。其中国内流量剩余1GB143.02MB（含国内结转流量567.20MB）。回复"18*月份#"查询相应月份移动数据流量使用情况；回复"CXJZLL"查询流量不清零套餐列表。更多流量信息，可通过...//

请输入内容



删除

退出

media

进行流量校准

流量校准结果如下：

总流量为：1GB175.89MB

已使用：32.87MB

剩余：1GB143.02MB

13-24

ContentURIURI13-4

13-4URI

内容名称	URI 常量名	实际路径
联系人	ContactsContract.Contacts.CONTENT_URI	content://com.android.contacts/ contacts
联系人电话	ContactsContract.CommonDataKinds. Phone.CONTENT_URI	content://com.android.contacts/data/ phones
联系人邮箱	ContactsContract.CommonDataKinds. Email.CONTENT_URI	content://com.android.contacts/data/ emails
SIM 卡联系人		content://icc/adn
短信	Telephony.Sms.CONTENT_URI	content://sms
彩信	Telephony.Mms.CONTENT_URI	content://mms
通话记录	CallLog.Calls.CONTENT_URI	content://call_log/calls
收件箱	Telephony.Sms.Inbox.CONTENT_URI (短信相关的 URI)	content://sms/inbox
已发送	Telephony.Sms.Sent.CONTENT_URI (短信相关的 URI)	content://sms/sent
草稿箱	Telephony.Sms.Draft.CONTENT_URI (短信相关的 URI)	content://sms/draft
发件箱	Telephony.Sms.Outbox.CONTENT_URI (短信相关的 URI)	content://sms/outbox
发送失败	无	content://sms/failed
待发送列表	无。比如开启飞行模式后，该短信就在待发送列表里	content://sms/queued

13.4

“_____”

13.4.1

(QQ)3

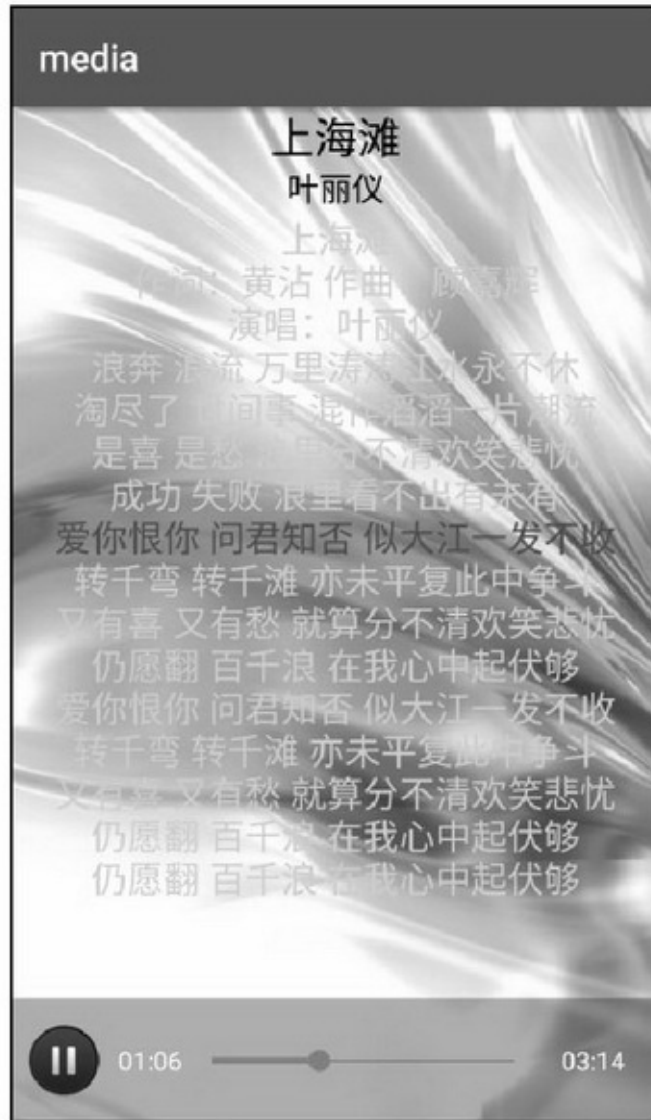
(1)

(2)

(3)

13-25“”13-26





13-26

3

“”App

LRC

App

(1)Service

(2)ApplicationApplication

(3)ContentResolver

MediaStore.Audio.Media.EXTERNAL_CONTENT_URI

(4)lrc

(5)Notify

(6)MediaPlayer

(7)KeyEvent

(8)Animation

(9)ListViewProgressBarSeekBar

ActivityMediaPlayerMediaController

13.4.2SpannableString setTextsetTextColor

setTextSizesetTextAppearance()

SpannableString3
AndroidSpannableString

步骤 01

SpannableString

步骤 02

SpannableStringsetSpan34(EditText

)13-5

步骤 03

setTextSpannableString

13-5

Spanned 类的范围标志	说明
SPAN_EXCLUSIVE_EXCLUSIVE	前后都不包括
SPAN_INCLUSIVE_EXCLUSIVE	前面包括, 后面不包括
SPAN_EXCLUSIVE_INCLUSIVE	前面不包括, 后面包括
SPAN_INCLUSIVE_INCLUSIVE	前后都包括

android.text.style30813-6

13-6

可变字符串的显示风格类	说明
RelativeSizeSpan	设置文字大小。1.0 表示正常大小，0.5 表示缩小到原来的一半，2.0 表示放大到原来的两倍
StyleSpan	设置文字字体。字体风格的取值说明见表 13-7
ForegroundColorSpan	设置文字的颜色
BackgroundColorSpan	设置文字的背景色
UnderlineSpan	给文字加下划线
StrikethroughSpan	给文字加删除线
ImageSpan	把文字替换为图片
URLSpan	给文字添加超链接

13-7

Typeface 类的字体风格	说明
NORMAL	正常字体
BOLD	加粗字体
ITALIC	倾斜字体
BOLD_ITALIC	既加粗又设置为斜体

SpannableString


```
public class SpannableActivity extends AppCompatActivity {
    private TextView tv_spannable;
    private String mText = "为人民服务";
    private String mKey = "人民";
    private int mBeginPos, mEndPos;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_spannable);
        tv_spannable = (TextView) findViewById(R.id.tv_spannable);
        tv_spannable.setText(mText);
        mBeginPos = mText.indexOf(mKey);
        mEndPos = mBeginPos + mKey.length();
        initSpannableSpinner();
    }

    private void initSpannableSpinner() {
        ArrayAdapter<String> spannableAdapter = new ArrayAdapter<String>(this,
            R.layout.item_select, spannableArray);
        Spinner sp_spannable = (Spinner) findViewById(R.id.sp_spannable);
    }
}
```

```

sp_spannable.setPrompt("可变字符串样式: ");
sp_spannable.setAdapter(spannableAdapter);
sp_spannable.setOnItemSelectedListener(new SpannableSelectedListener());
sp_spannable.setSelection(0);
}

private String[] spannableArray={
    "增大字号", "加粗字体", "前景红色", "背景绿色", "下划线", "表情图片" };
class SpannableSelectedListener implements OnItemSelectedListener {
    public void onItemSelected(AdapterView<?> arg0, View arg1, int arg2, long arg3) {
        SpannableString spanText = new SpannableString(mText);
        if (arg2 == 0) {
            spanText.setSpan(new RelativeSizeSpan(1.5f), mBeginPos, mEndPos,
                Spanned.SPAN_EXCLUSIVE_EXCLUSIVE);
        } else if (arg2 == 1) {
            spanText.setSpan(new StyleSpan(Typeface.BOLD), mBeginPos, mEndPos,
                Spanned.SPAN_EXCLUSIVE_EXCLUSIVE);
        } else if (arg2 == 2) {
            spanText.setSpan(new ForegroundColorSpan(Color.RED), mBeginPos,
                mEndPos, Spanned.SPAN_EXCLUSIVE_EXCLUSIVE);
        } else if (arg2 == 3) {
            spanText.setSpan(new BackgroundColorSpan(Color.GREEN), mBeginPos,
                mEndPos, Spanned.SPAN_EXCLUSIVE_EXCLUSIVE);
        } else if (arg2 == 4) {
            spanText.setSpan(new UnderlineSpan(), mBeginPos, mEndPos,
                Spanned.SPAN_EXCLUSIVE_EXCLUSIVE);
        } else if (arg2 == 5) {
            spanText.setSpan(new ImageSpan(SpannableActivity.this, R.drawable.people),
                mBeginPos, mEndPos, Spanned.SPAN_EXCLUSIVE_EXCLUSIVE);
        }
        tv_spannable.setText(spanText);
    }

    public void onNothingSelected(AdapterView<?> arg0) {
    }
}
}

```

SpannableString13-27~13-3213-2713-2813-2913-30
13-3113-32



13-27



13-28



13-29



13-30



13-31



13-32QQ“:)”AndroidSpannableString

13.4.3

5

(1)XMLres/anim(2)AndroidManifest.xmlSD

<!-- SD -->

<uses-permission

android:name="android.permission.WRITE_EXTERNAL_STORAGE" />

<uses-permission

android:name="android.permission.READ_EXTERNAL_STORAGE" /> <uses-permission

android:name="android.permission.MOUNT_UNMOUNT_FILESYSTEMS" />

(3)AndroidManifest.xmlapplicationandroid:name=".MainApplication"

service

<service android:name=".service.MusicService" android:enabled="true" />

(4)AndroidAndroid 4.4.24.4.2

(5)MP3MP3GBKUTF8

3

1.ContentResolver

MediaStore.Audio.Media.EXTERNAL_CONTENT_URIqueryURI

13-8

MediaStore 类的音频资源字段	说明
Audio.Media._ID	歌曲编号
Audio.Media.TITLE	歌曲的标题名称
Audio.Media.ALBUM	歌曲的专辑名称
Audio.Media.DURATION	歌曲的播放时间
Audio.Media.SIZE	歌曲文件的大小
Audio.Media.ARTIST	歌曲的演唱者
Audio.Media.DATA	歌曲文件的完整路径

2.LRC

LRCLRC

[offset:500]

[00:26.53]

[00:32.78]

[00:38.87]

[00:45.68]

[02:26.49][00:51.68]

[02:32.68][00:57.94]

offset“:.”

3.

TranslateAnimationAndroid

TranslateAnimationObjectAnimator

App13-3313-34

media

一剪梅

黑鸭子演唱组

专辑：《流行的经典》

真情在草原广阔

层层风雨不能阻隔

总有云开日出时候

万丈阳光照耀你我

真情像梅花开过

冷冷冰雪不能淹没

就在最冷

枝头绽放

看见春天走向你我

雪花飘飘北风啸啸

天地一片苍茫

一剪寒梅

傲立雪中

只为伊人飘香



00:38

爱我所爱无怨无悔

03:56

此情（此情）长留（长留）

media

一剪梅

黑鸭子演唱组

就在最冷
枝头绽放
看见春天走向你我
雪花飘飘北风萧萧
天地一片苍茫
一剪寒梅
傲立雪中
只为伊人飘香
爱我所爱无怨无悔
此情（此情）长留（长留）
心间.....

(Music)

真情像梅花开过
冷冷冰雪不能淹没
就在最冷
枝头绽放
看见春天走向你我
雪花飘飘北风萧萧
天地一片苍茫
一剪寒梅
傲立雪中



01:41

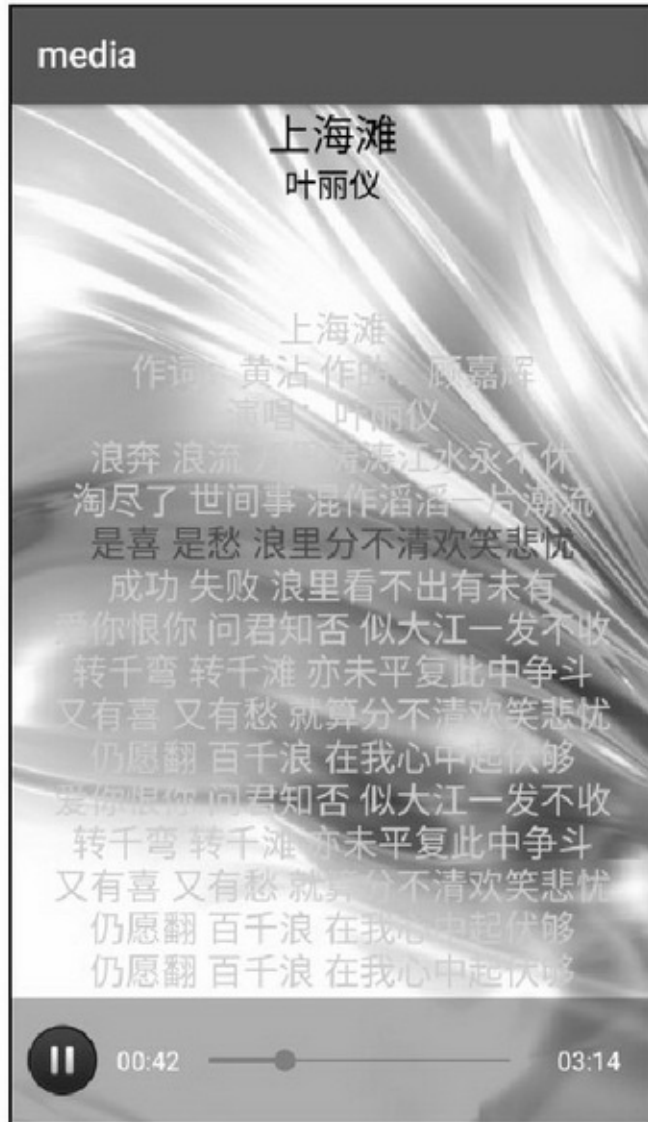
一剪寒梅

03:56

13-34

13-3513-36





13-36

13-37



13-37

```

@TargetApi(Build.VERSION_CODES.KITKAT)
public class MusicDetailActivity extends AppCompatActivity implements
    AnimatorListener, OnSeekBarChangeListener, VolumeAdjustListener {
    private static final String TAG = "MusicDetailActivity";
    private TextView tv_title;
    private TextView tv_artist;
    private TextView tv_music;
    private MusicInfo mMusic;
    private AudioController ac_play;
    private LyricsLoader mLoader;
    private ArrayList<LrcContent> mLrcList;
    private AudioManager mAudioMgr;
    private VolumeDialog dialog;
    private MainApplication app;
    private Handler mHandler = new Handler();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_music_detail);
        tv_title = (TextView) findViewById(R.id.tv_title);
        tv_artist = (TextView) findViewById(R.id.tv_artist);
        tv_music = (TextView) findViewById(R.id.tv_music);
        ac_play = (AudioController) findViewById(R.id.ac_play);
        ac_play.setOnSeekBarChangeListener(this);
        mMusic = getIntent().getParcelableExtra("music");
        tv_title.setText(mMusic.getTitle());
        tv_artist.setText(mMusic.getArtist());
        mLoader = LyricsLoader.getInstance(mMusic.getUrl());
        mLrcList = mLoader.getLrcList();
        mLineHeight = Math.round(MeasureUtil.getTextHeight("好", tv_music.getTextSize()));
        mAudioMgr = (AudioManager) getSystemService(Context.AUDIO_SERVICE);
        app = MainApplication.getInstance();
        playMusic();
    }

    @Override

```

```

protected void onDestroy() {
    super.onDestroy();
    mHandler.removeCallbacksAndMessages(null);
}

private int frequency = 8000;
private int channelConfig = AudioFormat.CHANNEL_IN_STEREO;
private int audioFormat = AudioFormat.ENCODING_PCM_16BIT;
// 播放歌曲
private void playMusic() {
    Log.d(TAG, "song="+mMusic.getTitle());
    if (Utils.getExtendName(mMusic.getUrl()).equals("pcm")) {
        ac_play.setVisibility(View.GONE);
        AudioPlayTask playTask = new AudioPlayTask();
        playTask.execute(mMusic.getUrl(), ""+frequency, ""+channelConfig, ""+audioFormat);
    } else {
        //下面处理歌词
        if (mLoader.getLrcList()!=null && mLrcList.size(>0) {
            mLrcStr = "";
            for (int i=0; i<mLrcList.size(); i++) {
                LrcContent item = mLrcList.get(i);
                mLrcStr = mLrcStr + item.getLrcStr() + "\n";
            }
            tv_music.setText(mLrcStr);
            tv_music.setAnimation(AnimationUtils.loadAnimation(this,R.anim.alpha_music));
        }
        //播放音乐
        if (app.mFilePath==null || !app.mFilePath.equals(mMusic.getUrl())) {
            Intent intent = new Intent(this, MusicService.class);
            intent.putExtra("is_play", true);
            intent.putExtra("music", mMusic);
            startService(intent);
            mHandler.postDelayed(mRefreshLrc, 150);
        } else {
            onMusicSeek(0, app.mMediaPlayer.getCurrentPosition());
        }
        mHandler.postDelayed(mRefreshCtrl, 100);
    }
}

//刷新进度条
private Runnable mRefreshCtrl = new Runnable() {
    @Override

```

```

        public void run() {
            ac_play.setCurrentTime(app.mMediaPlayer.getCurrentPosition(), 0);
            if (app.mMediaPlayer.getCurrentPosition() >= app.mMediaPlayer.getDuration()) {
                ac_play.setCurrentTime(0, 0);
            }
            mHandler.postDelayed(this, 500);
        }
};

private int mCount = 0;
private float mCurrentHeight = 0;
private float mLineHeight = 0;
//计算每行歌词的动画
private Runnable mRefreshLrc = new Runnable() {
    @Override
    public void run() {
        if (mLoader.getLrcList() == null || mLrcList.size() <= 0) {
            return;
        }
        int offset = mLrcList.get(mCount).getLrcTime()
            - ((mCount == 0) ? 0 : mLrcList.get(mCount - 1).getLrcTime()) - 50;
        if (offset <= 0) {
            return;
        }
        startAnimation(mCurrentHeight - mLineHeight, offset);
    }
};

private int mPrePos = -1, mNextPos = 0;
private String mLrcStr;
private ObjectAnimator animTranY;
//歌词滚动动画
public void startAnimation(float aimHeight, int offset) {
    animTranY = ObjectAnimator.ofFloat(tv_music, "translationY", mCurrentHeight, aimHeight);
    animTranY.setDuration(offset);
    animTranY.setRepeatCount(0);
    animTranY.addListener(this);
    animTranY.start();
    mCurrentHeight = aimHeight;
    if (app.mMediaPlayer.isPlaying() != true) {
        mHandler.postDelayed(new Runnable() {
            @Override
            public void run() {

```

```

        animTranY.pause();
    }
    }, offset+100);
}
}

@Override
public void onAnimationStart(Animator animation) {
}

@Override
public void onAnimationEnd(Animator animation) {
    if (mCount < mLrcList.size()) {
        mNextPos = mLrcStr.indexOf("n", mPrePos+1);
        SpannableString spanText = new SpannableString(mLrcStr);
        spanText.setSpan(new ForegroundColorSpan(Color.RED), mPrePos+1,
            mNextPos>0?mNextPos:mLrcStr.length()-1, Spanned.SPAN_EXCLUSIVE_
EXCLUSIVE);
        mCount++;
        tv_music.setText(spanText);
        if (mNextPos > 0 && mNextPos < mLrcStr.length()-1) {
            mPrePos = mLrcStr.indexOf("n", mNextPos);
            mHandler.postDelayed(mRefreshLrc, 50);
        }
    }
}

@Override
public void onAnimationCancel(Animator animation) {
}

@Override
public void onAnimationRepeat(Animator animation) {
}

//音乐控制条的拖动操作
@Override
public void onMusicSeek(int current, int seekto) {
    Log.d(TAG, "current="+current+", seekto="+seekto);
    if (animTranY != null) {
        animTranY.cancel();
    }
    mHandler.removeCallbacks(mRefreshLrc);
}

```

```

int i;
for (i=0; i<mLrcList.size(); i++) {
    LrcContent item = mLrcList.get(i);
    if (item.getLrcTime() > seekto) {
        break;
    }
}
mCount = i;
mPrePos = -1;
mNextPos = 0;
if (mCount > 0) {
    for (int j = 0; j < mCount; j++) {
        mNextPos = mLrcStr.indexOf("\n", mPrePos + 1);
        mPrePos = mLrcStr.indexOf("\n", mNextPos);
    }
}
startAnimation(-mLineHeight*i, 100);
}

@Override
public void onMusicPause() {
    animTranY.pause();
}

@Override
public void onMusicResume() {
    animTranY.resume();
}

//音量调节对话框
@Override
public boolean onKeyDown(int keyCode, KeyEvent event) {
    if (keyCode == KeyEvent.KEYCODE_VOLUME_UP) {
        showVolumeDialog(AudioManager.ADJUST_RAISE);
        return true;
    } else if (keyCode == KeyEvent.KEYCODE_VOLUME_DOWN) {
        showVolumeDialog(AudioManager.ADJUST_LOWER);
        return true;
    } else if (keyCode == KeyEvent.KEYCODE_BACK) {
        finish();
    }
}
return false;
}

```

```
private void showVolumeDialog(int direction) {
    if (dialog==null || dialog.isShowing()!=true) {
        dialog = new VolumeDialog(this);
        dialog.setVolumeAdjustListener(this);
        dialog.show();
    }
    dialog.adjustVolume(direction, true);
    onVolumeAdjust(mAudioMgr.getStreamVolume(AudioManager.STREAM_MUSIC));
}

@Override
public void onVolumeAdjust(int volume) {
}
}
```


13.5

App()("")ContentProvider(
)"—"App

5

(1)

(2)

(3)

(4)ContentProvider

(5)

第14章 融合技术



14.1

——WebAndroidiOSWindows

14.1.1 AssetManager Appres

(1)res/drawable

(2)res/values/strings.xml

(3)res/values/colors.xml

(4)res/values/integers.xml

(5)res/values/arrays.xml

(6)res/raw

resres

(1)Appres/values(key-value)

