



Android

# Android手机 应用程序开发

Android Shouji Yingyong Chengxu Kaifa

主 审 李明革  
主 编 李 季 张永华  
副主编 孙佳帝 王素华



北 京 科 技 大 学 出 版 社  
NORTH CHINA UNIVERSITY OF TECHNOLOGY PRESS

# Android 手机应用程序开发

主 审 李明革  
主 编 李 季 张永华  
副主编 孙佳帝 王素华

东北师范大学出版社

长 春

---

图书在版编目(CIP)数据

Android 手机应用程序开发/李季,张永华主编. —  
长春:东北师范大学出版社,2014.8  
ISBN 978-7-5681-0163-9

I. ①A… II. ①李…②张… III. ①移动电话机—应  
用软件—程序设计—高等职业教育—教材 IV.  
①TN929.53

中国版本图书馆 CIP 数据核字(2014)第 192290 号

---

责任编辑:刘远舰 封面设计:宣是设计  
责任校对:黄小凤 责任印制:刘兆辉

---

东北师范大学出版社出版发行  
长春净月经济开发区金宝街 118 号(邮政编码:130117)  
电话:0431-85687213 010-82893515  
传真:0431-85691969 010-82896571  
网址:<http://www.nenup.com>

东北师范大学出版社激光照排中心制版  
北京京华虎彩印刷有限公司印刷  
北京市顺义区顺平路南彩段 5 号(邮政编码:101300)  
2014 年 8 月第 1 版 2014 年 8 月第 1 次印刷  
幅面尺寸:185 mm×260 mm 印张:17.75 字数:358 千

---

定价:36.00 元

# 前 言

本教材以项目案例为导引,以项目开发实现为主线,通过对项目案例的剖析,讲解了 Android 技术软件开发所应具备的基础知识,在完成项目开发的同时应掌握 Android 软件开发所需要的技术,为下一步进行工程项目的开发实训打下良好的技术基础。

## 教材主要内容:

本教材包含大量实例,以由浅入深、循序渐进的方式介绍了 Android 应用开发的相关基础知识与技能。结构层次清晰、语言简洁、案例简单、易于理解。本教材内容共分为 8 个项目和 1 个附录,项目 1 详细讲解了开发 Android 应用程序的步骤以及工程目录结构;项目 2 介绍了 Android 系统应用中 5 个常用的界面布局;项目 3 介绍了 Android 系统中常用的基本组件;项目 4 详细讲解了 Android 事件处理机制;项目 5 介绍了适配器、ListView 列表等高级组件;项目 6 讲解了 GridView 网格组件、Gellery 画廊组件的应用;项目 7 主要介绍了菜单、数据存储和对话框的应用;项目 8 主要介绍了 AsyncTask 异步任务、HTTP 网络访问技术及 JSON 格式字符串组装与解析;附录详细讲解了 Win7 系统下 Android4.4.2 开发环境的搭建过程。

## 教材特点:

本教材采用项目驱动方式设计,通过对项目的分析,引出知识点和技能点,根据对知识点和技能点的学习与掌握,来完成项目的实施。最后,通过对拓展项目的练习,提高对技能的训练与掌握,即双线并行设计。教材的设计符合高等职业教育行动导向思想,重在使学生获得知识的过程与实践技能相对应,建立学习项目与知识、技能的联系,增强学生的直观体验,激发学生的学习兴趣,培养学生编程思想,使学生达到学以致用目的。

本教材主要由长春职业技术学院的老师共同编写。本教材项目 1、5 由王素华编写,项目 3、4 由孙佳帝编写,项目 2、6 由张永华编写,项目 7、8 及附录由李季编写,全书由李季进行统稿,李明革主审。

本教材主要面向高职软件技术专业的学生,或有一定 Java 基础爱好 Android 开发的初学者。

由于作者水平有限,编写时间仓促,书中难免有疏漏之处,敬请读者批评指正。

编 者

2014 年 5 月于长春

# 目 录

项目 1 编写第一个手机应用程序 .....	1
1.1 项目描述 .....	1
1.2 项目分析 .....	1
1.3 知识准备 .....	1
1.4 项目实现 .....	6
1.5 拓展项目 .....	9
1.6 课后练习 .....	9
项目 2 设计用户登录界面 .....	10
2.1 项目描述 .....	10
2.2 项目分析 .....	10
2.3 知识准备 .....	13
2.4 项目实现 .....	27
2.5 拓展项目 .....	32
2.6 课后练习 .....	33
项目 3 实现用户信息注册 .....	34
3.1 项目描述 .....	34
3.2 项目分析 .....	34
3.3 知识准备 .....	35
3.4 项目实现 .....	43
3.5 拓展项目 .....	52
3.6 课后练习 .....	53
项目 4 计算器的设计与实现 .....	54
4.1 项目描述 .....	54
4.2 项目分析 .....	54
4.3 知识准备 .....	55
4.4 项目实现 .....	61
4.5 拓展项目 .....	73
4.6 课后练习 .....	75

项目 5 实现打电话和发短信功能 .....	76
5.1 项目描述 .....	76
5.2 项目分析 .....	77
5.3 知识准备 .....	77
5.4 项目实施 .....	94
5.5 拓展项目 .....	102
5.6 课后练习 .....	102
项目 6 实现数字媒体展示 .....	103
6.1 项目描述 .....	103
6.2 项目分析 .....	104
6.3 知识准备 .....	104
6.4 项目实施 .....	121
6.5 拓展项目 .....	137
6.6 课后练习 .....	138
项目 7 实现用户信息的管理 .....	139
7.1 项目描述 .....	139
7.2 项目分析 .....	140
7.3 知识准备 .....	141
7.4 项目实施 .....	187
7.5 拓展项目 .....	208
7.6 课后练习 .....	209
项目 8 网上购物手机客户端的设计 .....	210
8.1 项目描述 .....	210
8.2 项目分析 .....	210
8.3 知识准备 .....	211
8.4 项目实施 .....	238
8.5 拓展项目 .....	269
8.6 课后练习 .....	269
附录 搭建 Android 开发环境 (Win7 系统) .....	270

# 项目1 编写第一个手机应用程序

## 学习目标

知识目标:

1. 掌握 Android 应用程序的创建过程;
2. 理解工程结构目录;
3. 掌握工程清单文件。

技能目标:

1. 能够创建 Android 工程;
2. 能够创建 AVD 模拟器;
3. 能够进行应用程序测试与调试;
4. 能够对清单文件进行修改。

## 1.1 项目描述

本项目主要创建 HelloWorld 应用。运行结果如图 1-1 所示。

## 1.2 项目分析

本项目没有难度,通过对项目的创建、运行,使读者了解 Android 程序开发的过程,了解 Android 工程的目录结构,理解相关文件的用途。



图 1-1 程序运行结果

## 1.3 知识准备

以 HelloWorld 工程为例子。通过解析 HelloWorld 工程的目录结构,让读者对 Android 程序有一个整体全面的认识,工程目录结构如图 1-2 所示。

### 1.3.1 src 目录

src 目录中存放项目的源代码,其内部结构会根据用户所声明的包名称自动组织,该目录的组织方式为 src/cvit.com/HelloWordActivity.java,其中 cvit.com 为包名,HelloWordActivity.java 是编写的源文件名称。软件开发人员在项目开发过程中,大部分时间是对该目录下的源代码文件进行编写。

HelloWorldActivity 类是 Activity 的子类,该类要重写 onCreate() 方法。setContentView() 方法是给 Activity 设置可以显示的视图(View),装载指定的布局文件。该类运行时自动执行 onCreate() 方法。其代码如下:

```
package cvit.com;
import android.app.Activity;
import android.os.Bundle;
public class HelloWorldActivity extends Activity
{
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        //设置 Activity 的布局文件
        setContentView(R.layout.main);
    }
}
```

### 提示

Activity 是 Android 程序的四大组件之一。Activity 是 Android 程序的表示层。一般可以认为程序的每一个显示屏幕就是一个 Activity。Activity 中的所有操作都与用户密切相关,是一个负责与用户交互的组件,可以通过 setContentView(View) 来显示指定控件。

当程序第一次运行时,用户就会看到这个 Activity,这个 Activity 可以通过启动其他的 Activity 进行相关操作。

当启动其他的 Activity 时,这个当前的 Activity 将会停止,新的 Activity 将会压入栈中,同时获取用户焦点,这时就可在这个 Activity 上操作了。栈的原则是先进后出,当用户按下 back 键时,当前的这个 Activity 销毁,前一个 Activity 重新恢复。

在大部分情况下,每个 Android 应用都将运行在自己的 Linux 进程当中。当这个应用的某些代码需要执行时,进程就会被创建,并且保持运行,直到该进程不再需要,而系统需要释放它所占用的内存为其他应用所用时,才停止。因此 Activity 生命周期分成三个阶段,通过七个方法实现。

七个方法如下:

```
void onCreate(Bundle savedInstanceState): 创建
void onStart(): 开始
void onRestart(): 重新开始
void onResume(): 继续
void onPause(): 暂停
void onStop(): 停止
void onDestroy(): 销毁
```