(2)JARSDKresjarres

(3)HTMLres

AndroidassetsAndroid Studioassetssrc/mainassets

assetsR.*.*——AssetManagerassets

ActivitygetAssetsAssetManagerAssetManager

list

openInputStreamAssetManager.ACCESS_STREAMING

close

assetsAssetManager

```
public static String getTxtFromAssets(Context context, String fileName) {
    String result = "";
    try {
        InputStream is = context.getAssets().open(fileName);
        int lenght = is.available();
        byte[] buffer = new byte[lenght];
        is.read(buffer);
        result = new String(buffer, "utf8");
    } catch (Exception e) {
        e.printStackTrace();
    }
    return result;
}

public static Bitmap getImgFromAssets(Context context, String fileName) {
    Bitmap bitmap = null;
    try {
        InputStream is = context.getAssets().open(fileName);
        bitmap = BitmapFactory.decodeStream(is);
    } catch (Exception e) {
        e.printStackTrace();
    }
    return bitmap;
}
```

14-114-214-1assets14-2assets

mixture

下面文字来源于资产文件file/libai.txt

望庐山瀑布
李白

日照香炉生紫烟，
遥看瀑布挂前川。
飞流直下三千尺，
疑是银河落九天。

mixture

下面图像来源于资产文件file/water.jpg



14-2

14.1.2 WebView assetshtmlWebViewWebView AndroidWebAppWeb

WebViewloadUrlfile: android_asset/assets

```
public class WebLocalActivity extends AppCompatActivity {
    private String mFilePath = "file:///android_asset/html/index.html";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_web_local);
        TextView tv_web_path = (TextView) findViewById(R.id.tv_web_path);
        WebView wv_assets_web = (WebView) findViewById(R.id.wv_assets_web);
        tv_web_path.setText("下面网页来源于资产文件"+mFilePath);
        wv_assets_web.loadUrl(mFilePath);
        wv_assets_web.setWebViewClient(new WebViewClient());
    }
}
```

mixture

下面网页来源于资产文件file:///android_asset/html/index.html



望庐山瀑布
李白

日照香炉生紫烟，
遥看瀑布挂前川。
飞流直下三千尺，
疑是银河落九天。

14-3

WebView14-3

App13UrlSpanURLURL

AppHTML

```

private void showUrlSpan() {
    SpannableString spanText = new SpannableString(mText);
    //调用 setMovementMethod 方法设置 LinkMovementMethod 后，点击超链接才有反应
    tv_spannable.setMovementMethod(LinkMovementMethod.getInstance());
    Spannable sp = (Spannable) Html.fromHtml("<a href=\"\">"+mKey+"</a>");
    CharSequence text = sp.toString();
    URLSpan[] urls = sp.getSpans(0, text.length(), URLSpan.class);
    for (URLSpan url : urls) {
        MyURLSpan myURLSpan = new MyURLSpan(url.getURL());
        spanText.setSpan(myURLSpan, mBeginPos, mEndPos, Spanned.SPAN_EXCLUSIVE_
EXCLUSIVE);
    }
    tv_spannable.setText(spanText);
}

private class MyURLSpan extends URLSpan {
    public MyURLSpan(String url) {
        super(url);
    }

    @Override
    public void onClick(View widget) {
        wv_spannable.setVisibility(View.VISIBLE);
        wv_spannable.loadUrl("http://blog.csdn.net/aqi00");
        wv_spannable.requestFocus();
        wv_spannable.setWebViewClient(new WebViewClient());
        return;
    }
}
}

```

14-4URLWeb14-5



14-4



14-5

14.1.3

WebViewloadUrl(setWebViewClient)WebView
loadUrlURLURLhttpfile
getSettingsWebSettings
addJavascriptInterfaceJavaScriptApp

setWebViewClientWebViewClientloadUrl
setWebChromeClientWebChromeClient
setDownloadListenerDownloadListener
loadDatatext/htmlbase64BASE64(null)URL
canGoBack
goBack
canGoForward
goForward
reload
stopLoading
4WebSettingsWebViewClientWebChromeClient

DownloadListener

1.WebSettings

WebSettingsWebViewgetSettingsWebSettingsWebSettings

setLoadsImagesAutomaticallyfalse
setDefaultTextEncodingNameUTF-8GBK
setJavaScriptEnabledJavaScript
setJavaScriptCanOpenWindowsAutomaticallyJavaScriptJS

window.open

setSupportZoom
setBuiltInZoomControls
setUseWideViewPort
setLoadWithOverviewMode
setLayoutAlgorithmLayoutAlgorithm.SINGLE_COLUMN

Android4.2.2

setAppCacheEnabledApp

setAppCachePathApp
setAllowFileAccessWebViewSD
setDatabaseEnabled
setDomStorageEnabled
setCacheMode14-1

14-1

WebSettings 类的缓存模式	说明
LOAD_CACHE_ELSE_NETWORK	优先使用缓存
LOAD_NO_CACHE	不使用缓存
LOAD_CACHE_ONLY	只使用缓存

2.WebViewClient

WebViewClientWebViewsetWebViewClient

onPageStartedProgressFialog

onPageFinished

onReceivedError

onReceivedSslErrorSSL

shouldOverrideUrlLoading

```
public boolean shouldOverrideUrlLoading(WebView view, String url) {  
    view.loadUrl(url);  
    return true;  
}
```

3.WebChromeClient

WebChromeClientAppWebViewsetWebChromeClient

WebChromeClient

onReceivedTitle

onProgressChanged

onJsAlertJSalert

onJsConfirmJSconfirm

onJsPromptJSprompt

onGeolocationPermissionsShowPrompt

```
public void onGeolocationPermissionsShowPrompt(String origin, Callback callback) {  
    callback.invoke(origin, true, false);  
    super.onGeolocationPermissionsShowPrompt(origin, callback);  
}
```

4.DownloadListener

DownloadListenerWebViewsetDownloadListenerDownloadListener

onDownloadStart

onDownloadStart

WebViewWebView

```
public class WebBrowserActivity extends AppCompatActivity implements OnClickListener {
    private final static String TAG = "WebBrowserActivity";
    private EditText et_web_url;
    private WebView wv_web;
    private ProgressDialog m_pd;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_web_browser);
        et_web_url = (EditText) findViewById(R.id.et_web_url);
        et_web_url.setText("news.qq.com/");
        wv_web = (WebView) findViewById(R.id.wv_web);
        findViewById(R.id.btn_web_go).setOnClickListener(this);
        findViewById(R.id.ib_back).setOnClickListener(this);
        findViewById(R.id.ib_forward).setOnClickListener(this);
        findViewById(R.id.ib_refresh).setOnClickListener(this);
        findViewById(R.id.ib_close).setOnClickListener(this);
        initWebViewSettings();
    }

    @SuppressWarnings("SetJavaScriptEnabled")
    private void initWebViewSettings() {
        WebSettings settings = wv_web.getSettings();
        settings.setLoadsImagesAutomatically(true);
        settings.setDefaultTextEncodingName("utf-8");
        settings.setJavaScriptEnabled(true);
        settings.setJavaScriptCanOpenWindowsAutomatically(false);
        settings.setSupportZoom(true);
        settings.setBuiltInZoomControls(true);
        settings.setUseWideViewPort(true);
    }
}
```

```

        settings.setLoadWithOverviewMode(true);
        settings.setLayoutAlgorithm(LayoutAlgorithm.SINGLE_COLUMN);
    }

    @Override
    public void onClick(View v) {
        if (v.getId() == R.id.btn_web_go) {
            InputMethodManager imm = (InputMethodManager) getSystemService(Context.INPUT_
METHOD_SERVICE);
            imm.hideSoftInputFromWindow(et_web_url.getWindowToken(), 0);
            String url = "http://" + et_web_url.getText().toString();
            Log.d(TAG, "url="+url);
            wv_web.loadUrl(url);
            wv_web.setWebViewClient(mWebViewClient);
            wv_web.setWebChromeClient(mWebChrome);
            wv_web.setDownloadListener(mDownloadListener);
        } else if (v.getId() == R.id.ib_back) {
            if (wv_web.canGoBack()) {
                wv_web.goBack();
            } else {
                Toast.makeText(this, "已经是最后一页了", Toast.LENGTH_SHORT).show();
            }
        } else if (v.getId() == R.id.ib_forward) {
            if (wv_web.canGoForward()) {
                wv_web.goForward();
            } else {
                Toast.makeText(this, "已经是最前一页了", Toast.LENGTH_SHORT).show();
            }
        } else if (v.getId() == R.id.ib_refresh) {
            //重新加载。停止加载用 stopLoading
            wv_web.reload();
        } else if (v.getId() == R.id.ib_close) {
            finish();
        }
    }

    @Override
    public void onBackPressed() {
        if (wv_web.canGoBack()) {
            wv_web.goBack();
            return;
        } else {
            finish();
        }
    }

```

```

    }
}

private WebViewClient mWebViewClient = new WebViewClient() {
    @Override
    public void onReceivedSslError(WebView view, android.webkit.SslErrorHandler handler,
        android.net.http.SslError error) {
        handler.proceed();
    };

    @Override
    public void onPageStarted(WebView view, String url, Bitmap favicon) {
        super.onPageStarted(view, url, favicon);
        Log.d(TAG, "onPageStarted:" + url);
        if (m_pd == null || m_pd.isShowing() == false) {
            m_pd = new ProgressDialog(WebBrowserActivity.this);
            m_pd.setTitle("稍等");
            m_pd.setMessage("页面加载中……");
            m_pd.setProgressStyle(ProgressDialog.STYLE_HORIZONTAL);
            m_pd.show();
        }
    }

    @Override
    public void onPageFinished(WebView view, String url) {
        super.onPageFinished(view, url);
        Log.d(TAG, "onPageFinished:" + url);
        if (m_pd != null && m_pd.isShowing() == true) {
            m_pd.dismiss();
        }
    }

    @Override
    public void onReceivedError(WebView view, int errorCode, String description, String failingUrl) {
        super.onReceivedError(view, errorCode, description, failingUrl);
        Log.d(TAG, "onReceivedError: url=" + failingUrl + ", errorCode=" + errorCode + ",
description=" + description);
        if (m_pd != null && m_pd.isShowing() == true) {
            m_pd.dismiss();
        }
        Toast.makeText(WebBrowserActivity.this, "页面加载失败，请稍候再试",
Toast.LENGTH_LONG).show();
    }
}

```

```

        @Override
        public boolean shouldOverrideUrlLoading(WebView view, String url) {
            view.loadUrl(url);
            return true;
        }
    };

    private WebChromeClient mWebChrome = new WebChromeClient() {
        @Override
        public void onProgressChanged(WebView view, int progress) {
            if (m_pd != null && m_pd.isShowing() == true) {
                m_pd.setProgress(progress);
            }
        }

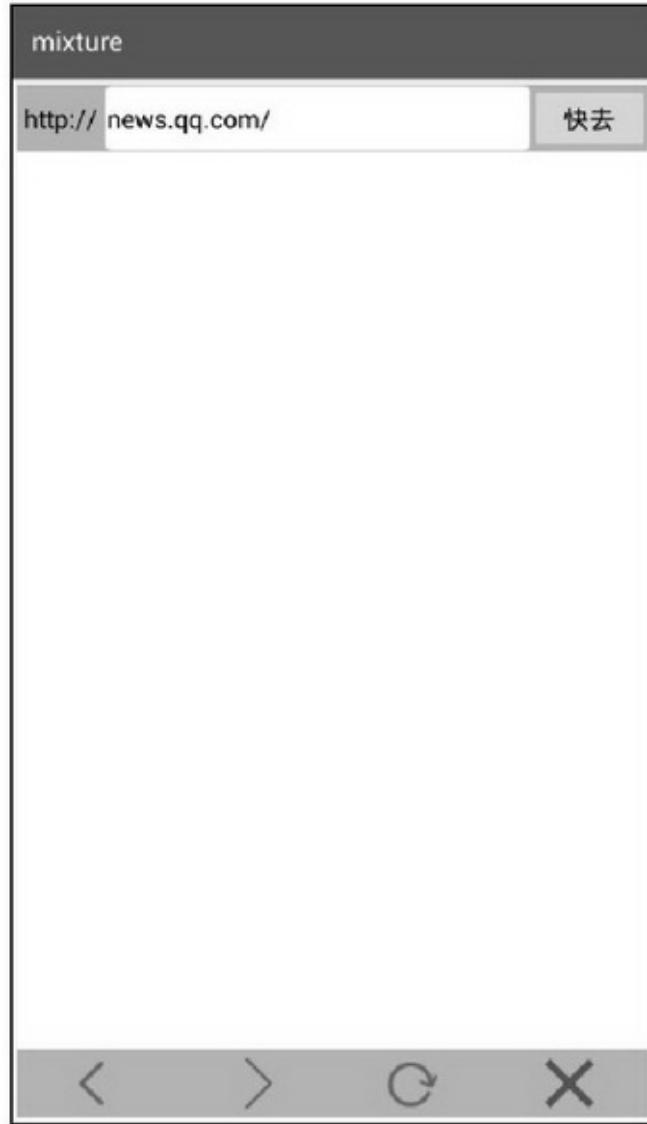
        @Override
        public void onGeolocationPermissionsShowPrompt(String origin, Callback callback) {
            callback.invoke(origin, true, false);
            super.onGeolocationPermissionsShowPrompt(origin, callback);
        }
    };

    private DownloadListener mDownloadListener = new DownloadListener() {
        @Override
        public void onDownloadStart(String url, String userAgent, String contentDisposition,
            String mimeType, long contentLength) {
            //此处操作文件下载
        }
    };
}

```

14-6~14-914-6()“”14-7

Web14-814-9



14-6



14-7





14.2JNI

—— JNIC/C++C/C++C/C++(AndroidiOS)
Android StudioNDKJNIJavaCJNI“JNI”

14.2.1NDK

Android Studio3JDKSDKNDK

(1)JDKJavaAppJavaJDK

(2)SDKAndroidAndroidAPIAppSDK

(3)NDKC/C++AppJNINDKAppJNINDK

NDKAppC/C++JavaJNIC/C++JNIAndroid StudioNDK

NDK

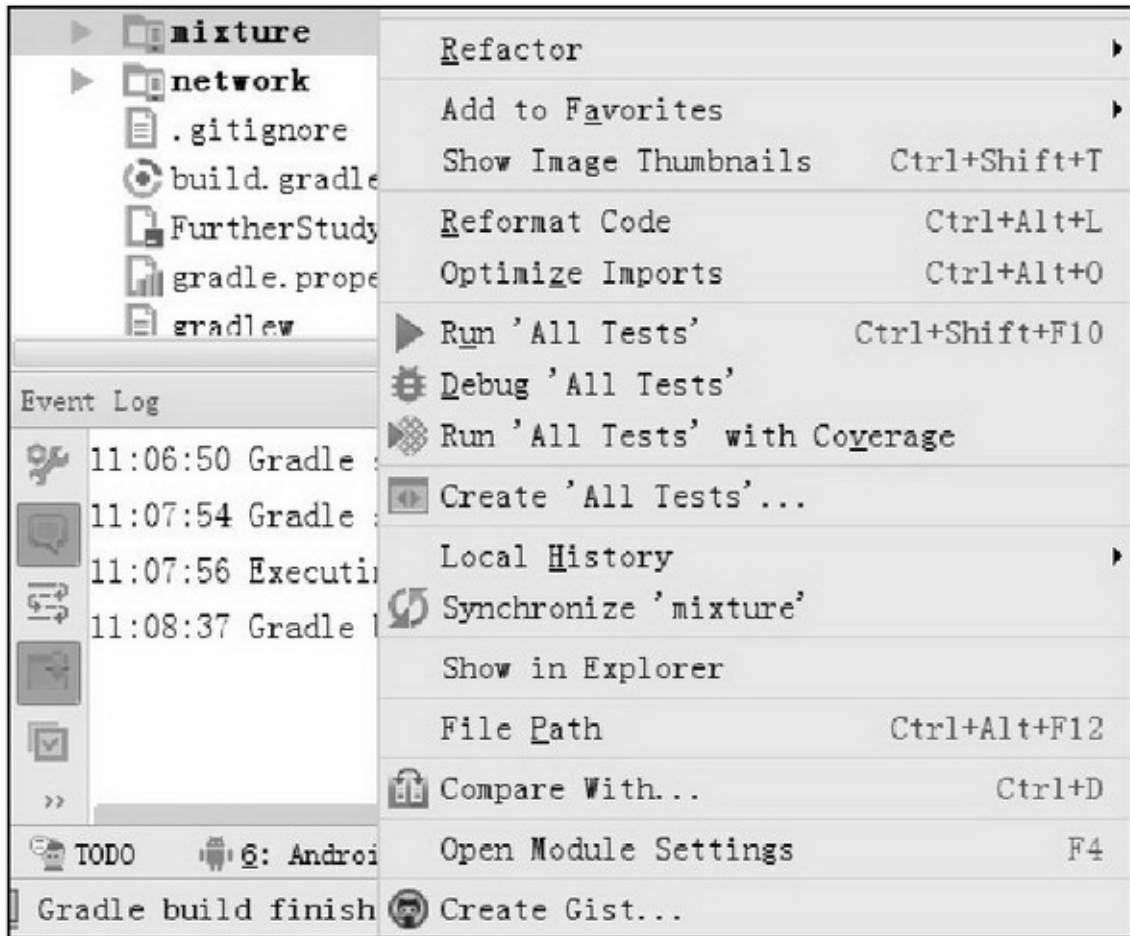
步骤 01 NDK

<https://developer.android.google.cn/ndk/downloads/index.html>NDK

D:\android-ndk-r13b

步骤 02 NDKNDK_ROOTD:\android-ndk-r13bPath

;%NDK_ROOT%



14-10

步骤 03

Open Module Settings 14-10 File → Project

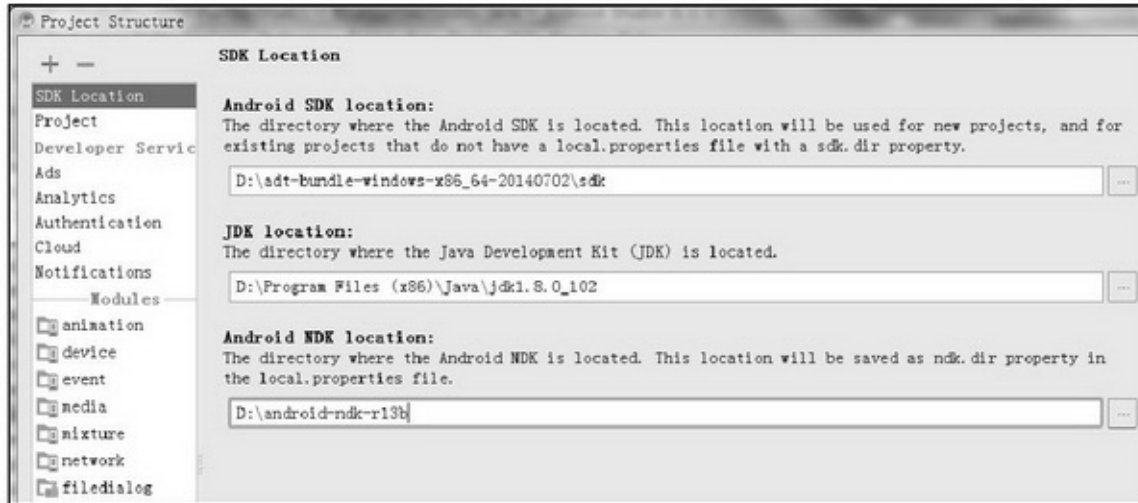
Structure

SDK Location → NDK Location NDKOK 14-11

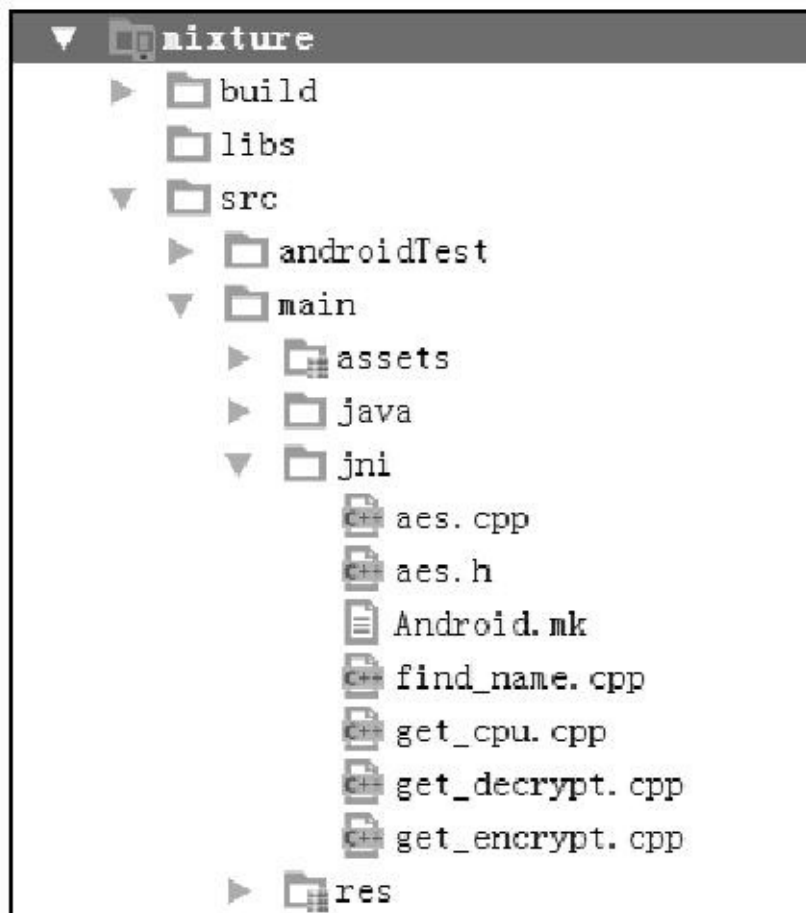
步骤 04

src/main/jni/hccppmkjni 14-12jni

javares



14-11NDK



14-12jni

步骤 05 build.gradledefaultConfig

```
ndk {
    moduleName "jni_mix"
    cFlags "-fexceptions"
    ldLibs "log", "z", "m"
    stl "stlport_static"
    abiFilters "armeabi", "armeabi-v7a", "x86"
}
```

Android Studio 2.2androidmk

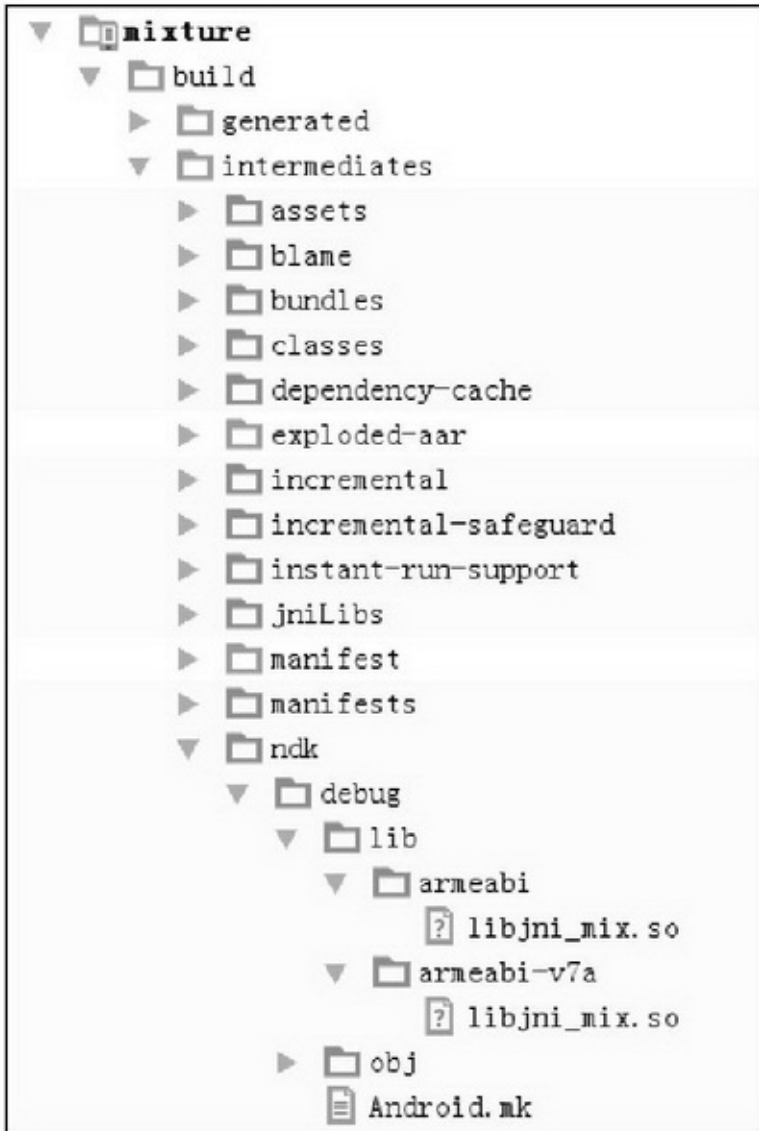
```
externalNativeBuild {
    ndkBuild {
        path file("src\\main\\jni\\Android.mk")
    }
}
```

步骤 06 gradle.propertiesandroid.useDeprecatedNdk=true

步骤 07 Build → Rebuild ProjectBuild → Make Module ***

C/C++“\build\intermediates\ndk\debug\lib”soso14-13

步骤 08 src/mainsojniLibssoso14-14jniLibsjni



14-13so



14-14jniLibs

步骤 09

AppApkAppso

14.2.2JNI

JNIJava Native InterfaceAPIJava(C/C++)JNIJavaAndroidJNI
NDKNDKC/C++(.a.so)JNIAppC/C++NDK
build.gradlemksoAppJavaJNI

JNI

步骤 01 NDKNDK

步骤 02 JNIActivityJNIJNI

```

public native String cpuFromJNI(int i1, float f1, double d1, boolean b1);
public native String unimplementedCpuFromJNI(int i1, float f1, double d1, boolean b1);
static {
    System.loadLibrary("jni_mix");
}

```

步骤 03 jnihccppC/C++C“Java__Activity_

”C

```

jstring Java_com_example_mixture_JniCpuActivity_cpuFromJNI( JNIEnv* env, jobject thiz, jint i1, jfloat f1,
jdouble d1, jboolean b1 )

```

步骤 04 build.gradleNDKjnimk(build.gradle externalNativeBuild)

步骤 05 JNIsojniLibsApp

3

1.

JNIJavaC/C++14-2

14-2

数据类型名称	Java 的数据类型	JNI 的数据类型	C/C++的数据类型
整型	int	jint	int
浮点数	float	jfloat	float
双精度	double	jdouble	double
布尔型	boolean	jboolean	unsigned char
字符串	String	jstring	const char*

3C/C++C/C++

(1)JavafalseC/C++0JavatruC/C++1

(2)JNIenv → GetStringUTFCharsjstringconst char*

env → NewStringUTFconst char*jstring

2.

Android StudioC/C++build.gradleAndroid.mk14-3

14-3

build.gradle 的规则名称	Android.mk 的规则名称	说明	常用值
moduleName	LOCAL_MODULE	so 库文件的名称	
无	LOCAL_SRC_FILES	需要编译的源文件	
cFlags	LOCAL_CPPFLAGS	C++的编译标志	-fexceptions (支持 try..catch..)
ldLibs	LOCAL_LDLIBS	需要链接的库, 多个库用逗号分隔	log (支持打印日志)
stl	APP_STL	stl 库的集成方式	stlport_static(表示使用 STLport 作为静态库)
abiFilters	APP_ABI	目标库的指令集, 多个指令集用逗号分隔	armeabi-v7a(表示高级的 ARM)

3.

JNIJavaC/C++

(1)C/C++

(2)JNIJavaJavaJNI

(3)JNIsocketsocket-1JNISDSD

CPUJNIJNIget_cpu.cpp

```

#include <jni.h>
#include <string.h>
#include <stdio.h>

extern "C"

jstring Java_com_example_mixture_JniCpuActivity_cpuFromJNI( JNIEnv* env, jobject thiz, jint i1, jfloat f1,
jdouble d1, jboolean b1 ) {
    #if defined(__arm__)
        #if defined(__ARM_ARCH_7A__)
            #if defined(__ARM_NEON__)
                #if defined(__ARM_PCS_VFP)
                    #define ABI "armeabi-v7a/NEON (hard-float)"
                #else
                    #define ABI "armeabi-v7a/NEON"
                #endif
            #endif
        #endif
        #if defined(__ARM_PCS_VFP)
            #define ABI "armeabi-v7a (hard-float)"
        #else
            #define ABI "armeabi-v7a"
        #endif
    #endif
    #else
        #define ABI "armeabi"
    #endif
    #elif defined(__i386__)
        #define ABI "x86"
    #elif defined(__x86_64__)
        #define ABI "x86_64"
    #elif defined(__mips64) /* mips64el-* toolchain defines __mips__ too */
        #define ABI "mips64"
    #elif defined(__mips__)
        #define ABI "mips"
    #elif defined(__aarch64__)
        #define ABI "arm64-v8a"
    #else
        #define ABI "unknown"
    #endif
    char desc[200] = {0};
    sprintf(desc, "%d %f %lf %u \nHello from JNI! Compiled with %s.", i1, f1, d1, b1, ABI);
}

```

```

return env->NewStringUTF(desc);
}

```

JavaBuildJNIC++

```

public class JniCpuActivity extends AppCompatActivity implements OnClickListener {
    private TextView tv_cpu_build;
    private TextView tv_cpu_jni;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_jni_cpu);
        tv_cpu_build = (TextView) findViewById(R.id.tv_cpu_build);
        tv_cpu_jni = (TextView) findViewById(R.id.tv_cpu_jni);
        findViewById(R.id.btn_cpu).setOnClickListener(this);
        tv_cpu_build.setText("Build 类获得的 CPU 指令集为"+Build.CPU_ABI);
    }

    @Override
    public void onClick(View v) {
        if (v.getId() == R.id.btn_cpu) {
            String desc = cpuFromJNI(1, 0.5f, 99.9, true);
            tv_cpu_jni.setText(desc);
        }
    }

    public native String cpuFromJNI(int i1, float f1, double d1, boolean b1);
    public native String unimplementedCpuFromJNI(int i1, float f1, double d1, boolean b1);
    static {
        System.loadLibrary("jni_mix");
    }
}

```

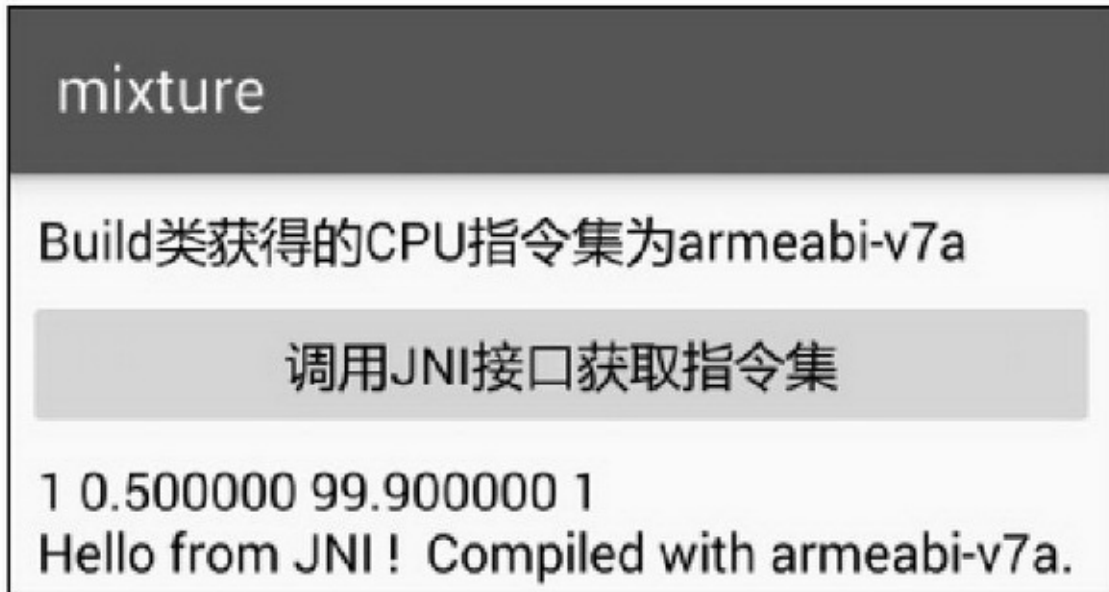
JNI14-1514-1614-1514-16

mixture

Build类获得的CPU指令集为x86

调用JNI接口获取指令集

1 0.500000 99.900000 1
Hello from JNI ! Compiled with armeabi-v7a.



14.2.3JNI

JNI

1.

JavaJavasoJNI

2.

JavaC/C++C/C++

3.

C/C++JavaiOSC/C++ObjectCAndroidJavaiOSAndroid

AndroidSQLiteJavaRealmSQLiteRealmC/C++

4.

AndroidiOSAppAndroidiOSC/C++AndroidJNI

iOS

JNIC/C++C++AESC++JNI

JNIAES

```

#include <jni.h>
#include <string.h>
#include <stdio.h>
#include "aes.h"
#include <android/log.h>
// log 标签
#define TAG "MyMsg"
// 定义 info 信息
#define LOGI(...) __android_log_print(ANDROID_LOG_INFO,TAG,__VA_ARGS__)

extern "C"

jstring Java_com_example_mixture_JniSecretActivity_encryptFromJNI( JNIEnv* env, jobject thiz, jstring
raw, jstring key) {
    const char* str_raw;
    const char* str_key;
    str_raw = env->GetStringUTFChars(raw, 0);
    str_key = env->GetStringUTFChars(key, 0);
    LOGI("str_raw=%s, str_key=%s ", str_raw, str_key);
    char encrypt[1024] = {0};
    AES aes_en((unsigned char*)str_key);
    aes_en.Cipher((char*)str_raw, encrypt);
    LOGI("encrypt=%s", encrypt);
    return env->NewStringUTF(encrypt);
}

```

JNIAES


```

#include <jni.h>
#include <string.h>
#include <stdio.h>
#include "aes.h"
#include <android/log.h>
// log 标签
#define TAG "MyMsg"
// 定义 info 信息
#define LOGI(...) __android_log_print(ANDROID_LOG_INFO,TAG,__VA_ARGS__)

extern "C"

JNIEXPORT jstring JNICALL Java_com_example_mixture_JniSecretActivity_decryptFromJNI( JNIEnv* env, jobject thiz, jstring des,
jstring key) {
    const char* str_des;
    const char* str_key;
    str_des = env->GetStringUTFChars(des, 0);
    str_key = env->GetStringUTFChars(key, 0);
    LOGI("str_des=%s, str_key=%s ", str_des, str_key);
    char decrypt[1024] = {0};
    AES aes_de((unsigned char*)str_key);
    aes_de.InvCipher((char*)str_des, decrypt);
    LOGI("decrypt=%s", decrypt);
    return env->NewStringUTF(decrypt);
}

```

Java

```

public class JniSecretActivity extends AppCompatActivity implements OnClickListener {
    private EditText et_origin;
    private EditText et_encrypt;
    private TextView tv_cpu_build;
    private TextView tv_decrypt;
    private String mKey = "123456789abcdef"; //该算法要求密钥值长度为 16 位

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_jni_secret);
        et_origin = (EditText) findViewById(R.id.et_origin);
        et_encrypt = (EditText) findViewById(R.id.et_encrypt);
        tv_decrypt = (TextView) findViewById(R.id.tv_decrypt);
        findViewById(R.id.btn_encrypt).setOnClickListener(this);
        findViewById(R.id.btn_decrypt).setOnClickListener(this);
    }

    @Override
    public void onClick(View v) {
        if (v.getId() == R.id.btn_encrypt) {
            String des = encryptFromJNI(et_origin.getText().toString(), mKey);
            et_encrypt.setText(des);
        } else if (v.getId() == R.id.btn_decrypt) {
            String raw = decryptFromJNI(et_encrypt.getText().toString(), mKey);
            tv_decrypt.setText(raw);
        }
    }

    public native String encryptFromJNI(String raw, String key);
    public native String unimplementedEncryptFromJNI(String raw, String key);
    public native String decryptFromJNI(String des, String key);
    public native String unimplementedEencryptFromJNI(String des, String key);
    static {
        System.loadLibrary("jni_mix");
    }
}

```

JNI14-1714-1814-17JNI14-18JNI

mixture

ABC888

调用JNI接口获取加密串

35E46A721F4483687547C9BE7D5262F0

调用JNI接口获取解密串

14-17JNI

mixture

ABC888

调用JNI接口获取加密串

35E46A721F4483687547C9BE7D5262F0

调用JNI接口获取解密串

ABC888

14-18JNI

14.3

——WIFI4

14.3.1 WifiManager 10AppConnectivityManager

WIFIConnectivityManagerWIFIWIFIWIFI

WifiManagerContext.WIFI_SERVICEWifiManager

isWifiEnabledWLAN

setWifiEnabledWLAN

getWifiStateWIFIWIFI14-4

14-4WIFI

WifiManager 类的连接状态	说明
WIFI_STATE_DISABLED	已断开 WIFI
WIFI_STATE_DISABLING	正在断开 WIFI
WIFI_STATE_ENABLED	已连上 WIFI
WIFI_STATE_ENABLING	正在连接 WIFI
WIFI_STATE_UNKNOWN	连接状态未知

➤ getSSIDWIFIMAC

➤ getRssiWIFI

➤ getLinkSpeed

➤ getNetworkIdWIFI

➤ getIpAddressIPIPv4

➤ getMacAddressMAC

getConnectionInfoWIFIWifiInfoWIFI

startScanWIFI

getScanResultsWIFI

calculateSignalLevel
getConfiguredNetworks
addNetworkWIFI



14-19WIFI

enableNetworkWIFIWIFI
disableNetworkWIFI
disconnectWIFI
WIFIWIFI14-19WIFIWIFIIPMAC

14.3.2BlueTooth WIFI4GABQQQQ

WIFI

Android4BuletoothAdapterBluetoothDeviceBluetoothServerSocket
BluetoothSocket

1.BuletoothAdapter

BuletoothAdapter***ManagerBuletoothAdapter

```

getDefaultAdapter
enable
Intent intent = new
Intent(BluetoothAdapter.ACTION_REQUEST_DISCOVERABLE);
startActivityForResult(intent, 1);
disable
isEnabledtruefalse
startDiscovery
cancelDiscovery
isDiscovering
getBondedDevices
setName
getName
getAddress
getRemoteDevice
getStateBluetoothAdapter.STATE_ON
listenUsingRfcommWithServiceRecordUUIDBluetoothServerSocket
listenUsingRfcommOnBluetoothServerSocket
2. BluetoothDevice
BluetoothDeviceBluetoothAdapterBluetoothDevice
getName
getAddress
getBondState14-5

```

14-5

BluetoothDevice 类的绑定状态	说明
BOND_NONE	未绑定（未配对）
BOND_BONDING	正在绑定（正在配对）
BOND_BONDED	已绑定（已配对）

createBond

createRfcommSocketToServiceRecordUUIDBluetoothSocket

createRfcommSocketBluetoothSocket

3. BluetoothServerSocket

BluetoothServerSocketSocketsocket

acceptBluetoothSocket

close

4. BluetoothSocket

BluetoothSocketSocket

connectsocket

closesocket

getInpuStreamssocket

getOutputStreamsocket

getRemoteDevicesocket

5

1.



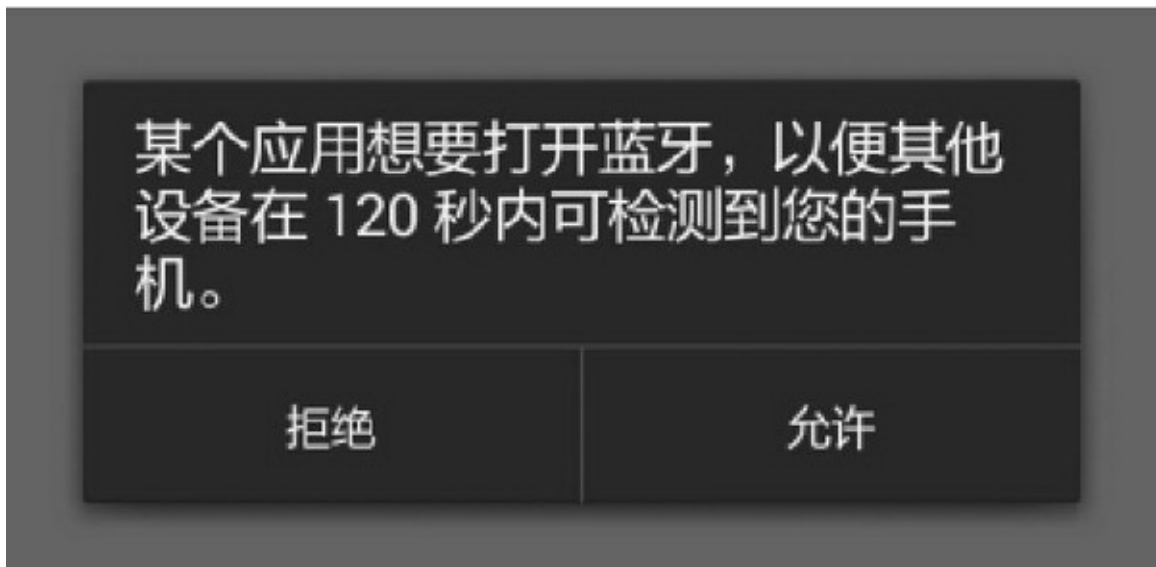
14-20DEMO

AppApp14-20

A14-21B14-22



14-21A



14-22B

“””A14-23B14-24



14-23A



14-24B

2.

A14-25B14-26

“””A14-27B14-28

蓝牙配对请求

设备

Lenovo A808t (B66A)

配对码

707653

配对之后，所配对的设备将可以在建立连接后访问您的通讯录和通话记录。

取消

配对

名称	地址	状态
小米手机	64:CC:2E:77:53:5A	未绑定

蓝牙配对请求

要与以下设备配对：
小米手机

请确保其显示的配对密钥为：
707653

取消 | **配对**

14-26B



14-27A



14-28B

3.

BluetoothSocketconnectBluetoothServerSocketaccept

“*****”“”“”A14-29B14-30



14-29A



14-30B

4.

AA14-31“”BAB14-32



14-31A



14-32B

```
demo
AppAndroidManifest.xml
<!-- -->
<uses-permission
android:name="android.permission.BLUETOOTH_ADMIN" /> <uses-
permission android:name="android.permission.BLUETOOTH" />
```

```
public class BluetoothActivity extends AppCompatActivity implements
    OnClickListener, OnItemClickListener, OnCheckedChangeListener,
    BlueConnectListener, InputCallbacks, BlueAcceptListener {
    private static final String TAG = "BluetoothActivity";
    private CheckBox ck_bluetooth;
    private TextView tv_discovery;
    private ListView lv_bluetooth;
    private BluetoothAdapter mBluetooth;
    private ArrayList<BlueDevice> mDeviceList = new ArrayList<BlueDevice>();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_bluetooth);
        ck_bluetooth = (CheckBox) findViewById(R.id.ck_bluetooth);
        tv_discovery = (TextView) findViewById(R.id.tv_discovery);
```



```

lv_bluetooth = (ListView) findViewById(R.id.lv_bluetooth);
if (BluetoothUtil.getBlueToothStatus(this) == true) {
    ck_bluetooth.setChecked(true);
}
ck_bluetooth.setOnCheckedChangeListener(this);
tv_discovery.setOnClickListener(this);
mBluetooth = BluetoothAdapter.getDefaultAdapter();
if (mBluetooth == null) {
    Toast.makeText(this, "本机未找到蓝牙功能", Toast.LENGTH_SHORT).show();
    finish();
}
}

@Override
public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
    if (buttonView.getId() == R.id.ck_bluetooth) {
        if (isChecked == true) {
            beginDiscovery();
            Intent intent = new Intent(BluetoothAdapter.ACTION_REQUEST_DISCOVERABLE);
            startActivityForResult(intent, 1);
            // 下面这行代码为服务端需要, 客户端不需要
            mHandler.postDelayed(mAccept, 1000);
        } else {
            cancelDiscovery();
            BluetoothUtil.setBlueToothStatus(this, false);
            mDeviceList.clear();
            BlueListAdapter adapter = new BlueListAdapter(this, mDeviceList);
            lv_bluetooth.setAdapter(adapter);
        }
    }
}

private Runnable mAccept = new Runnable() {
    @Override
    public void run() {
        if (mBluetooth.getState() == BluetoothAdapter.STATE_ON) {
            BlueAcceptTask acceptTask = new BlueAcceptTask(true);
            acceptTask.setBlueAcceptListener(BluetoothActivity.this);
            acceptTask.executeOnExecutor(AsyncTask.THREAD_POOL_EXECUTOR);
        } else {
            mHandler.postDelayed(this, 1000);
        }
    }
};
};

```

```

@Override
public void onClick(View v) {
    if (v.getId() == R.id.tv_discovery) {
        beginDiscovery();
    }
}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent intent) {
    super.onActivityResult(requestCode, resultCode, intent);
    if (requestCode == 1) {
        if (resultCode == RESULT_OK) {
            Toast.makeText(this, "允许本地蓝牙被附近其他蓝牙设备发现",
                Toast.LENGTH_SHORT).show();
        } else if (resultCode == RESULT_CANCELED) {
            Toast.makeText(this, "不允许蓝牙被附近其他蓝牙设备发现",
                Toast.LENGTH_SHORT).show();
        }
    }
}

private Runnable mRefresh = new Runnable() {
    @Override
    public void run() {
        beginDiscovery();
        mHandler.postDelayed(this, 2000);
    }
};

private void beginDiscovery() {
    if (mBluetooth.isDiscovering() != true) {
        mDeviceList.clear();
        BlueListAdapter adapter = new BlueListAdapter(BluetoothActivity.this, mDeviceList);
        lv_bluetooth.setAdapter(adapter);
        tv_discovery.setText("正在搜索蓝牙设备");
        mBluetooth.startDiscovery();
    }
}

private void cancelDiscovery() {
    mHandler.removeCallbacks(mRefresh);
    tv_discovery.setText("取消搜索蓝牙设备");
    if (mBluetooth.isDiscovering() == true) {
        mBluetooth.cancelDiscovery();
    }
}

```

```

    }

    @Override
    protected void onStart() {
        super.onStart();
        mHandler.postDelayed(mRefresh, 50);
        blueReceiver = new BluetoothReceiver();
        //需要过滤多个动作，则调用 IntentFilter 对象的 addAction 添加新动作
        IntentFilter foundFilter = new IntentFilter(BluetoothDevice.ACTION_FOUND);
        foundFilter.addAction(BluetoothAdapter.ACTION_DISCOVERY_FINISHED);
        foundFilter.addAction(BluetoothDevice.ACTION_BOND_STATE_CHANGED);
        registerReceiver(blueReceiver, foundFilter);
    }

    @Override
    protected void onStop() {
        super.onStop();
        cancelDiscovery();
        unregisterReceiver(blueReceiver);
    }

    private BluetoothReceiver blueReceiver;
    private class BluetoothReceiver extends BroadcastReceiver {
        @Override
        public void onReceive(Context context, Intent intent) {
            String action = intent.getAction();
            Log.d(TAG, "onReceive action="+action);
            // 获得已经搜索到的蓝牙设备
            if (action.equals(BluetoothDevice.ACTION_FOUND)) {
                BluetoothDevice device = intent.getParcelableExtra(BluetoothDevice.EXTRA_DEVICE);
                BlueDevice item = new BlueDevice(device.getName(), device.getAddress(),
device.getBondState()-10);
                mDeviceList.add(item);
                BlueListAdapter adapter = new BlueListAdapter(BluetoothActivity.this, mDeviceList);
                lv_bluetooth.setAdapter(adapter);
                lv_bluetooth.setOnItemClickListener(BluetoothActivity.this);
            } else if (action.equals(BluetoothAdapter.ACTION_DISCOVERY_FINISHED)) {
                mHandler.removeCallbacks(mRefresh);
                tv_discovery.setText("蓝牙设备搜索完成");
            } else if (action.equals(BluetoothDevice.ACTION_BOND_STATE_CHANGED)) {
                BluetoothDevice device = intent.getParcelableExtra(BluetoothDevice.EXTRA_DEVICE);
                if (device.getBondState() == BluetoothDevice.BOND_BONDING) {
                    tv_discovery.setText("正在配对"+device.getName());
                } else if (device.getBondState() == BluetoothDevice.BOND_BONDED) {

```

```

        tv_discovery.setText("完成配对"+device.getName());
        mHandler.postDelayed(mRefresh, 50);
    } else if (device.getBondState() == BluetoothDevice.BOND_NONE) {
        tv_discovery.setText("取消配对"+device.getName());
    }
}
}
}

@Override
public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
    cancelDiscovery();
    BlueDevice item = mDeviceList.get(position);
    BluetoothDevice device = mBluetooth.getRemoteDevice(item.address);
    try {
        if (device.getBondState() == BluetoothDevice.BOND_NONE) {
            Method createBondMethod = BluetoothDevice.class.getMethod("createBond");
            Log.d(TAG, "开始配对");
            Boolean result = (Boolean) createBondMethod.invoke(device);
        } else if (device.getBondState() == BluetoothDevice.BOND_BONDED &&
            item.state != BlueListAdapter.CONNECTED) {
            tv_discovery.setText("开始连接");
            BlueConnectTask connectTask = new BlueConnectTask(item.address);
            connectTask.setBlueConnectListener(this);
            connectTask.executeOnExecutor(AsyncTask.THREAD_POOL_EXECUTOR, device);
        } else if (device.getBondState() == BluetoothDevice.BOND_BONDED &&
            item.state == BlueListAdapter.CONNECTED) {
            tv_discovery.setText("正在发送消息");
            InputDialogFragment dialog = InputDialogFragment.newInstance(
                "", 0, "请输入要发送的消息");
            String fragTag = getResources().getString(R.string.app_name);
            dialog.show(getFragmentManager(), fragTag);
        }
    } catch (Exception e) {
        e.printStackTrace();
        tv_discovery.setText("配对异常 : "+e.getMessage());
    }
}

//向对方发送消息
@Override
public void onInput(String title, String message, int type) {
    Log.d(TAG, "onInput message="+message);
    BluetoothUtil.writeOutputStream(mBlueSocket, message);
}

```

```

}

private BluetoothSocket mBlueSocket;
//客户端主动连接
@Override
public void onBlueConnect(String address, BluetoothSocket socket) {
    mBlueSocket = socket;
    tv_discovery.setText("连接成功");
    refreshAddress(address);
}

//刷新已连接的状态
private void refreshAddress(String address) {
    for (int i=0; i<mDeviceList.size(); i++) {
        BlueDevice item = mDeviceList.get(i);
        if (item.address.equals(address) == true) {
            item.state = BlueListAdapter.CONNECTED;
            mDeviceList.set(i, item);
        }
    }
    BlueListAdapter adapter = new BlueListAdapter(this, mDeviceList);
    lv_bluetooth.setAdapter(adapter);
}

//服务端侦听到连接
@Override
public void onBlueAccept(BluetoothSocket socket) {
    Log.d(TAG, "onBlueAccept socket is "+(socket==null?"null":"not null"));
    if (socket != null) {
        mBlueSocket = socket;
        BluetoothDevice device = mBlueSocket.getRemoteDevice();
        refreshAddress(device.getAddress());
        BlueReceiveTask receive = new BlueReceiveTask(mBlueSocket, mHandler);
        receive.start();
    }
}

//收到对方发来的消息
private Handler mHandler = new Handler() {
    @Override
    public void handleMessage(Message msg) {
        if (msg.what == 0) {
            byte[] readBuf = (byte[]) msg.obj;
            String readMsg = new String(readBuf, 0, msg.arg1);
            Log.d(TAG, "handleMessage readMessage="+ readMsg);
        }
    }
}

```

```
        AlertDialog.Builder builder = new AlertDialog.Builder(BluetoothActivity.this);
        builder.setTitle("我收到消息啦").setMessage(readMsg).setPositiveButton("确定", null);
        builder.create().show();
    }
}
};

@Override
protected void onDestroy() {
    super.onDestroy();
    if (mBlueSocket != null) {
        try {
            mBlueSocket.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
}
```

14.4WIFI

WIFIWIFI——“WIFI
”

14.4.1

WIFIWIFI/WLAN/WIFIWIFI

WIFIWIFI14-33

App



14-33WIFI

- (1)WifiManagerWifiManager
 - (2)/proc/net/arpIPMAC
 - (3)InetAddressInetAddressisReachable
 - (4)AsyncTask
 - (5)AssetManagerMACMACIEEEEMAC6MACMAC
- <http://standards.ieee.org/regauth/oui/oui.txt>
- (6)SQLiteMACSQLite
 - (7)IntentServiceassetsMAC

(8)SocketsocketNetBIOS
(9)JNICNetBIOSJNIJava
App

14.4.2NetBIOS

NetBIOSWindowsTCP/IPNetBIOSWindowsNetBIOS

NetBIOSIPWindowsNetBIOSIPMAC

JavaIPGetClientName.java14-34

```
public static void main(String[] args) throws Exception {  
    String ip = "192.168.0.212";  
    GetClientName add = new GetClientName(ip);  
    System.out.println("下面是"+ip+"的探测结果");  
    System.out.println(add.getRemoteInfo());  
}
```

Problems @ Javadoc Declaration Console LogCat

<terminated> GetClientName (1) [Java Application] D:\Program Files (x
下面是192.168.0.212的探测结果
192.168.0.212|OUYANGSHEN|54-EE-75-22-40-30

14-34JavaNetBIOS

NetBIOSJNICJNI

```

#include <jni.h>
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <netdb.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <sys/select.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>

#define send_MAXSIZE 50
#define recv_MAXSIZE 1024
struct NETBIOSNS {
    unsigned short int tid;        //unsigned short int 占 2 字节
    unsigned short int flags;
    unsigned short int questions;
    unsigned short int answerRRS;
    unsigned short int authorityRRS;
    unsigned short int additionalRRS;
    unsigned char name[34];
    unsigned short int type;
    unsigned short int classe;
};
char *getNameFromIp(const char *ip);

extern "C"

jstring Java_com_example_mixture_WifiShareActivity_nameFromJNI( JNIEnv* env, jobject thiz, jstring ip) {
    const char* str_ip;
    str_ip = env->GetStringUTFChars(ip, 0);
    return env->NewStringUTF(getNameFromIp(str_ip));
}

char *getNameFromIp(const char *ip) {
    char str_info[1024] = {0};
    struct sockaddr_in toAddr;        //在 sendto 中使用的对方地址

```

```

struct sockaddr_in fromAddr; //在 recvfrom 中使用的对方主机地址
char send_buff[send_MAXSIZE];
char recv_buff[recv_MAXSIZE];
memset(send_buff, 0, sizeof(send_buff));
memset(recv_buff, 0, sizeof(recv_buff));
int sockfd; //socket
unsigned int udp_port = 137;
int inetat;
if ( (inetat = inet_aton(ip, &toAddr.sin_addr)) == 0) {
    sprintf(str_info, "[%s] is not a valid IP address\n", ip);
    return str_info;
}
if ( (sockfd = socket(AF_INET, SOCK_DGRAM, IPPROTO_UDP)) < 0) {
    sprintf(str_info, "%s socket error sockfd=%d, inetat=%d\n", ip, sockfd, inetat);
    return str_info;
}
bzero((char*)&toAddr, sizeof(toAddr));
toAddr.sin_family = AF_INET;
toAddr.sin_addr.s_addr = inet_addr(ip);
toAddr.sin_port = htons(udp_port);
//构造 NetBIOS 结构包
struct NETBIOS nbns;
nbns.tid=0x0000;
nbns.flags=0x0000;
nbns.questions=0x0100;
nbns.answerRRS=0x0000;
nbns.authorityRRS=0x0000;
nbns.additionalRRS=0x0000;
nbns.name[0]=0x20;
nbns.name[1]=0x43;
nbns.name[2]=0x4b;
int j=0;
for (j=3j<34j++) {
    nbns.name[j]=0x41;
}
nbns.name[33]=0x00;
nbns.type=0x2100;
nbns.classe=0x0100;
memcpy(send_buff, &nbns, sizeof(nbns));
int send_num =0;
send_num = sendto(sockfd, send_buff, sizeof(send_buff), 0, (struct sockaddr *)&toAddr,
sizeof(toAddr) );
if (send_num != sizeof(send_buff)) {

```

```

        sprintf(str_info, "%s sendto() error sockfd=%d, send_num=%d, sizeof(send_buff)=%d\n", ip,
sockfd, send_num, sizeof(send_buff));
        shutdown(sockfd, 2);
        return str_info;
    }
    int recv_num = recvfrom(sockfd, recv_buff, sizeof(recv_buff), 0, (struct sockaddr *)NULL,
(socklen_t*)NULL);
    if (recv_num < 56) {
        sprintf(str_info, "%s recvfrom() error sockfd=%d, recv_num=%d\n", ip, sockfd, recv_num);
        shutdown(sockfd, 2);
        return str_info;
    }
    //这里要初始化。因为发现 Linux 和模拟器都没问题，真机上该变量如果不初始化，值就不可预知
    unsigned short int NumberOfNames=0;
    memcpy(&NumberOfNames, recv_buff+56, 1);
    char str_name[1024] = {0};
    unsigned short int mac[6]={0};
    int i=0;
    for (i=0; i<NumberOfNames; i++) {
        char NetbiosName[16];
        memcpy(NetbiosName, recv_buff+57+i*18, 16); //依次读取 NetBIOS name
        if (i == 0) {
            sprintf(str_name, "%s", NetbiosName);
        }
    }
    sprintf(str_info, "%s|%s|", ip, str_name);
    for (i=0; i<6; i++) {
        memcpy(&mac[i], recv_buff+57+NumberOfNames*18+i, 1);
        sprintf(str_info, "%s%02X", str_info, mac[i]);
        if (i != 5) {
            sprintf(str_info, "%s-", str_info);
        }
    }
    return str_info;
}

```

14.4.3

(1)MACassets

(2)WIFIAndroidManifest.xml

```
<!-- -->
```

```
<uses-permission
```

```
android:name="android.permission.ACCESS_NETWORK_STATE" /> <uses-
permission android:name="android.permission.ACCESS_WIFI_STATE" /> <!--
```

WLAN -->

```
<uses-permission  
android:name="android.permission.ACCESS_WIFI_STATE" /> <uses-  
permission android:name="android.permission.CHANGE_WIFI_STATE" /> <!--  
-->
```

```
<uses-permission android:name="android.permission.INTERNET" /> (3)  
AndroidManifest.xmlMAC
```

```
<service android:name=".service.ImportService" android:enabled="true" />
```

(4)WIFI

```
WIFIWifiManagerWifiManagerWIFI  
setWifiApEnabledWIFI  
getWifiApState :WIFIWIFI14-6
```

14-6WIFI

WifiManager 类的 WIFI 热点状态	说明
WIFI_AP_STATE_DISABLED	WIFI 热点已关闭
WIFI_AP_STATE_DISABLING	WIFI 热点正在关闭
WIFI_AP_STATE_ENABLED	WIFI 热点已开启
WIFI_AP_STATE_ENABLING	WIFI 热点正在开启
WIFI_AP_STATE_FAILED	WIFI 热点开启失败

```
isWifiApEnabledWIFItruefalse  
getWifiApConfigurationWIFI  
setWifiApConfigurationWIFI  
addToBlacklistMAC  
clearBlacklist  
WIFIAndroidAndroidSDKhideJava  
hide  
WIFI
```

```

public static String setWifiApEnabled(WifiManager wifiMgr, WifiConfiguration apConfig, boolean
enabled) {
    String desc = "";
    if (apConfig.SSID == null || apConfig.SSID.length() <= 0) {
        desc = "WIFI 名称为空";
        return desc;
    }
    try {
        if (enabled) {
            // wifi 和 WIFI 不能同时打开，所以打开 WIFI 时需要关闭 wifi
            wifiMgr.setWifiEnabled(false);
        }
        // 通过反射调用设置 WIFI
        Method method = wifiMgr.getClass().getMethod("setWifiApEnabled",
            WifiConfiguration.class, Boolean.TYPE);
        // 返回 WIFI 打开状态
        if ((Boolean) method.invoke(wifiMgr, apConfig, enabled) != true) {
            desc = "WIFI 操作失败";
        }
    } catch (Exception e) {
        e.printStackTrace();
        desc = "WIFI 操作异常 : " + e.getMessage();
    }
    return desc;
}

public static int getWifiApState(WifiManager wifiMgr) {
    try {
        Method method = wifiMgr.getClass().getMethod("getWifiApState");
        int i = (Integer) method.invoke(wifiMgr);
        if (i > 9) {
            i -= 10;
        }
        Log.d(TAG, "wifi state: " + i);
        return i;
    } catch (Exception e) {
        e.printStackTrace();
        return WIFI_AP_STATE_FAILED;
    }
}

```

WIFI14-35WIFI14-36

mixture

开关WIFI



WIFI名称: CMAY9DG6PRJJBI99

WIFI密码:

加密方式:

无

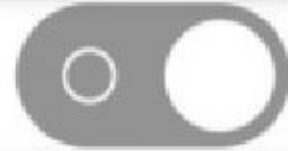


保存设置

当前没有设备连接

mixture

开关WIFI



WIFI名称: CMAY9DG6PRJJBI99

WIFI密码:

加密方式: 无



保存设置

当前没有设备连接



14-37WIFI

WIFI“CMAY9D***”()wifiWIFI“CMAY9D***”
OUYANGSHEN14-37WIFIWIFI
WIFI3WIFI
WIFI

```

public class WifiShareActivity extends AppCompatActivity implements
    OnClickListener, OnCheckedChangeListener, GetClientListener, FindNameListener {
    private static final String TAG = "WifiShareActivity";
    private CheckBox ck_wifi_switch;
    private EditText et_wifi_name, et_wifi_password;
    private Spinner sp_wifi_des;
    private TextView tv_connect;
    private LinearLayout ll_client_title;
    private ListView lv_wifi_client;
    private WifiManager mWifiManager;
    private WifiConfiguration mWifiConfig = new WifiConfiguration();

```

```

    private int mDesType = 0;
    private ArrayList<ClientScanResult> mClientArray = new ArrayList<ClientScanResult>();
    private HashMap<String, String> mapName = new HashMap<String, String>();
    private Handler mHandler = new Handler();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_wifi_share);
        ck_wifi_switch = (CheckBox) findViewById(R.id.ck_wifi_switch);
        et_wifi_name = (EditText) findViewById(R.id.et_wifi_name);
        et_wifi_password = (EditText) findViewById(R.id.et_wifi_password);
        tv_connect = (TextView) findViewById(R.id.tv_connect);
        ll_client_title = (LinearLayout) findViewById(R.id.ll_client_title);
        lv_wifi_client = (ListView) findViewById(R.id.lv_wifi_client);
        findViewById(R.id.btn_wifi_save).setOnClickListener(this);
        et_wifi_name.setText(Build.SERIAL);
        et_wifi_password.setText("");
        ck_wifi_switch.setOnCheckedChangeListener(this);
        mWifiManager = (WifiManager) getSystemService(Context.WIFI_SERVICE);
        setWifiConfig();

        sp_wifi_des = (Spinner) findViewById(R.id.sp_wifi_des);
        ArrayAdapter<String> starAdapter = new ArrayAdapter<String>(this,
            R.layout.item_select, desNameArray);
        starAdapter.setDropDownViewResource(R.layout.item_select);
        sp_wifi_des.setAdapter(starAdapter);
        sp_wifi_des.setSelection(mDesType);
        sp_wifi_des.setPrompt("请选择加密方式");
        sp_wifi_des.setOnItemSelectedListener(new DesTypeSelectedListener());
        sp_wifi_des.setSelection(mDesType);
        mHandler.postDelayed(mClientTask, 50);
    }

    private String[] desNameArray = {"无", "WPA PSK", "WPA2 PSK"};
    private int[] desTypeArray = {WifiConfiguration.KeyMgmt.NONE, WifiConfiguration.KeyMgmt.
WPA_PSK, 4};

    private class DesTypeSelectedListener implements OnItemSelectedListener {
        @Override
        public void onItemSelected(AdapterView<?> arg0, View arg1, int arg2, long arg3) {
            mDesType = arg2;
        }
    }

```

```

@Override
public void onNothingSelected(AdapterView<?> arg0) {
}
}

private Runnable mClientTask = new Runnable() {
@Override
public void run() {
    GetClientListTask getClientTask = new GetClientListTask();
    getClientTask.setGetClientListener(WifiShareActivity.this);
    getClientTask.execute();
    mHandler.postDelayed(this, 3000);
}
};

@Override
public void onClick(View v) {
    if (v.getId() == R.id.btn_wifi_save) {
        if (et_wifi_name.getText().length() < 4) {
            Toast.makeText(this, "WIFI 名称长度需不小于四位", Toast.LENGTH_SHORT).show();
            return;
        } else if (mDesType!=0 && et_wifi_password.getText().length() < 8) {
            Toast.makeText(this, "WIFI 密码长度需不小于八位", Toast.LENGTH_SHORT).show();
            return;
        }
        Toast.makeText(this, "已保存本次 WIFI 设置", Toast.LENGTH_SHORT).show();
        //只有已开启 WIFI, 才需要断开并重连。当前未开启 WIFI, 保存设置后不自动开启 WIFI
        int timeout = 0;
        if (ck_wifi_switch.isChecked() == true) {
            ck_wifi_switch.setChecked(false);
            timeout = 2000;
        }
        setWifiConfig();
        mHandler.postDelayed(mReOpenTask, timeout);
    }
}

private void setWifiConfig() {
    mWifiConfig.allowedKeyManagement.clear();
    mWifiConfig.SSID = et_wifi_name.getText().toString();
    if (mDesType == 0) {
        mWifiConfig.preSharedKey = "";
        mWifiConfig.wepKeys[0] = et_wifi_password.getText().toString();
    }
}

```

```

        mWifiConfig.wepTxKeyIndex = 0;
    } else {
        mWifiConfig.allowedKeyManagement.set(desTypeArray[mDesType]);
        mWifiConfig.preSharedKey = et_wifi_password.getText().toString();
        mWifiConfig.allowedAuthAlgorithms.set(WifiConfiguration.AuthAlgorithm.OPEN);
        mWifiConfig.allowedProtocols.set(WifiConfiguration.Protocol.RSN);
        mWifiConfig.allowedPairwiseCiphers.set(WifiConfiguration.PairwiseCipher.CCMP);
        mWifiConfig.allowedPairwiseCiphers.set(WifiConfiguration.PairwiseCipher.TKIP);
        mWifiConfig.allowedGroupCiphers.set(WifiConfiguration.GroupCipher.CCMP);
        mWifiConfig.allowedGroupCiphers.set(WifiConfiguration.GroupCipher.TKIP);
    }
}

private Runnable mReOpenTask = new Runnable() {
    @Override
    public void run() {
        if (WifiUtil.getWifiApState(mWifiManager) == WifiUtil.WIFI_AP_STATE_DISABLED) {
            ck_wifi_switch.setChecked(true);
        } else {
            mHandler.postDelayed(this, 2000);
        }
    }
};

@Override
public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
    if (buttonView.getId() == R.id.ck_wifi_switch) {
        String result = "";
        if (isChecked == false) {
            result = WifiUtil.setWifiApEnabled(mWifiManager, mWifiConfig, isChecked);
        } else {
            setWifiConfig();
            result = WifiUtil.setWifiApEnabled(mWifiManager, mWifiConfig, isChecked);
        }
        Log.d(TAG, "onCheckedChanged: " + isChecked + ", " + result);
        if (result != null && result.length() > 0) {
            Toast.makeText(this, result, Toast.LENGTH_SHORT).show();
            ck_wifi_switch.setChecked(!isChecked);
        }
    }
}

@Override

```

```

public void onGetClient(ArrayList<ClientScanResult> clientList) {
    mClientArray = clientList;
    Log.d(TAG, "mClientArray.size()=" + mClientArray.size());
    if (WifiUtil.getWifiApState(mWifiManager) != WifiUtil.WIFI_AP_STATE_ENABLING
        && WifiUtil.getWifiApState(mWifiManager) != WifiUtil.WIFI_AP_STATE_ENABLED) {
        mClientArray.clear();
    } else if (mClientArray == null) {
        mClientArray = new ArrayList<ClientScanResult>();
    }
    if (mClientArray.size() <= 0) {
        tv_connect.setText("当前没有设备连接");
        ll_client_title.setVisibility(View.GONE);
    } else {
        String desc = String.format("当前已有%d 台设备连接", mClientArray.size());
        tv_connect.setText(desc);
        ll_client_title.setVisibility(View.VISIBLE);
    }
    for (int i = 0; i < mClientArray.size(); i++) {
        ClientScanResult item = mClientArray.get(i);
        String ipAddr = item.getIpAddr();
        item.setDevice(MacManager.getInstance(this).getMacDevice(item.getHWAddr()));
        if (mapName.containsKey(ipAddr)) {
            item.setHostName(mapName.get(ipAddr));
        } else {
            item.setHostName(MacManager.getInstance(this).getDeviceName(item.getDevice()));
            String upperDevice = item.getDevice().toUpperCase();
            if (upperDevice.equals("INTEL") || upperDevice.equals("HEWLETT")
                || upperDevice.equals("DELL") || upperDevice.equals("ASUS")
                || upperDevice.equals("ACER") || upperDevice.equals("TOSHIBA")) {
                Log.d(TAG, "new GetClientNameTask");
                GetClientNameTask getNameTask = new GetClientNameTask();
                getNameTask.setFindNameListener(WifiShareActivity.this);
                getNameTask.execute(ipAddr);
            }
        }
    }
    ClientListAdapter clientAdapter = new ClientListAdapter(this, mClientArray);
    lv_wifi_client.setAdapter(clientAdapter);
}

public static native String nameFromJNI(String ip);
public static native String unimplementedNameFromJNI(String ip);
static {

```

```

        System.loadLibrary("jni_mix");
    }

    @Override
    public void onFindName(String info) {
        if (info != null && info.length() > 0) {
            String[] split = info.split("\\|");
            if (split.length > 1 && split[1].length() > 0) {
                mapName.put(split[0], split[1]);
            }
        }
    }
}

```

14.5

App()JNI(NDKJNIJNI)“WIFI
”AppWIFINetBIOS

4

(1)

(2)JNI

(3)

(4)WIFI

第15章 第三方开发包



AppAndroidApp()(QQ)()“”

15.1 SDK

AndroidAppSDK

15.1.1

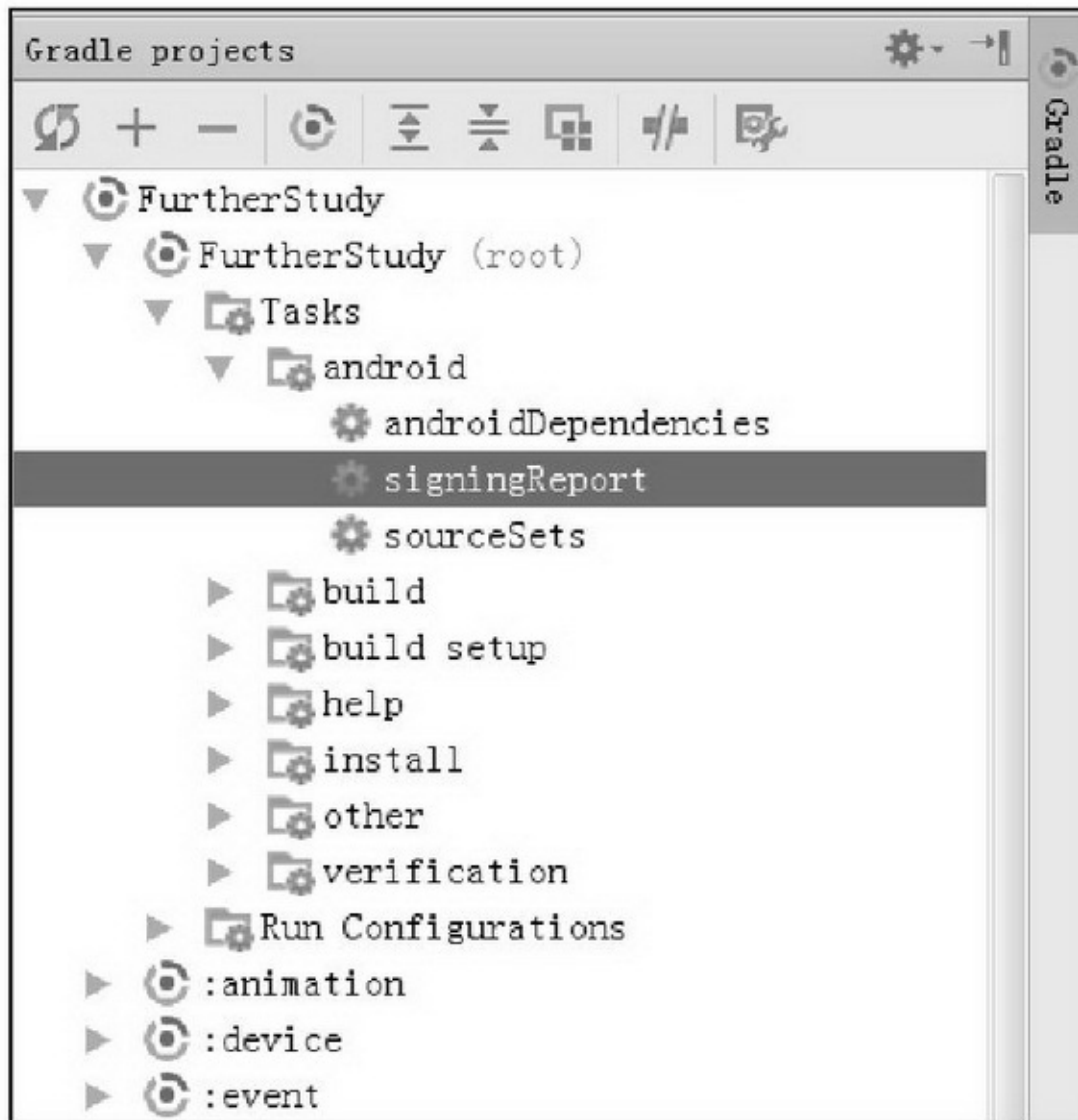
AppAppAppAppApp

AppAppAppAndroid

SHA1Android Studiodebug.keystoreAppAppSDK()

AppAndroid Studio

1.



15-1 Gradle

Android Studio.android/debug.keystoreAndroid StudioGradle
Tasks/android3androidDependenciesissigningReportsourceSets15-1

signingReportsigningReportAndroid StudiosigningReport
SHA1thirdsdk15-2



15-2

SDKdemobuild.gradleandroiddebug.keystore

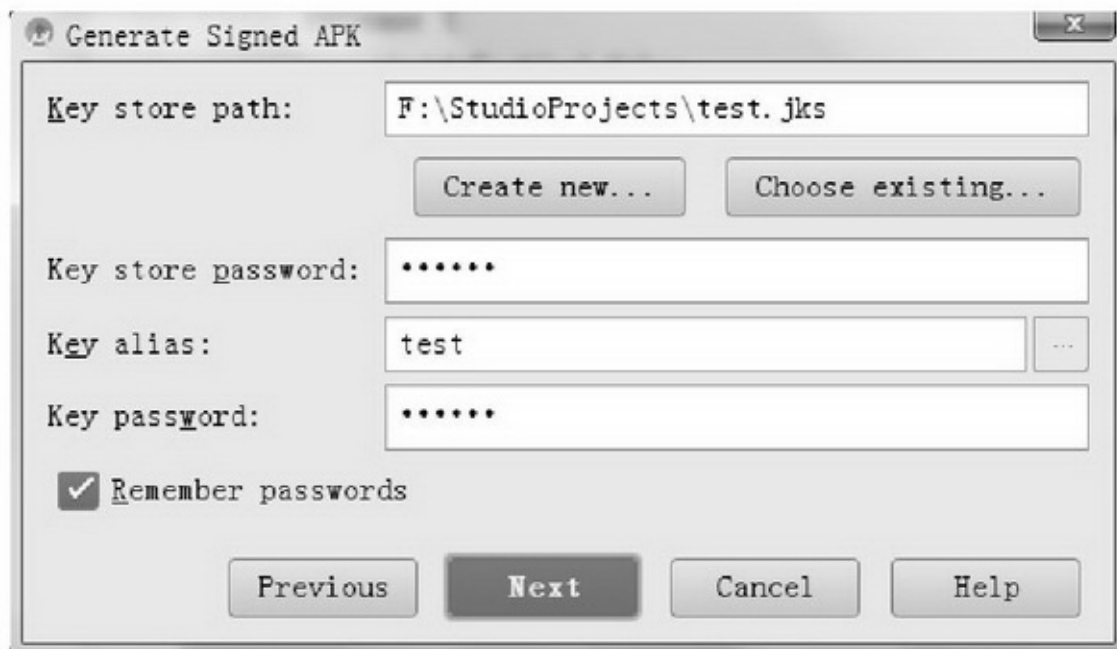
```
signingConfigs {  
    debug {  
        storeFile file("debug.keystore")  
    }  
}
```

2.

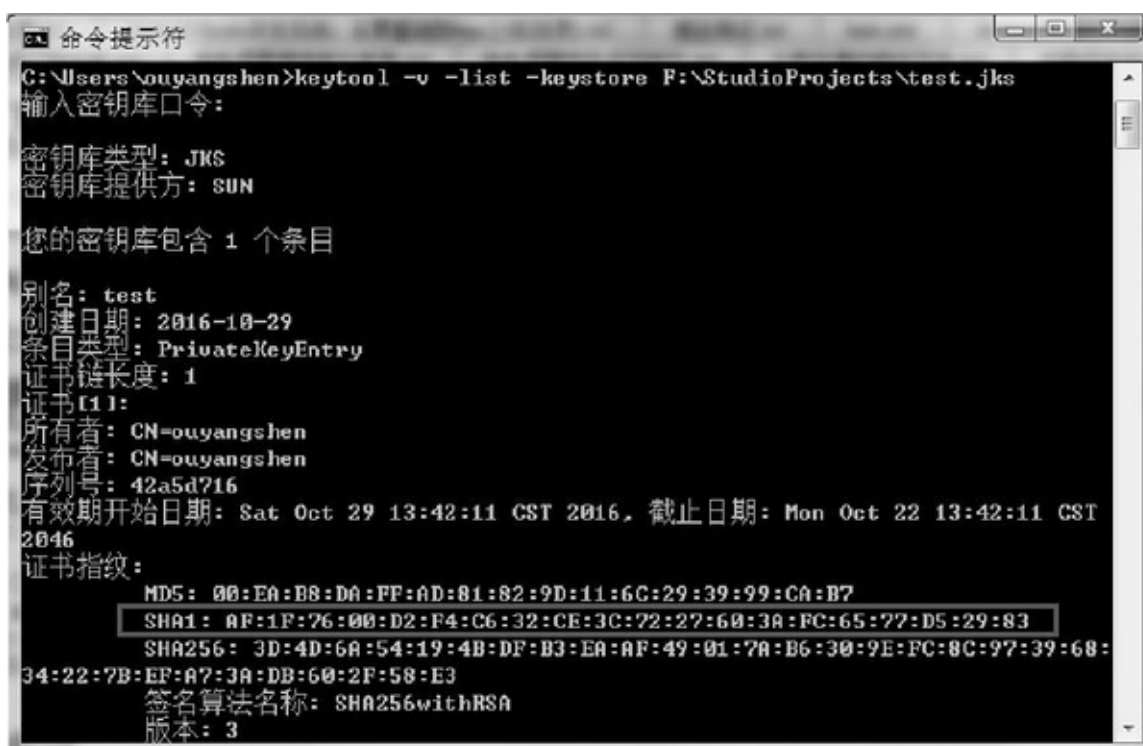
8AppAppBuild → Generate Signed APK...APK15-3

test.jkskeytool -v -list -keystore F:\StudioProjects\test.jks

15-4SHA1



15-3APK



15-4

15.1.2

http://lbsyun.baidu.com/“” → “Android” → “Android SDK” → “”SDK“”“”SDK“”15-5



15-5SDK

“”SDK“”demo

SDK

步骤 01

http://developer.baidu.com/“” → “” → “”

15-6



15-6

步骤 02 “”15-7



15-7

步骤

03

“”API Key“LBS”15-8

The screenshot shows the '修改应用' (Modify Application) page in the Baidu API Console. The page is for an application with ID 9087711. The application name is '第三方地图' (Third-party Map) and the application type is 'Android SDK'. The '启用服务' (Enable Services) section lists various APIs with checkboxes, including Cloud Search API, Geocoding API v2, Android Map SDK, Static Map API, Map API, Cloud Geocoding API, Javascript API, IP Location API, Android Navigation SDK, Full Screen Static Map API, Full Screen URL API, Routematrix API, Place API v2, Route Traffic API, Android Navigation SDK, Coordinate Conversion API, Android Navigation HUD SDK, and Cloud Geocoding API. The '发布版SHA1' (Production SHA1) field contains 'E0:95:CF:F8:D7:43:9C:74:90:95:86:F6:96:36:10:E8:DF:0C:A6:63'. The '开发版SHA1' (Development SHA1) field is empty with the placeholder '请输入开发版SHA1'. The '包名' (Package Name) field contains 'com.example.thirdsdk'. The '安全码' (Security Code) field is empty. A note at the bottom states: 'Android SDK安全码组成: SHA1+包名。(查看详情配置方法)'. A warning below the note says: '新申请的Mobile与Browser类型的ak不再支持云存储接口的访问, 如要使用云存储, 请申请Server类型ak.' A '提交' (Submit) button is at the bottom.

15-8

步骤

04

Android SDKSHA1“”

15-9



15-9

AndroidManifest.xml application android:value API Key

```

<!-- 百度地图密钥 -->
<meta-data
    android:name="com.baidu.lbsapi.API_KEY"
    android:value="vRbVCiHqbhdccoG8wOQwQvdX" />

```

SDK BaiduLBS_Android.jar
 libs\jars\src\main\jniLibs\democom\baidu\mapapi\overlayutilPOISDK.jar
 SDK3.6

POI

1.

SDK

(1) SDK setContentView

SDKInitializer.initialize(getApplicationContext());

(2)

API

```

private MapView mMapView;
private BaiduMap mMapLayer;
private LocationClient mLocClient;
private boolean isFirstLoc = true;           // 是否首次定位
private double m_latitude;
private double m_longitude;

private void initLocation() {
    mMapView = (MapView) findViewById(R.id.bmapView);
    mMapView.setVisibility(View.INVISIBLE); // 先隐藏地图, 待定位到当前城市时再显示
    mMapLayer = mMapView.getMap();
    mMapLayer.setOnMapClickListener(this);
    mMapLayer.setMyLocationEnabled(true);   // 开启定位图层
}

```

```

mLocClient = new LocationClient(this);
mLocClient.registerLocationListener(new MyLocationListener()); // 设置定位监听器
LocationClientOption option = new LocationClientOption();
option.setOpenGps(true);           // 打开 GPS
option.setCoorType("bd09ll");     // 设置坐标类型
option.setScanSpan(1000);         // 设置定位的时间间隔
option.setIsNeedAddress(true);    // 设置 true 才能获得详细的地址信息
mLocClient.setLocOption(option);   // 设置定位参数
mLocClient.start();               // 开始定位
}

public class MyLocationListener implements BDLocationListener {
    @Override
    public void onReceiveLocation(BDLocation location) {
        if (location == null || mMapView == null) { // map view 销毁后不再处理新接收的位置
            Log.d(TAG, "location is null or mMapView is null");
            return;
        }
        m_latitude = location.getLatitude();
        m_longitude = location.getLongitude();
        String position = String.format("当前位置 : %s%s%s%s%s%s",
            location.getProvince(), location.getCity(), location.getDistrict(), location.getStreet(),
            location.getStreetNumber(), location.getAddrStr(), location.getTime());
        loc_position.setText(position);
        MyLocationData locData = new MyLocationData.Builder().accuracy(location.getRadius())
            // 此处设置开发者获取的方向信息, 顺时针 0~360
            .direction(100).latitude(m_latitude).longitude(m_longitude).build();
        mMapLayer.setMyLocationData(locData);
        Toast.makeText(MapBaiduActivity.this, "isFirstLoc=" + isFirstLoc,
            Toast.LENGTH_LONG).show();
        if (isFirstLoc) {
            isFirstLoc = false;
            LatLng ll = new LatLng(m_latitude, m_longitude);
            MapStatusUpdate update = MapStatusUpdateFactory.newLatLngZoom(ll, 14);
            mMapLayer.animateMapStatus(update);
            mMapView.setVisibility(View.VISIBLE); // 定位到当前城市时再显示图层
        }
    }

    public void onReceivePoi(BDLocation poiLocation) {
    }
}

```




15-10

15-10

2. POI

POI Point Of Interest POISDK

POI POI 15-11 15-12 15-11 “ ” 15-12



15-11POI



15-12POI

3.

SDK

0)abca

$$a^2 + b^2 = c^2$$

$$S = \sqrt{p(p-a)(p-b)(p-c)}$$

0)3

S = ((x

$$y_1 - x_1 y_0) + (x_1 y_2 - x_2 y_1)$$

MapLayeraddOverlay15-1

百度地图的标记类	MapLayer 类的添加方法	说明
ArcOptions	addOverlay	弧线
CircleOptions	addOverlay	圆圈
MarkerOptions	addOverlay	图片
PolygonOptions	addOverlay	多边形
PolylineOptions	addOverlay	线段
TextOptions	addOverlay	文本

15-1315-1415-135.915-1484





15.1.3

<http://lbs.amap.com/> → “Android” → “AndroidSDK”
“”15-15
“”“”SDK



15-15SDK

“”” → “””15-16



15-16“”

“”15-17



15-17

“Key”“Key”KeySHA115-18



15-18“Key”

KeyAndroidSDK()Package()“***”“”

Key15-19Key



15-19

AndroidManifest.xmlapplicationandroid:valueKey

```
<!-- 高德地图密钥 -->
<meta-data
    android:name="com.amap.api.v2.apikey"
    android:value="d2d98282615cb90e78b4be537636647c" />
```

```
<service android:name="com.amap.api.location.APSService" />
```

```
SDKAMap***.jarlibsJARAMap2DMap***.jar
```

```
AMapSearch***.jarAMapLocation***.jar
```

3POI

1.

```
private MapView mMapView;
private AMap mMapLayer;
private AMapLocationClient mLocClient;
private boolean isFirstLoc = true; // 是否首次定位
private double m_latitude;
private double m_longitude;

private void initLocation(Bundle savedInstanceState) {
    mMapView = (MapView) findViewById(R.id.amapView);
    mMapView.onCreate(savedInstanceState);
```



```

mMapView.setVisibility(View.INVISIBLE); // 先隐藏地图，待定位到当前城市时再显示
if (mMapLayer == null) {
    mMapLayer = mMapView.getMap();
}
mMapLayer.setOnMapClickListener(this);
mMapLayer.setMyLocationEnabled(true); // 开启定位图层
mLocClient = new AMapLocationClient(this, getApplicationContext());
mLocClient.setLocationListener(new MyLocationListener()); // 设置定位监听器
AMapLocationClientOption option = new AMapLocationClientOption();
option.setLocationMode(AMapLocationMode.Battery_Saving);
option.setNeedAddress(true); // 设置 true 才能获得详细的地址信息
mLocClient.setLocationOption(option); // 设置定位参数
mLocClient.startLocation(); // 开始定位
}

public class MyLocationListener implements AMapLocationListener {
    @Override
    public void onLocationChanged(AMapLocation location) {
        if (location == null || mMapView == null) { // map view 销毁后不再处理新接收的位置
            Log.d(TAG, "location is null or mMapView is null");
            return;
        }
        m_latitude = location.getLatitude();
        m_longitude = location.getLongitude();
        String position = String.format("当前位置 : %s|%s|%s|%s|%s|%s",
            location.getProvince(), location.getCity(), location.getDistrict(),
location.getStreet(),
            location.getAdCode(), location.getAddress(), location.getTime());
        loc_position.setText(position);
        if (isFirstLoc) {
            isFirstLoc = false;
            LatLng ll = new LatLng(m_latitude, m_longitude);
            CameraUpdate update = CameraUpdateFactory.newLatLngZoom(ll, 12);
            mMapLayer.moveCamera(update);
            mMapView.setVisibility(View.VISIBLE); // 定位到当前城市时再显示图层
        }
    }
}
}

```

15-20

2. POI

POI POI15-21 15-22 15-21 “” 15-22

Thirsdsk

城市中搜索 ▾ 开始 下一组数据

在 市内找

清除标记 当前位置: 福建省福州市鼓楼区北环中路|350102|福建省福州市鼓楼区北环中路靠近思儿亭高架桥|1482367059464

高德地图



15-21POI



15-22POI

3.

MapLayeradd***15-2

15-2

高德地图的标记类	MapLayer 类的添加方法	说明
CircleOptions	addCircle	圆圈
MarkerOptions	addMarker	图片
PolygonOptions	addPolygon	多边形
PolylineOptions	addPolyline	线段
TextOptions	addText	文本

15-2315-2415-235A4.015-245955





15.2 SDK

```
//AppAppQQQQQQ
```

15.2.1 QQ

```
QQQQQQQQhttps://connect.qq.com/“” → “SDK” → “SDK”  
”QQSDKQQ  
QQTencent  
createInstanceappidTencent  
loginQQIUiListener  
setAccessToken  
setOpenId  
getQQTokenQQ  
shareToQQQQ15-3  
shareToQzoneQQ15-3
```

15-3 QQ

QQShare 类的分享参数	说明
SHARE_TO_QQ_KEY_TYPE	分享类型。图文分享（普通分享）填 Tencent.SHARE_TO_QQ_TYPE_DEFAULT
PARAM_TARGET_URL	分享消息被点击后的跳转 URL
PARAM_TITLE	分享的标题
PARAM_SUMMARY	分享的消息摘要
SHARE_TO_QQ_IMAGE_URL	分享图片的 URL 或本地路径
SHARE_TO_QQ_APP_NAME	在手机 QQ 顶部的“返回”按钮文字后加上应用名。若为空，则“返回”按钮保持原样

```
QQActivityonActivityResultTencentonActivityResultData
```

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    Log.d(TAG, "-->onActivityResult " + requestCode + " resultCode=" + resultCode);
    if (requestCode==Constants.REQUEST_LOGIN || requestCode==Constants.REQUEST_APPBAR) {

        Tencent.onActivityResultData(requestCode,resultCode,data,ItemTencentAdapter.mLoginListener);
    } else if (requestCode==Constants.REQUEST_QQ_SHARE || requestCode==Constants.
REQUEST_QZONE_SHARE) {
        Tencent.onActivityResultData(requestCode,resultCode,data,ItemTencentAdapter.mShareListener);
    }
    super.onActivityResult(requestCode, resultCode, data);
}
```

QQ15-25~15-2815-2515-26QQQQ3QQ
15-2715-28

Thirdsdk

分享标题:

分享内容:



15-26QQ



15-27



15-28

15.2.2

QQ<https://open.weixin.qq.com/>“” → “” → “Android”
”3

1.demodebug.keystore build.gradleandroid

```
signingConfigs {
    debug {
        storeFile file("debug.keystore")
    }
}
```

2.demonet.sourceforge.simcpux AndroidManifest.xmlpackage
build.gradleapplicationId

```

defaultConfig {
    applicationId "net.sourceforge.simcpux"
    minSdkVersion 15
    targetSdkVersion 25
    versionCode 1
    versionName "1.0"
}

```

3. AndroidManifest.xml WXEntryActivity WXEntryActivity.java
 .wxapi” AndroidManifest.xml

```

<activity
    android:name="net.sourceforge.simcpux.wxapi.WXEntryActivity"
    android:configChanges="keyboardHidden|orientation|screenSize"
    android:exported="true"
    android:screenOrientation="portrait"
    android:theme="@android:style/Theme.Translucent.NoTitleBar" />

```

IWXAPISendMessageToWX.ReqWXMediaMessageIWXAPI
 createWXAPIAPIappidregisterAppappid
 registerAppappid
 sendReqSendMessageToWX.Req
 SendMessageToWX.Req
 transaction
 sceneSendMessageToWX.Req.WXSceneSession
 SendMessageToWX.Req.WXSceneTimeline
 messageWXMediaMessage
 WXMediaMessage
 title
 description
 mediaObject15-4
 thumbData

媒体对象类	说明
WXTextObject	文本
WXImageObject	图片
WXWebpageObject	图文（既有文本，又有图片）
WXMusicObject	音乐
WXVideoObject	视频
WXFileObject	文件

QQ

```
public void onItemClick(AdapterView<?> arg0, View arg1, int arg2, long arg3) {  
    ShareChanel item = mChannelList.get(arg2);  
    mHandler.sendMessageDelayed(0, 1500);  
    if (item.channelType == WEIXIN) {  
        SendMessageToWX.Req req = new SendMessageToWX.Req();
```

```

// transaction 字段用于唯一标识一个请求
req.transaction = "wx_share" + System.currentTimeMillis();
req.message = getWXMessage();
req.scene = SendMessageToWX.Req.WXSceneSession;
mWeixinApi.sendReq(req);
} else if (item.channelType == CIRCLE) {
    SendMessageToWX.Req req = new SendMessageToWX.Req();
    req.transaction = "wx_share" + System.currentTimeMillis();
    req.message = getWXMessage();
    req.scene = SendMessageToWX.Req.WXSceneTimeline;
    mWeixinApi.sendReq(req);
} else if (item.channelType == QQ) {
    mShareListener = new ShareQQListener(mContext, item.channelName);
    Bundle params = new Bundle();
    params.putInt(QQShare.SHARE_TO_QQ_KEY_TYPE, QQShare.SHARE_TO_QQ_
TYPE_DEFAULT);
    params.putString(QQShare.SHARE_TO_QQ_TITLE, mTitle);
    params.putString(QQShare.SHARE_TO_QQ_SUMMARY, mContent);
    params.putString(QQShare.SHARE_TO_QQ_TARGET_URL, mUrl);
    params.putString(QQShare.SHARE_TO_QQ_IMAGE_URL, mImageUrl);
    params.putString(QQShare.SHARE_TO_QQ_APP_NAME, mContext.getPackageName());
    mTencent.shareToQQ((Activity) mContext, params, mShareListener);
} else if (item.channelType == QZONE) {
    mShareListener = new ShareQQListener(mContext, item.channelName);
    ArrayList<String> urlList = new ArrayList<String>();
    urlList.add(mImageUrl);
    Log.d(TAG, "mImageUrl=" + mImageUrl);
    Bundle params = new Bundle();
    params.putInt(QzoneShare.SHARE_TO_QZONE_KEY_TYPE, QzoneShare.SHARE_TO_
QZONE_TYPE_IMAGE_TEXT);
    params.putString(QzoneShare.SHARE_TO_QQ_TITLE, mTitle);
    params.putString(QzoneShare.SHARE_TO_QQ_SUMMARY, mContent);
    params.putString(QzoneShare.SHARE_TO_QQ_TARGET_URL, mUrl);
    params.putStringArrayList(QzoneShare.SHARE_TO_QQ_IMAGE_URL, urlList);
    Log.d(TAG, "begin shareToQzone");
    mTencent.shareToQzone((Activity) mContext, params, mShareListener);
    Log.d(TAG, "end shareToQzone");
} else if (item.channelType == WEIBO) { // 腾讯微博分享需要 QQ 登录授权
    mTencent.login((Activity) mContext, "all", mLoginListener);
}
}

private int THUMB_SIZE = 150;

```

```

private WXMediaMessage getWXMessage() {
    WXMediaMessage msg = new WXMediaMessage();
    if (!TextUtils.isEmpty(mTitle) && !TextUtils.isEmpty(mImageUrl)) { // 分享文本消息
        WXTextObject textObj = new WXTextObject();
        textObj.text = mContent;
        msg.mediaObject = textObj;
        msg.title = mTitle;
        msg.description = mContent;
    } else if (TextUtils.isEmpty(mTitle) && !TextUtils.isEmpty(mImageUrl)) { // 分享图片消息
        Bitmap bmp = BitmapFactory.decodeFile(mImageUrl);
        WXImageObject imgObj = new WXImageObject(bmp);
        msg.mediaObject = imgObj;
        Bitmap thumbBmp = Bitmap.createScaledBitmap(bmp, THUMB_SIZE, THUMB_SIZE,
true);
        msg.thumbData = CacheUtil.bmpToByteArray(thumbBmp, true); // 设置缩略图
    } else if (!TextUtils.isEmpty(mTitle) && !TextUtils.isEmpty(mImageUrl)) { // 分享图文消息
        WXWebpageObject webpage = new WXWebpageObject();
        webpage.webpageUrl = mUrl;
        msg.title = mTitle;
        msg.description = mContent;
        msg.mediaObject = webpage;
        Bitmap bmp = BitmapFactory.decodeFile(mImageUrl);
        Bitmap thumbBmp = Bitmap.createScaledBitmap(bmp, THUMB_SIZE, THUMB_SIZE,
true);
        msg.thumbData = CacheUtil.bmpToByteArray(thumbBmp, true); // 设置缩略图
    }
    return msg;
}

public static ShareQQListener mShareListener;
private static class ShareQQListener implements IUiListener {
    private Context context;
    private String cn;
    public ShareQQListener(final Context context, final String channelName) {
        this.context = context;
        this.cn = channelName;
    }

    @Override
    public void onComplete(Object object) {
        Toast.makeText(context, cn + "分享完成:" + object.toString(), Toast.LENGTH_LONG).show();
    }
}

```

```

@Override
public void onError(UiError error) {
    Toast.makeText(context, cn + "分享失败:" + error.errorMessage, Toast.LENGTH_LONG).show();
}

@Override
public void onCancel() {
    Toast.makeText(context, cn + "分享取消", Toast.LENGTH_LONG).show();
}
}

```

15-29~15-3215-2915-30515-31

15-32

weixin

分享标题： 别看我只是一只羊

分享内容： 喜羊羊，美羊羊，懒羊羊，沸羊羊，慢羊羊，软绵绵，红太狼，灰太狼，别看我只是一只羊，绿草因为我变得更香，天空因为我变得更蓝，白云因为我变得柔软

分享到微信



15-30



15-31



15-32

15.3SDK

71.5%15.99%7/8(2015)

15.3.1

AppSDK

(1)

(2)App

(3)RSA+BASE64MD5

<https://open.alipay.com/>“” → “” → “” → “App

DEMO&SDK”SDKDEMO

“” → “”“”RSA15-33

APPID



15-33

SDKAndroidManifest.xmlJARalipaySdk-***.jar()
(APPID)RSA(RSA)
SDK
(1)
(2)RSAURL
(3)(UI)
(4)SDK
(5)

DEMOThread+HandlerActivityAsyncTask
SDK15-3415-3515-34“”15-35

Thirdsdk



商品名称: 大白兔奶糖

商品描述: 中国驰名商标, 畅销全球一百多个国家和地区。

商品价格: 0.01 元

支付宝支付



15-35

15.3.2

<https://open.weixin.qq.com/>“” → “” → “Android”
libammsdk.jarSDK
“” → “”15-36
DEMO

接口信息			
接口名称	接口介绍	接口状态	操作
分享到朋友圈	将内容分享到微信朋友圈	已获得	--
发送给朋友	将内容发送给朋友或者群聊	已获得	--
微信支付	获得微信支付能力 详情	未获得	申请开通
微信登录	使用微信帐号登录App或者网站 详情	未获得	申请开通
微信卡券	移动应用内领取卡券收入微信卡包 详情	未获得	申请开通
智能接口	获得语音识别、图像识别、语义理解等模式识别能力	未获得	申请开通

15-36

(1)WXPayEntryActivity.java“.wxapi”AndroidManifest.xml
activity

```

<!-- 微信支付回调页面 -->
<activity
    android:name="net.sourceforge.simcpux.wxapi.WXPayEntryActivity"
    android:exported="true"
    android:launchMode="singleTop" >
</activity>

```

(2)Failed to find provider info for
com.tencent.mm.sdk.plugin.provider

(1)APP_IDAPP_SECRET

(2)(PARTNER_IDPARTNER_KEY)MD5

(3)

(4)

(5)SDK

(6)

App

SDK15-3715-3815-37“”15-38

weixin



商品名称: 老干妈辣酱

商品描述: 中国名牌称号, 歪果仁眼中来自中国的进口奢侈品。

商品价格: 0.01 元

微信支付

15-37



15.4 SDK

App

15.4.1

[http://www.xfyun.cn/“SDK”\(Android\)“](http://www.xfyun.cn/SDK(Android)“)
SDK”

SDK“” → “”15-39

The screenshot shows the '讯飞开放平台' (Xfyun Open Platform) interface for creating a new application. The browser address bar shows the URL: `www.xfyun.cn/index.php/mycloud/app/create`. The page title is '创建新应用' (Create New Application). The form contains the following fields:

- *应用名称: 第三方应用
- *应用分类: 应用-教育学习-其他
- *应用功能描述: 语音识别和语音合成的测试demo
- *应用平台: Android
- 我已阅读并接受《讯飞用户协议》

At the bottom of the form, there are two buttons: '提交' (Submit) and '取消' (Cancel). The left sidebar contains navigation links such as '我的语音云首页', '应用管理', 'SDK下载中心', '个人中心', '个人资料', '资质信息', '奖品券', '我的订单', '权限管理', and '账户管理'.

15-39

“***”“”15-40



15-40

“”15-41

“”“”“”“”“”“”“”

15-42



15-41



15-42

15-43Appid



15-43

SDK

(1)Msc.jarSunflower.jarlibslibmsc.sosrc/main/jniLibs(SDK SDK“”)

(2)Application.onCreateAppid

SpeechUtility.createUtility(MainApplication.this, "appid=58561727"); (3)

AndroidManifest.xml Application

(4)RecognizerDialogDEMO assets

(5)-keep class com.iflytek.**{*}; SDK

SDKSpeechRecognizer

createRecognizer

setParameter15-5

15-5

SpeechConstant 类的识别参数	说明
ENGINE_TYPE	设置听写引擎。TYPE_LOCAL 表示本地，TYPE_CLOUD 表示云端，TYPE_MIX 表示混合
RESULT_TYPE	设置返回结果格式。比如 json 表示 json 格式
LANGUAGE	设置语言。zh_cn 表示中文，en_us 表示英文
ACCENT	设置方言。mandarin 表示普通话，cantonese 表示粤语，henanese 表示河南话
VAD_BOS	设置静音超时时间，即用户多长时间不说话就当作超时处理
VAD_EOS	设置静音检测时间，即用户停止说话多长时间内就认为不再输入，自动停止录音
ASR_PTT	设置标点符号。0 表示返回结果无标点，1 表示返回结果有标点
AUDIO_FORMAT	设置音频的保存格式
ASR_AUDIO_PATH	设置音频的保存路径
AUDIO_SOURCE	设置音频的来源。-1 表示音频流，与 writeAudio 配合使用；-2 表示外部文件，同时设置 ASR_SOURCE_PATH 指定文件路径
ASR_SOURCE_PATH	设置外部音频文件的路径

➤ onBeginOfSpeech

➤ onError10118()

➤ onEndOfSpeech

➤ onResult

➤ onVolumeChanged

➤ onEvent

startListeningRecognizerListener

stopListening

writeAudio

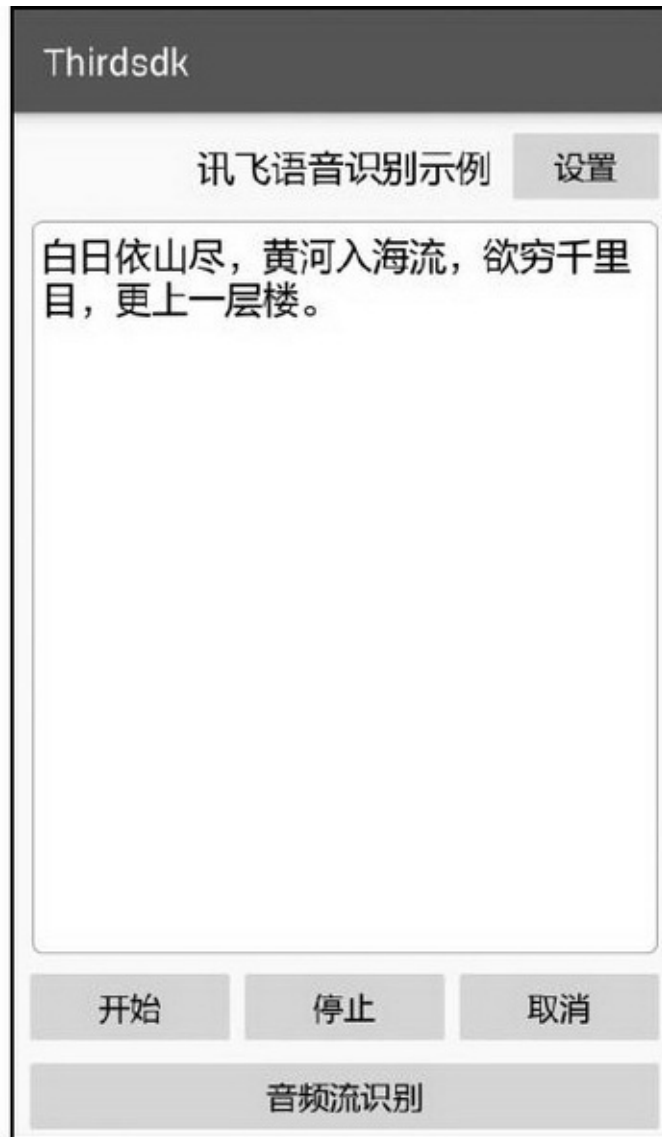
cancel

destroy

15-4415-4515-44“”AppSDK15-45



15-44App



15-45App

15.4.2

SDKSpeechSynthesizer
createSynthesizer
setParameter15-6

15-6

SpeechConstant 类的合成参数	说明
ENGINE_TYPE	设置合成引擎。TYPE_LOCAL 表示本地，TYPE_CLOUD 表示云端，TYPE_MIX 表示混合
VOICE_NAME	设置朗读者。默认 xiaoyan（女青年，普通话）
SPEED	设置朗读的语速，取值 0~100
PITCH	设置朗读的音调，取值 0~100
VOLUME	设置朗读的音量，取值 0~100
STREAM_TYPE	设置音频流类型。默认是 3，表示音乐
KEY_REQUEST_FOCUS	设置是否在播放合成音频时打断音乐播放，默认为 true
AUDIO_FORMAT	设置音频的保存格式
TTS_AUDIO_PATH	设置音频的保存路径

- onSpeakBegin
- onSpeakPaused
- onSpeakResumed
- onBufferProgress
- onSpeakProgress
- onCompleted
- onEvent

startSpeakingSynthesizerListener

synthesizeToUriStartSpeakingSynthesizerListener

pauseSpeaking

resumeSpeaking

stopSpeaking

destroy

15-4615-4715-46“”SDK15-47

在线合成发音人选项

小燕-女青、中英、普通话

小宇-男青、中英、普通话

凯瑟琳-女青、英语

亨利-男青、英语

玛丽-女青、英语

小研-女青、中英、普通话

小琪-女青、中英、普通话

小峰-男青、中英、普通话

小梅-女青、中英、粤语

小莉-女青、中英、台湾普通话

小蓉-女青、中、四川话

Thirsdsk

讯飞语音合成示例

设置

锄禾日当午，
汗滴禾下土。
谁知盘中餐，
粒粒皆辛苦。

发音人

开始合成

开始播放

取消

暂停播放

继续播放

15.5

AppAppApp“”

15.5.1

→ → → → 15-4815-4915-48App

15-49

“”“”

(1)SDK

(2)SDK“”“”



15-48App



15-49

(3)SDK

(4)SDK

AppSDKSDK“”4SDK

15.5.2RatingBar

RatingBar

RatingBarSeekBarRatingBarSeekBar515-7

15-7RatingBar

XML 的新增属性	RatingBar 的新增方法	说明
isIndicator	setIsIndicator	是否作为指示器。如果是指示器，就不可通过触摸修改评级
numStars	setNumStars	设置星星的个数
rating	setRating	设置初始评价等级
stepSize	setStepSize	设置每次增减的大小。默认为总数的十分之一，比如星星总数为 5，默认值为 0.5
无	setOnRatingBarChangeListener	设置评分监听器。 需实现接口 OnRatingBarChangeListener 的 onRatingChanged 方法

RatingBar315-8

15-8

评分条 style 属性的风格	星星的规格大小	默认能否触摸改变评级
?android:attr/ratingBarStyle	大, 默认值	能
?android:attr/ratingBarStyleIndicator	中	不能
?android:attr/ratingBarStyleSmall	小	不能

RatingBar3RatingBarprogressDrawableXML

```

<layer-list xmlns:android="http://schemas.android.com/apk/res/android" >
  <item
    android:id="@+android:id/background"
    android:drawable="@drawable/star_background">
  </item>
  <item
    android:id="@+android:id/secondaryProgress"
    android:drawable="@drawable/star_background">
  </item>
  <item
    android:id="@+android:id/progress"
    android:drawable="@drawable/star_foreground">
  </item>
</layer-list>

```

RatingBar

```

public class RatingBarActivity extends AppCompatActivity implements
    OnCheckedChangeListener, OnRatingBarChangeListener {
    private CheckBox ck_whole;
    private RatingBar rb_score;
    private TextView tv_rating;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_rating_bar);
        ck_whole = (CheckBox) findViewById(R.id.ck_whole);
        rb_score = (RatingBar) findViewById(R.id.rb_score);
        tv_rating = (TextView) findViewById(R.id.tv_rating);
        ck_whole.setOnCheckedChangeListener(this);
        rb_score.setOnRatingBarChangeListener(this);
    }

    @Override
    public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
        if (buttonView.getId() == R.id.ck_whole) {
            rb_score.setStepSize(ck_whole.isChecked()?1:rb_score.getNumStars()/10.0f);
        }
    }

    @Override
    public void onRatingChanged(RatingBar ratingBar, float rating, boolean fromUser) {
        String desc = String.format("当前选中的是%s 颗星", CacheUtil.formatDecimal(rating,1));
        tv_rating.setText(desc);
    }
}

```

15-5015-5115-5015-513



15-50



15-513

15.5.3

4

(1)libssrc/main/jniLibsSDK

(2)AndroidManifest.xmlAPPKEYactivityservice

(3)

(4)

“”App“”App“

”15-52App“”15-53

App“”15-54App“”“”15-

55







15-54

“”15-56“”15-57





Thirtdsk

行程结束, 请您进行评价



分享给好友



QQ好友



QQ空间



腾讯微博

已评价

15.6

AppSDK()SDK(QQ)SDK()SDK(
)"App4

4

(1)SDK

(2)SDKQQ

(3)SDK

(4)SDK

第16章 性能优化



App“”

16.1

AndroidApp

16.1.1

7ToolbarAppToolbarXMLToolbarXML

AndroidincludeJavainportC/C++include
includeXMLincludecommon_title.xml

```
<include layout="@layout/common_title" />
```

LinearLayoutRelativeLayoutAndroidmerge
mergemergeLinearLayoutRelativeLayoutAppmerge
UI

includemerge

```
<merge xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:app="http://schemas.android.com/apk/res-auto" >

  <android.support.v7.widget.Toolbar
    android:id="@+id/tl_head"
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:background="@color/blue_light"
    app:navigationIcon="@drawable/ic_back" >

    <RelativeLayout
      android:layout_width="match_parent"
      android:layout_height="wrap_content" >

      <TextView
        android:id="@+id/tv_title"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_centerInParent="true"
        android:paddingRight="50dp"
        android:textColor="@color/black"
        android:textSize="20sp" />

      <ImageView
        android:id="@+id/iv_share"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_alignParentRight="true"
        android:src="@drawable/ic_share"
        android:scaleType="fitCenter" />

    </RelativeLayout>
  </android.support.v7.widget.Toolbar>
</merge>
```

BaseActivityBaseActivity

```

public class BaseActivity extends AppCompatActivity {
    @Override
    protected void onResume() {
        super.onResume();
        Toolbar tl_head = (Toolbar) findViewById(R.id.tl_head);
        setSupportActionBar(tl_head);
        tl_head.setNavigationOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View view) {
                finish();
            }
        });
        findViewById(R.id.iv_share).setOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View v) {
                Toast.makeText(BaseActivity.this, "请先实现分享功能", Toast.LENGTH_LONG).
show();
            }
        });
    }

    protected void setTitle(String title) {
        TextView tv_title = (TextView) findViewById(R.id.tv_title);
        tv_title.setText(title);
    }
}

```

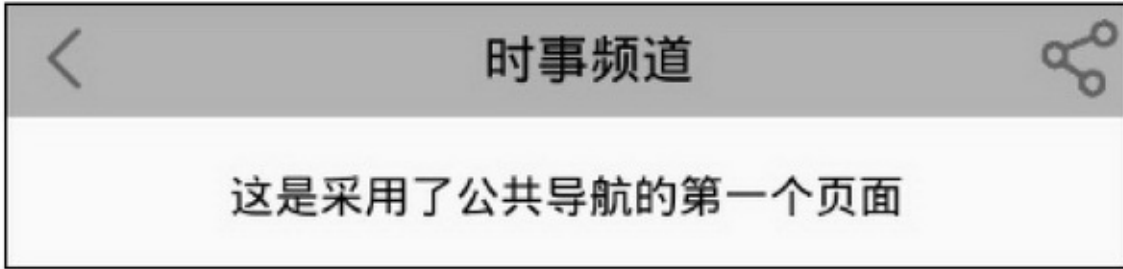
includecommon_titleBaseActivity

```

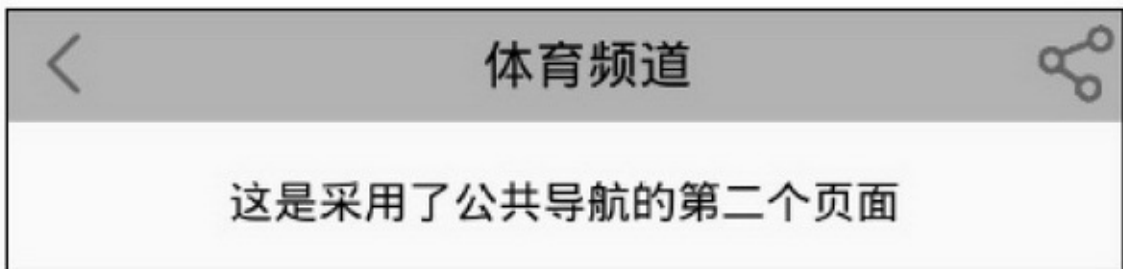
public class IncludeOneActivity extends BaseActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_include_one);
        setTitle("时事频道");
    }
}

```

16-116-216-116-2



16-1



16-2

16.1.2

`setVisibility(View.VISIBLE)`
`setVisibility(View.GONE)`

`ViewStub`

`ViewStubViewlayoutAppViewStubViewStubinflatelayout`

`ViewStub`

(1) `ViewStubViewlayout`

(2) `ViewStubinflatesetVisibility`

`ViewStubViewStubListViewGridView`

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <include layout="@layout/common_title" />

    <ViewStub
        android:id="@+id/vs_list"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout="@layout/viewstub_list" />

    <ViewStub
        android:id="@+id/vs_grid"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout="@layout/viewstub_grid" />
</LinearLayout>
```

```

public class ScreenSuitableActivity extends BaseActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_screen_suitable);
        setTitle("自适应布局演示页面");
        Configuration config = getResources().getConfiguration();
        if(config.orientation == Configuration.ORIENTATION_PORTRAIT){
            showList();
        } else {
            showGrid();
        }
    }

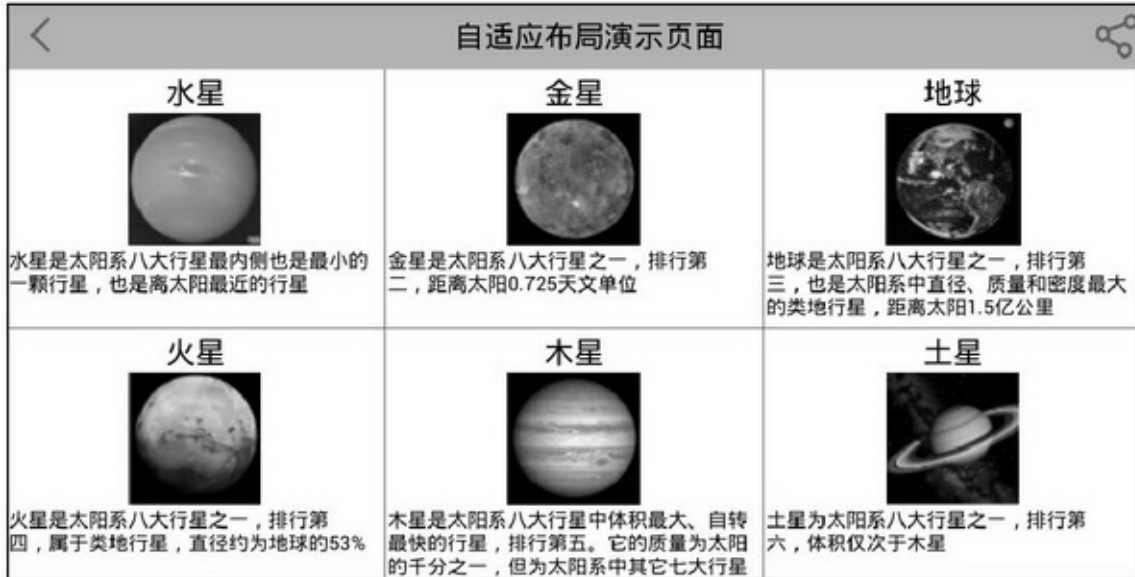
    private void showList() {
        ViewStub vs_list = (ViewStub) findViewById(R.id.vs_list);
        vs_list.inflate();
        ListView lv_hello = (ListView) findViewById(R.id.lv_hello);
        PlanetAdapter adapter = new PlanetAdapter(this, R.layout.item_list,
            Planet.getDefaultList(), Color.WHITE);
        lv_hello.setAdapter(adapter);
    }

    private void showGrid() {
        ViewStub vs_grid = (ViewStub) findViewById(R.id.vs_grid);
        vs_grid.inflate();
        GridView gv_hello = (GridView) findViewById(R.id.gv_hello);
        PlanetAdapter adapter = new PlanetAdapter(this, R.layout.item_grid,
            Planet.getDefaultList(), Color.WHITE);
        gv_hello.setAdapter(adapter);
    }
}

```

16-316-416-316-4

自适应布局演示页面	
	<h3>水星</h3> <p>水星是太阳系八大行星最内侧也是最小的一颗行星，也是离太阳最近的行星</p>
	<h3>金星</h3> <p>金星是太阳系八大行星之一，排行第二，距离太阳0.725天文单位</p>
	<h3>地球</h3> <p>地球是太阳系八大行星之一，排行第三，也是太阳系中直径、质量和密度最大的类地行星，距离太阳1.5亿公里</p>
	<h3>火星</h3> <p>火星是太阳系八大行星之一，排行第四，属于类地行星，直径约为地球的53%</p>
	<h3>木星</h3> <p>木星是太阳系八大行星中体积最大、自转最快的行星，排行第五。它的质量为太阳的千分之一，但为太阳系中其它七大行星质量总和的2.5倍</p>
	<h3>土星</h3> <p>土星为太阳系八大行星之一，排行第六，体积仅次于木星</p>



16-4

16.1.3

Android StudioAppTheme.AppCompat.Light.DarkActionBar
AppApp

res/values/styles.xmlstylestyle

(1)AndroidManifest.xmlapplicationandroid:themeAppactivity
android:theme

(2)ActivitysetContentviewsetTheme(R.style.***)

(3)Dialog

android:gravity

android:background

android:windowBackground

android:windowFramewindowFrame

android:windowNoTitleActionBar

android:windowFullscreen

android:windowIsTranslucent

android:windowIsFloating
android:windowAnimationStyle
android:windowEnterAnimation
android:windowExitAnimation



16-5windowBackground

3android:windowFrameandroid:windowBackground
android:background3
android:windowBackground16-5
android:background16-6
android:windowFrame16-7



16-6background



16-7windowFrame

16.2

Android

16.2.1

C/C++C/C++JavaAndroid

JavaJNIJNIC/C++

AppAppAppAppApp

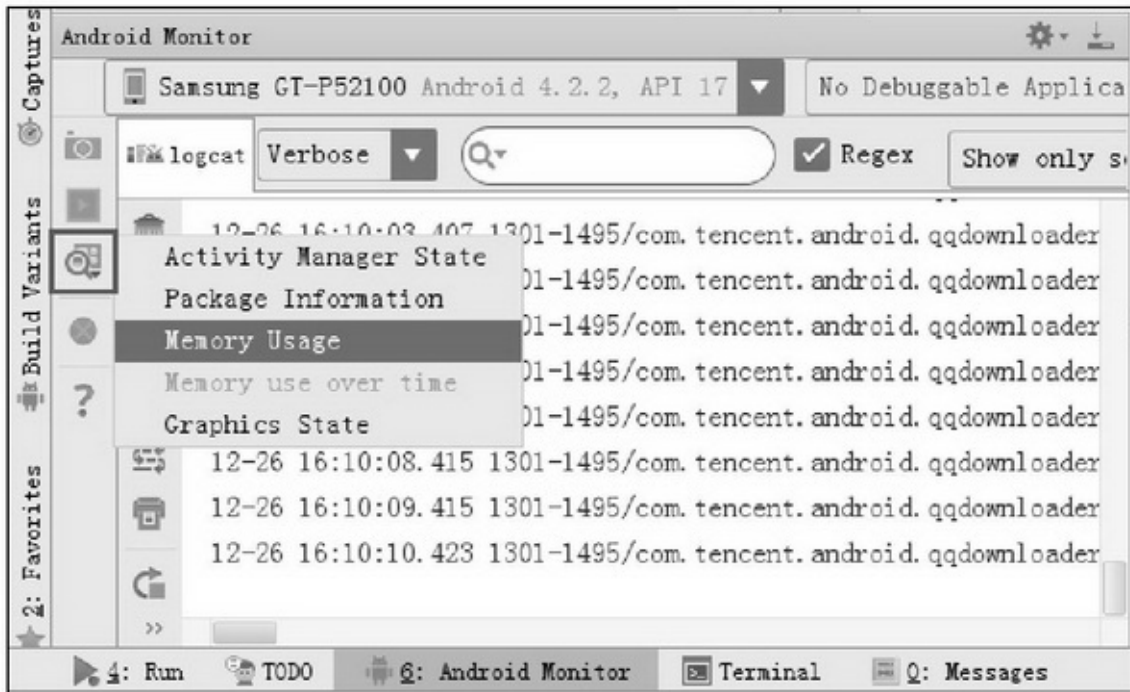
AppApp

Android StudioAndroid Studiologcat3System Information16-

8

Memory Usage16-9performance8573KBApp

Memory Usage16-10



```
activity_main.xml x MainActivity.java x FurtherStudy_2016.12.26_16.11.txt x
12751 kB: qzyd.speed.nethelper (pid 1204)
10770 kB: com.tencent.android.qqdownloader:connect (pid 1675)
10317 kB: com.android.vending (pid 931)
10296 kB: com.mygolbs.mybus (pid 1766)
8573 kB: com.example.performance (pid 2000)
7825 kB: com.tencent.android.qqdownloader:getapkinfo (pid 1537)
5967 kB: com.google.process.gapps (pid 887)
5891 kB: com.google.android.gms (pid 1449)
5732 kB: android.process.media (pid 1139)
```

16-9

```
activity_main.xml x MainActivity.java x FurtherStudy_2016.12.26_16.15.txt x
Applications Memory Usage (kB):
Uptime: 603392 Realtime: 603392

Total PSS by process:
38873 kB: com.android.systemui (pid 484)
25212 kB: com.tencent.android.qqdownloader (pid 1301)
24322 kB: com.tencent.mobileqq (pid 1255)
22212 kB: com.example.performance (pid 2000)
18432 kB: com.microvirt.market (pid 588)
```

16-10App

performance22212KBApp

1.Runnable

HandlerpostDelayedRunnablepostDelayed

App

onDestroy

```

public class RemoveTaskActivity extends AppCompatActivity implements OnClickListener {
    private boolean bRun = false;
    private CheckBox ck_remove;
    private TextView tv_remove;
    private Button btn_remove;
    private String mDesc = "";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_remove_task);

```

```

        ck_remove = (CheckBox) findViewById(R.id.ck_remove);
        tv_remove = (TextView) findViewById(R.id.tv_remove);
        btn_remove = (Button) findViewById(R.id.btn_remove);
        btn_remove.setOnClickListener(this);
        TextView tv_start = (TextView) findViewById(R.id.tv_start);
        tv_start.setText("页面打开时间为 : "+Utils.getNowTime());
    }

    @Override
    public void onClick(View v) {
        if (v.getId() == R.id.btn_remove) {
            if (bRun != true) {
                btn_remove.setText("取消定时任务");
                mHandler.post(mTask);
            } else {
                btn_remove.setText("开始定时任务");
                mHandler.removeCallbacks(mTask);
            }
            bRun = !bRun;
        }
    }

    @Override
    protected void onDestroy() {
        super.onDestroy();
        if (ck_remove.isChecked() == true) {
            mHandler.removeCallbacks(mTask);
        }
    }

    private Handler mHandler = new Handler();
    private Runnable mTask = new Runnable() {
        @Override
        public void run() {
            Intent intent = new Intent(TASK_EVENT);
            LocalBroadcastManager.getInstance(RemoveTaskActivity.this).sendBroadcast(intent);
            mHandler.postDelayed(this, 2000);
        }
    };

    @Override
    public void onStart() {
        super.onStart();
        taskReceiver = new TaskReceiver();
        IntentFilter filter = new IntentFilter(TASK_EVENT);

```

```

        LocalBroadcastManager.getInstance(this).registerReceiver(taskReceiver, filter);
    }

    @Override
    public void onStop() {
        LocalBroadcastManager.getInstance(this).unregisterReceiver(taskReceiver);
        super.onStop();
    }

    private String TASK_EVENT = "com.example.performance.task";
    private TaskReceiver taskReceiver;
    private class TaskReceiver extends BroadcastReceiver {
        @Override
        public void onReceive(Context context, Intent intent) {
            if (intent != null) {
                mDesc = String.format("%s%s 打印了一行测试日志\n", mDesc, Utils.getNowTime());
                tv_remove.setText(mDesc);
            }
        }
    }
}

```

“”16-11“”16-12





16-12

2.

postDelayedApp

3onDestroy

```
public class LogoutServiceActivity extends AppCompatActivity implements OnClickListener {  
    private static final String TAG = "LogoutServiceActivity";
```

```
private boolean bRun = false;
private CheckBox ck_logout;
private Button btn_alarm;
private static TextView tv_alarm;
private PendingIntent pIntent;
private AlarmManager mAlarmManager;
private static String mDesc;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_logout_service);
    ck_logout = (CheckBox) findViewById(R.id.ck_logout);
    tv_alarm = (TextView) findViewById(R.id.tv_alarm);
    btn_alarm = (Button) findViewById(R.id.btn_alarm);
    btn_alarm.setOnClickListener(this);
    Intent intent = new Intent(SERVICE_EVENT);
    pIntent = PendingIntent.getBroadcast(this, 0, intent, PendingIntent.FLAG_UPDATE_CURRENT);
    mAlarmManager = (AlarmManager) getSystemService(ALARM_SERVICE);
    mDesc = "";
    TextView tv_start = (TextView) findViewById(R.id.tv_start);
    tv_start.setText("页面打开时间为 : "+Utils.getNowTime());
}

@Override
protected void onDestroy() {
    super.onDestroy();
    if (ck_logout.isChecked() == true) {
        mAlarmManager.cancel(pIntent);
    }
}

@Override
public void onClick(View v) {
    if (v.getId() == R.id.btn_alarm) {
        if (bRun != true) {
            mAlarmManager.setRepeating(AlarmManager.RTC_WAKEUP,
                System.currentTimeMillis(), 3000, pIntent);
            mDesc = Utils.getNowTime() + " 设置闹钟";
            tv_alarm.setText(mDesc);
            btn_alarm.setText("取消闹钟");
        } else {
            mAlarmManager.cancel(pIntent);
        }
    }
}
```




16-14

16.2.2

App5

(1)Cursor

(2)AdapterconvertView

(3)Bitmaprecycle

(4)Activity

(5)

5

1.

CursorSQLiteContentResolverDownloadManager

Cursorclose

2.

AppListViewGridViewBaseAdaptergetView“ → → →

”

.....

convertView.setTagTag

```

ViewHolder holder = null;
if (convertView == null) {
    holder = new ViewHolder();
    convertView = inflater.inflate(R.layout.list_title, null);
    holder.tv_seq = (TextView) convertView.findViewById(R.id.tv_seq);
    holder.iv_title = (ImageView) convertView.findViewById(R.id.iv_title);
    convertView.setTag(holder);
} else {
    holder = (ViewHolder) convertView.getTag();
}

```

ListView GridView Android 5.0 RecyclerView ViewHolder

3.

Android Bitmap Java SDK Bitmap Factory native Decode Stream
 native JNI Bitmap C/C++ C/C++ Bitmap recycle
 native Recycle native JNI

Bitmap recycle JNI JNI

4.

Handler Android Studio This Handler class should be static or leaks
 might occur Handler post Delayed Runnable Activity

Activity

3

(1) Handler post Delayed remove Callbacks

(2) Android Handler Handler Weak Reference

Handler Activity Handler

Handler

```

private static class MyHandler extends Handler {
    public static WeakReference<HandlerActivity> mActivity;
    public MyHandler(HandlerActivity activity) {
        mActivity = new WeakReference<HandlerActivity>(activity);
    }

    @Override
    public void handleMessage(Message msg) {
        HandlerActivity act = mActivity.get();
        if (act != null) {
            String desc = ProcessUtil.getRunningAppProcessInfo(act);
            act.tv_memory.setText(desc);
        }
    }
}

```

(3)HandlerAppHandlerApplication

AppHandler

5.

AppAndroidApp

onDestroy

16-1

16-1

系统服务的管理器	注销操作说明	注销函数
AlarmManager	取消定时广播	cancel
ConnectivityManager	取消监听网络状态	unregisterNetworkCallback
DownloadManager	移除下载任务	remove
LocationManager	取消监听位置信息的变化	removeUpdates
LocationManager	取消监听定位状态的变化	removeGpsStatusListener
NotificationManager	取消通知	cancel
TelephonyManager	取消监听电话状态	使用 listen 方法注册一个空事件 PhoneStateListener.LISTEN_NONE
Vibrator	取消震动	cancel

16.3

16.3.1

10ThreadThread

(1)

(2)

AsyncTaskexecuteOnExecutorAsyncTaskTHREAD_
POOL_EXECUTOR

AndroidJavaExecutors16-2

16-2

线程池的创建方法	线程池类型	说明
<code>newSingleThreadExecutor</code>	<code>ExecutorService</code>	创建只有单个线程的线程池
<code>newFixedThreadPool</code>	<code>ThreadPoolExecutor</code>	创建线程数量固定的线程池
<code>newCachedThreadPool</code>	<code>ThreadPoolExecutor</code>	创建无个数限制的线程池
<code>newSingleThreadScheduledExecutor</code>	<code>ScheduledThreadPoolExecutor</code>	创建只有单个线程的定时器线程池
<code>newScheduledThreadPool</code>	<code>ScheduledThreadPoolExecutor</code>	创建线程数量固定的定时器线程池

`ThreadPoolExecutor`

`int corePoolSize`

`int maximumPoolSize`

`long keepAliveTime`

`TimeUnit unit`16-3

16-3

TimeUnit 类的时间单位	说明
SECONDS	秒
MILLISECONDS	毫秒
MICROSECONDS	微秒

BlockingQueue<Runnable> workQueue
new LinkedBlockingQueue
<Runnable>()

java.util.concurrent.RejectedExecutionException()

ThreadFactory threadFactory

ThreadPoolExecutor

execute

remove

shutdown

isTerminated

setCorePoolSize

setMaximumPoolSize

setKeepAliveTime

getPoolSize

getActiveCount

16-15~16-18 16-15 16-16(4) 4 16-17

16-18()

performance

普通线程池类型

单线程线程池

17:00:52 当前序号是0
17:00:54 当前序号是1
17:00:56 当前序号是2
17:00:58 当前序号是3
17:01:00 当前序号是4
17:01:02 当前序号是5
17:01:04 当前序号是6
17:01:06 当前序号是7
17:01:08 当前序号是8
17:01:10 当前序号是9
17:01:12 当前序号是10
17:01:14 当前序号是11
17:01:16 当前序号是12
17:01:18 当前序号是13
17:01:20 当前序号是14
17:01:22 当前序号是15
17:01:24 当前序号是16
17:01:26 当前序号是17
17:01:28 当前序号是18
17:01:30 当前序号是19

performance

普通线程池类型

多线程线程池

16:59:52 当前序号是0
16:59:52 当前序号是1
16:59:52 当前序号是2
16:59:52 当前序号是3
16:59:54 当前序号是4
16:59:54 当前序号是5
16:59:54 当前序号是6
16:59:54 当前序号是7
16:59:56 当前序号是8
16:59:56 当前序号是9
16:59:56 当前序号是10
16:59:56 当前序号是11
16:59:58 当前序号是12
16:59:58 当前序号是13
16:59:58 当前序号是14
16:59:58 当前序号是15
17:00:00 当前序号是16
17:00:00 当前序号是18
17:00:00 当前序号是17
17:00:00 当前序号是19

performance

普通线程池类型

无限制线程池

17:02:04 当前序号是0
17:02:04 当前序号是1
17:02:04 当前序号是2
17:02:04 当前序号是3
17:02:04 当前序号是4
17:02:04 当前序号是5
17:02:04 当前序号是6
17:02:04 当前序号是7
17:02:04 当前序号是8
17:02:04 当前序号是9
17:02:04 当前序号是10
17:02:04 当前序号是11
17:02:04 当前序号是12
17:02:04 当前序号是13
17:02:04 当前序号是14
17:02:04 当前序号是15
17:02:04 当前序号是16
17:02:04 当前序号是17
17:02:04 当前序号是18
17:02:04 当前序号是19



16-18

16.3.2

()

AndroidnewScheduledThreadPool

newSingleThreadScheduledExecutor16-2

ScheduledThreadPoolExecutor

int corePoolSize

ThreadFactory threadFactoryThreadFactorynewThread

RunnableThreadDefaultThreadFactory

RejectedExecutionHandler handler

ScheduledExecutorServiceThreadPoolExecutor

schedule

scheduleAtFixedRate

scheduleWithFixedDelay

scheduleAtFixedRatescheduleWithFixedDelay

16-1916-2016-1916-20(3)33



performance

定时器线程池类型 多线程定时器固定速率

17:23:55 当前序号是0

17:23:55 当前序号是1

17:23:55 当前序号是2

17:23:58 当前序号是0

17:23:58 当前序号是1

17:23:58 当前序号是2

17:24:01 当前序号是0

17:24:01 当前序号是1

17:24:01 当前序号是2

16.4

AppAppApp

16.4.1

AndroidIntent.ACTION_BATTERY_CHANGEDApp
AndroidManifest.xmlregisterReceiver16-4

16-4

BatteryManager 类的字段名称	字段获取方法	说明
EXTRA_SCALE	getIntExtra	电量刻度, 通常是 100
EXTRA_LEVEL	getIntExtra	当前电量
EXTRA_STATUS	getIntExtra	当前状态
BATTERY_STATUS_UNKNOWN	当前状态取值	未知
BATTERY_STATUS_CHARGING	当前状态取值	正在充电
BATTERY_STATUS_DISCHARGING	当前状态取值	正在断电
BATTERY_STATUS_NOT_CHARGING	当前状态取值	不在充电
BATTERY_STATUS_FULL	当前状态取值	电量充满
EXTRA_HEALTH	getIntExtra	健康程度
BATTERY_HEALTH_UNKNOWN	健康程度取值	未知
BATTERY_HEALTH_GOOD	健康程度取值	良好
BATTERY_HEALTH_OVERHEAT	健康程度取值	过热
BATTERY_HEALTH_DEAD	健康程度取值	坏了
BATTERY_HEALTH_OVER_VOLTAGE	健康程度取值	短路
BATTERY_HEALTH_UNSPECIFIED_FAILURE	健康程度取值	未知错误
BATTERY_HEALTH_COLD	健康程度取值	冷却
EXTRA_VOLTAGE	getIntExtra	当前电压
EXTRA_PLUGGED	getIntExtra	当前电源
0	当前电源取值	电池
BATTERY_PLUGGED_AC	当前电源取值	充电器
BATTERY_PLUGGED_USB	当前电源取值	USB
BATTERY_PLUGGED_WIRELESS	当前电源取值	无线
EXTRA_TECHNOLOGY	getStringExtra	当前技术, 比如返回 Li-ion 表示锂电池
EXTRA_TEMPERATURE	getIntExtra	当前温度
EXTRA_PRESENT	getBooleanExtra	是否提供电池

16-2116-2216-2116-22

performance

18:17:07 : 收到广播: android.intent.action.BATTERY_CHANGED
电量刻度=100
当前电量=18
当前状态=正在充电
健康程度=良好
当前电压=3813
当前电源=USB
当前技术=Li-ion
当前温度=31
是否提供电池=是

performance

18:17:23 : 收到广播: android.intent.action.BATTERY_CHANGED
电量刻度=100
当前电量=18
当前状态=不在充电
健康程度=良好
当前电压=3776
当前电源=电池
当前技术=Li-ion
当前温度=31
是否提供电池=是

16-22

16.4.2

AppApp

(1)

(2)JSONXML

(3)GPS

(4)0

(5)

(6)

(7)

(8)

(9)

ServiceService“”

AppIntent.ACTION_SCREEN_ON

Intent.ACTION_SCREEN_OFF

3

(1)AndroidManifest.xml

(2)ActivityApplicationonCreate

(3)Application


```
performance
18:17:39 : 收到广播: android.intent.action.SCREEN_OFF
这是屏幕关闭事件, 可在此暂停耗电操作
18:17:43 : 收到广播: android.intent.action.SCREEN_ON
这是屏幕点亮事件, 可在此开启日常操作
```

16.5

AppHTTPJSONXML

“”

16.5.1

4“”“” → “SD”3“” → “SD” → “”

AppSDSKSDSDURL3App

16-2416-2516-2416-25

performance

占位加载

列表加载

月坛公园的玫瑰花盛开啦





16-25

App

(1)ImageView

(2)AlphaAnimation

(3)

(4)

(5)SDSD

(6)HTTPHTTP

(7)Handler

(8)

(9)HandlerBitmap

AndroidApp

16.5.2LRU

FIFOLRUFIFO

LinkedList

LRUJavaAndroidLruCacheLRULruCache

App

LruCache

resize

put

get

remove

evictAll

size

maxSize

snapshot

LruCacheJava

(1)LruCachecontainsgetgetnull

(2)LruCachesnapshotMap

LruCache

```

public class LruCacheActivity extends AppCompatActivity implements OnClickListener {
    private TextView tv_lru_cache;
    private LruCache<String, String> mLanguageLru;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_lru_cache);
        tv_lru_cache = (TextView) findViewById(R.id.tv_lru_cache);
        findViewById(R.id.btn_android).setOnClickListener(this);
        findViewById(R.id.btn_ios).setOnClickListener(this);
        findViewById(R.id.btn_java).setOnClickListener(this);
        findViewById(R.id.btn_cpp).setOnClickListener(this);
        findViewById(R.id.btn_python).setOnClickListener(this);
        findViewById(R.id.btn_net).setOnClickListener(this);
        findViewById(R.id.btn_php).setOnClickListener(this);
        findViewById(R.id.btn_perl).setOnClickListener(this);
        mLanguageLru = new LruCache<String, String>(5);
    }

    @Override
    public void onClick(View v) {
        String language = ((Button) v).getText().toString();
        mLanguageLru.put(language, Utils.getNowTime());
        printCache();
    }

    private void printCache() {
        String desc = "";
        Map<String, String> cache = mLanguageLru.snapshot();
        for (Map.Entry<String, String> item : cache.entrySet()) {
            desc = String.format("%s%s 最后一次更新时间为%s\n",
                desc, item.getKey(), item.getValue());
        }
        tv_lru_cache.setText(desc);
    }
}

```

16-26~16-29
 16-26LRUAndroid
 ANDROIDAndroid
 AndroidiOS
 16-27

performance

ANDROID	IOS	JAVA	C/C++
PYTHON	.NET	PHP	PERL

Android 最后一次更新时间为17:05:01
 iOS 最后一次更新时间为17:05:03
 JAVA 最后一次更新时间为17:05:05
 C/C++ 最后一次更新时间为17:05:07
 Python 最后一次更新时间为17:05:09

16-26LRU

performance

ANDROID	IOS	JAVA	C/C++
PYTHON	.NET	PHP	PERL

iOS 最后一次更新时间为17:05:03
 JAVA 最后一次更新时间为17:05:05
 C/C++ 最后一次更新时间为17:05:07
 Python 最后一次更新时间为17:05:09
 Android 最后一次更新时间为17:05:45

16-27Android

PHPPHPPHPiOSJAVA16-28JAVAJavaJava

performance

ANDROID	IOS	JAVA	C/C++
PYTHON	.NET	PHP	PERL

JAVA 最后一次更新时间为17:05:05
C/C++ 最后一次更新时间为17:05:07
Python 最后一次更新时间为17:05:09
Android 最后一次更新时间为17:05:45
PHP 最后一次更新时间为17:07:15

16-28PHPLRU

performance

ANDROID	IOS	JAVA	C/C++
PYTHON	.NET	PHP	PERL

C/C++ 最后一次更新时间为17:05:07
Python 最后一次更新时间为17:05:09
Android 最后一次更新时间为17:05:45
PHP 最后一次更新时间为17:07:15
JAVA 最后一次更新时间为17:07:43

16-29JavaLRU

16.5.3

3

(1)AndroidManifest.xml

```
<!-- --> <uses-permission  
android:name="android.permission.INTERNET" /> <!-- SD --> <uses-  
permission  
android:name="android.permission.WRITE_EXTERNAL_STORAGE" />  
<uses-permission  
android:name="android.permission.READ_EXTERNAL_STORAGE" /> <uses-  
permission  
android:name="android.permission.MOUNT_UNMOUNT_FILESYSTEMS" />
```

(2)ListViewListView

(3)

16-3016-31

performance

占位加载

列表加载

月坛公园的玫瑰花盛开啦





16-31

“” → “SD” → “” 316-32

16-33

performance

占位加载

列表加载

月坛公园的玫瑰花盛开啦





16-33

16-34 ListView 16-35

16-36 16-37

performance

占位加载

列表加载

欢迎来到风景秀美的鼓浪屿

1



2



performance

占位加载

列表加载

欢迎来到风景秀美的鼓浪屿




1

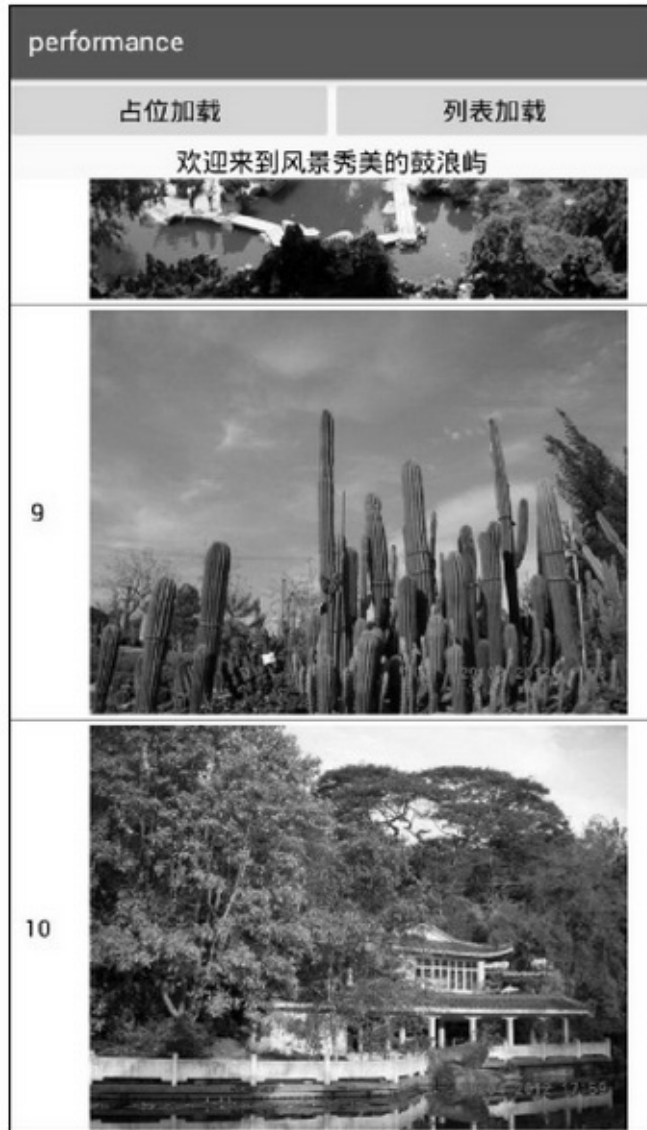


2



performance

占位加载	列表加载
欢迎来到风景秀美的鼓浪屿	
	
9	
10	



16-37

show

```

ImageCacheConfig config = new ImageCacheConfig.Builder()
    .setBeginImage(R.drawable.load_default) //设置初始图片
    .setErrorImage(R.drawable.load_error) //设置出错图片
    .setCacheStyle(ImageCacheConfig.LRU) //设置 LRU 缓存算法
    .setFadeInterval(2000).build(); //设置渐变动画的持续时间
//初始化缓存设置，并把地址为 file 的图片显示到控件 iv_cache 上
mCache.initConfig(config).show(file, iv_cache);

```

```

public class ImageCache {
    private final static String TAG = "ImageCache";
    //内存中的图片缓存
    private HashMap<String, Bitmap> mImageMap = new HashMap<String, Bitmap>();
    //uri 与视图控件的映射关系
    private HashMap<String, ImageView> mViewMap = new HashMap<String, ImageView>();
    //缓存队列，采用 FIFO 先进先出策略，需操作队列首尾两端，故采用双端队列
    private LinkedList<String> mFifoList = new LinkedList<String>();
    //缓存队列，采用 LRU 近期最少使用策略，Android 专门提供了 LruCache 实现该算法
    private LruCache<String, Bitmap> mImageLru;

```

```

    private ImageCacheConfig mConfig;
    private String mDir = "";
    private ThreadPoolExecutor mPool;
    private static Handler mMyHandler;
    private static ImageCache mCache = null;
    private static Context mContext;

    public static ImageCache getInstance(Context context) {
        if (mCache == null) {
            mCache = new ImageCache();
            mCache.mContext = context;
        }
        return mCache;
    }

    public ImageCache initConfig(ImageCacheConfig config) {
        mCache.mConfig = config;
        mCache.mDir = mCache.mConfig.mDir;
        if (mCache.mDir == null || mCache.mDir.length() <= 0) {
            mCache.mDir = Environment.getExternalStorageDirectory() + "/image_cache";
        }
        Log.d(TAG, "mDir=" + mCache.mDir);
        File dir = new File(mCache.mDir);
        if (dir.exists() != true) {
            dir.mkdirs(); // 若目录不存在，则先创建新目录
        }
        mCache.mPool = (ThreadPoolExecutor) Executors.newFixedThreadPool(
mCache.mConfig.mThreadCount);
        mCache.mMyHandler = new MyHandler((Activity)mCache.mContext);
        if (mCache.mConfig.mCacheStyle == ImageCacheConfig.LRU) {
            mImageLru = new LruCache(mCache.mConfig.mMemoryFileCount);
        }
        return mCache;
    }

    public void show(String uri, ImageView iv) {
        if (mConfig.mBeginImage != 0) {
            iv.setImageResource(mConfig.mBeginImage);
        }
        mViewMap.put(uri, iv);
        if (checkExist(uri) == true) {
            mCache.render(uri, getBitmap(uri));
        } else {

```

```

String path = getFilePath(uri);
if ((new File(path)).exists() == true) {
    Bitmap bitmap = ImageUtil.openBitmap(path);
    if (bitmap != null) {
        mCache.render(uri, bitmap);
    } else {
        mPool.execute(new MyRunnable(uri));
    }
} else {
    mPool.execute(new MyRunnable(uri));
}
}

private boolean checkExist(String uri) {
    if (mCache.mConfig.mCacheStyle == ImageCacheConfig.LRU) {
        return (mImageLru.get(uri) == null) ? false : true;
    } else {
        return mImageMap.containsKey(uri);
    }
}

private Bitmap getBitmap(String uri) {
    if (mCache.mConfig.mCacheStyle == ImageCacheConfig.LRU) {
        return mImageLru.get(uri);
    } else {
        return mImageMap.get(uri);
    }
}

private String getFilePath(String uri) {
    String file_path = String.format("%s/%d.jpg", mDir, uri.hashCode());
    return file_path;
}

private static class MyHandler extends Handler {
    public static WeakReference<Activity> mActivity;
    public MyHandler(Activity activity) {
        mActivity = new WeakReference<Activity>(activity);
    }
}

@Override
public void handleMessage(Message msg) {

```

```

        Activity act = mActivity.get();
        if (act != null) {
            ImageData data = (ImageData) (msg.obj);
            if (data != null && data.bitmap != null) {
                mCache.render(data.uri, data.bitmap);
            } else {
                mCache.showError(data.uri);
            }
        }
    }
}

private class MyRunnable implements Runnable {
    private String mUri;
    public MyRunnable(String uri) {
        mUri = uri;
    }

    @Override
    public void run() {
        Activity act = MyHandler.mActivity.get();
        if (act != null) {
            Bitmap bitmap = ImageHttp.getImage(mUri);
            if (bitmap != null) {
                if (mConfig.mSize != null) {
                    bitmap = Bitmap.createScaledBitmap(bitmap, mConfig.mSize.x,
mConfig.mSize.y, false);
                }
                ImageUtil.saveBitmap(getFilePath(mUri), bitmap);
            }
            ImageData data = new ImageData(mUri, bitmap);
            Message msg = mMyHandler.obtainMessage();
            msg.obj = data;
            mMyHandler.sendMessage(msg);
        }
    }
};

private void render(String uri, Bitmap bitmap) {
    ImageView iv = mViewMap.get(uri);
    if (mConfig.mFadeInterval <= 0) {
        iv.setImageBitmap(bitmap);
    } else {

```

```

//内存中已有图片的就直接显示，不再展示淡入淡出动画
if (checkExist(uri) == true) {
    iv.setImageBitmap(bitmap);
} else {
    iv.setAlpha(0.0f);
    AlphaAnimation alphaAnimation = new AlphaAnimation(0.0f, 1.0f);
    alphaAnimation.setDuration(mConfig.mFadeInterval);
    alphaAnimation.setFillAfter(true);
    iv.setImageBitmap(bitmap);
    iv.setAlpha(1.0f);
    iv.setAnimation(alphaAnimation);
    alphaAnimation.start();
    mCache.refreshList(uri, bitmap);
}
}
}

private synchronized void refreshList(String uri, Bitmap bitmap) {
    if (mCache.mConfig.mCacheStyle == ImageCacheConfig.LRU) {
        mImageLru.put(uri, bitmap);
    } else {
        if (mFifoList.size() >= mConfig.mMemoryFileCount) {
            String out_uri = mFifoList.pollFirst();
            mImageMap.remove(out_uri);
        }
        mImageMap.put(uri, bitmap);
        mFifoList.addLast(uri);
    }
}

private void showError(String uri) {
    ImageView iv = mViewMap.get(uri);
    if (mConfig.mErrorImage != 0) {
        iv.setImageResource(mConfig.mErrorImage);
    }
}

public void clear() {
    for (Map.Entry<String, Bitmap> item_map : mImageMap.entrySet()) {
        Bitmap bitmap = item_map.getValue();
        bitmap.recycle();
    }
    if (mImageLru != null) {

```

```

        mImageLru.evictAll();
    }
    mCache = null;
}
}

```

16.6

App0000“”App

LRU

4

(1)

(2)

(3)

(4)