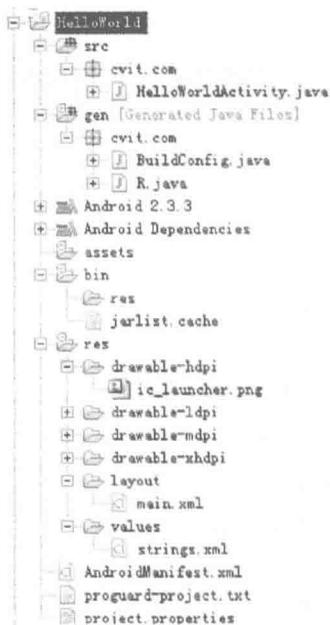


图 1-2 HelloWorld 工程目录结构



三个阶段如下:

**开始 Activity:**在这个阶段依次执行三个生命周期的方法,分别是 onCreate()、onStart()、onResume()方法。

**Activity 重新获得焦点:**依次执行三个生命周期的方法,分别是 onRestart()、onStart()、onResume()方法。

**关闭 Activity:**当 Activity 被关闭时系统会依次执行三个生命周期的方法,分别是 onPause()、onStop()、onDestroy()。

当一个 Activity 实例被创建、销毁或者启动另外一个 Activity 时,它在这几种状态之间进行转换,这种转换的发生依赖于用户程序的动作。图 1-3 说明了 Activity 在不同状态间转换的时机和条件。

### 1.3.2 gen 目录

gen 目录下的文件全部都是 ADT 自动生成的,一般并不需要去修改。实际上该目录下主要定义了一个 R.java 文件,此文件是应用程序资源 Resource 类,包括组件 id、Layout、Strings、Drawable,以方便在整个应用程序中对资源的使用和统一管理。项目中用户界面、字符串、图片等资源都会在该类中创建其唯一的 id,当项目中使用这些资源时,可以通过该 id 得到资源的引用。

R.java 文件代码如下:

```
package cvit.com;
public final class R {
    public static final class attr {
    }
    public static final class drawable {
        public static final int ic_launcher=0x7f020000;
    }
    public static final class layout {
        public static final int main=0x7f030000;
    }
    public static final class string {
        public static final int app_name=0x7f040001;
        public static final int hello=0x7f040000;
    }
}
```

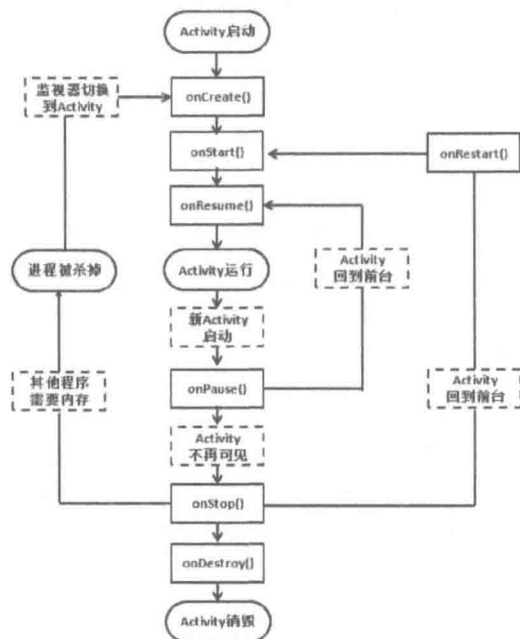


图 1-3 Activity 生命周期

### 1.3.3 Android2.x

该目录中存放的是项目支持的 jar 包,同时其中还包含项目打包时需要的 META-INF 目录。

### 1.3.4 bin 目录

bin 目录中的文件都是自动生成的,包括系统在编译的时候,会生成的 apk 和一些编译成的文件。

### 1.3.5 assets 目录

该目录用于存放项目相关的资源文件,如文本文件、视频等。在程序中可以使用 getResources().getAssets().open("text.txt") 方式得到资源文件的输入流 InputStream 对象,这样就可读取文件内容。

### 1.3.6 res 目录

这个目录中存放的是资源文件,资源文件的统一管理,也是 Android 系统的一大特色。该目录主要可以存放一些图标、视图布局文件和应用中用到的文字信息。

#### 1. drawable - \* dpi 文件夹

该类文件夹主要用来存放图片,将图片按分辨率的高低放入不同的目录。其中 drawable-hdpi 用来存放高分辨率的图标,drawable-mdpi 用来存放中等分辨率的图片,drawable-ldpi 用来存放低分辨率的图片。

#### 2. layout 文件夹

该文件夹用来存放视图布局文件和用户设计的界面信息。在 Activity 类的 onCreate() 方法中通过方法 setContentView(R.id.main) 设置,实现 Activity 类与 main.xml 布局的关联。main.xml 布局文件代码如下:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical"
    >
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/hello"
    />
</LinearLayout>
```

#### 3. values 文件夹

该文件夹用来存放当前应用程序中使用到的字符串——以“键值对”的形式存放,如应用程序的名称、按钮等提示文本。这样做的好处是将文本字符串与代码分离,便于国际化的实现。

string.xml 文件代码如下:

```
<?xml version = "1.0" encoding = "utf - 8" ? >
<resources >
    <string name = "hello" >Hello World, HelloWorldActivity! </string >
    <string name = "app_name" >HelloWorld </string >
</resources >
```

### 1.3.7 AndroidManifest.xml 清单文件

每个应用程序都会对应一个 AndroidManifest.xml 清单文件,为 Android 系统提供应用的基本信息。该文件向 Android 操作系统描述了本程序所包括的组件,所实现的功能,能处理的数据,要请求的资源,等等。系统在运行之前必须知道这些信息,否则无法运行。

#### 1. AndroidManifest.xml 的功能

- (1)命名应用程序的 Java 应用包,这个包名用来唯一标识应用程序;
- (2)描述应用程序的组件,对实现每个组件和公布其功能的类进行命名,这些声明使得 Android 系统了解这些组件以及它们在什么条件下可以被启动;
- (3)决定哪个组件运行在哪个进程里;
- (4)声明应用程序必须具备的权限,用以访问受保护的 API,以及和其他进程的交互;
- (5)声明应用程序其他的必备权限,用以组件之间的交互。

#### 2. AndroidManifest.xml 解析

```
<?xml version = "1.0" encoding = "utf - 8" ? >
<manifest xmlns:android = "http://schemas.android.com/apk/res/android"
    package = "cvit.com"
    android:versionCode = "1"
    android:versionName = "1.0" >
    <uses - sdk android:minSdkVersion = "10" / >
    <application
        android:icon = "@drawable/ic_launcher"
        android:label = "@string/app_name"
        >
        <activity
            android:name = ". HelloWorldActivity"
            android:label = "@string/app_name"
            >
            <intent - filter >
                <action android:name = "android.intent.action.MAIN" / >
                <category android:name = "android.intent.category.LAUNCHER" / >
            </intent - filter >
        </activity >
    </application >
```

</manifest >

#### (1) < manifest > 元素

< manifest > 元素是 AndroidManifest.xml 的根元素;xmlns:android 是指该文件的命名空间;package 属性是 Android 应用所在的包;android:versionCode 用来指定应用的版本号,如果应用不断升级,则需要修改这个值;android:versionName 是版本的名称。

#### (2) < application > 元素

< application > 元素是一个很重要的元素,开发组件都会在此下定义。< application > 元素的 icon 属性是用来设定应用的图标,其中@drawable/ic\_launcher 的意思是指向 drawable-mdpi 文件夹下的 launcher.png 图片。而@drawable/ic\_launcher 写法是图片在 R.java 文件中自动生成的引用。< application > 元素的 label 属性用来设定应用的名称,其中@string/app\_name 和上述的一样,也是 R.java 文件中的 string 静态内部类下的 app\_name。

#### (3) < activity > 元素

< activity > 元素的作用是注册一个 Activity 信息。当在创建“HelloWorld”这个项目时,指定了 Created Activity 属性为 HelloWorld,然后 ADT 在生成项目时自动创建了一个 Activity,就是 HelloWorldActivity.java。

< activity > 元素的 name 属性指定的是 Activity 的类名,其中“. HelloWorldActivity”中的“.”指的是 < manifest > 元素中的 package 属性中指定的包,所以“. HelloWorldActivity”就相当于 cvit.com.HelloWorldActivity.java。

#### (4) < intent - filter > 元素

< intent - filter > 如果直接翻译成中文是“意图过滤器”,组件通过 < intent - filter > 告诉程序它们所具备的功能,就是能响应的意图的类型。在 intent 中设置 action, data, category 之后,在对应的 < intent-filter > 中设置相同的属性即可通过过滤被 Activity 调用。

代码、文件、运行图之间的关系如图 1-4 所示。

### 1.3.8 project.properties 文件

应用要求运行的最低 Android 版本。

## 1.4 项目实施

### 1. 创建 HelloWorld 工程

ADT 插件提供了一个创建新工程的向导,以使开发者能够快速的创建一个新的 Eclipse 工程或者为现有代码创建一个新工程。创建步骤如下:

第一步,在 Eclipse 中,选择 File > New > Android Project 菜单,弹出的对话框如图 1-5 所示,在图 1-5 中给工程起 HelloWorld 工程名,之后单击【Next】命令按钮,进入选择 SDK 版本窗口,如图 1-6 所示;

第二步,选择 Android 构建目标 2.3.3,SDK 版本,如图 1-6 所示,之后单击【Next】命令按钮,进入如图 1-7 所示对话框设置 Activity 类信息;

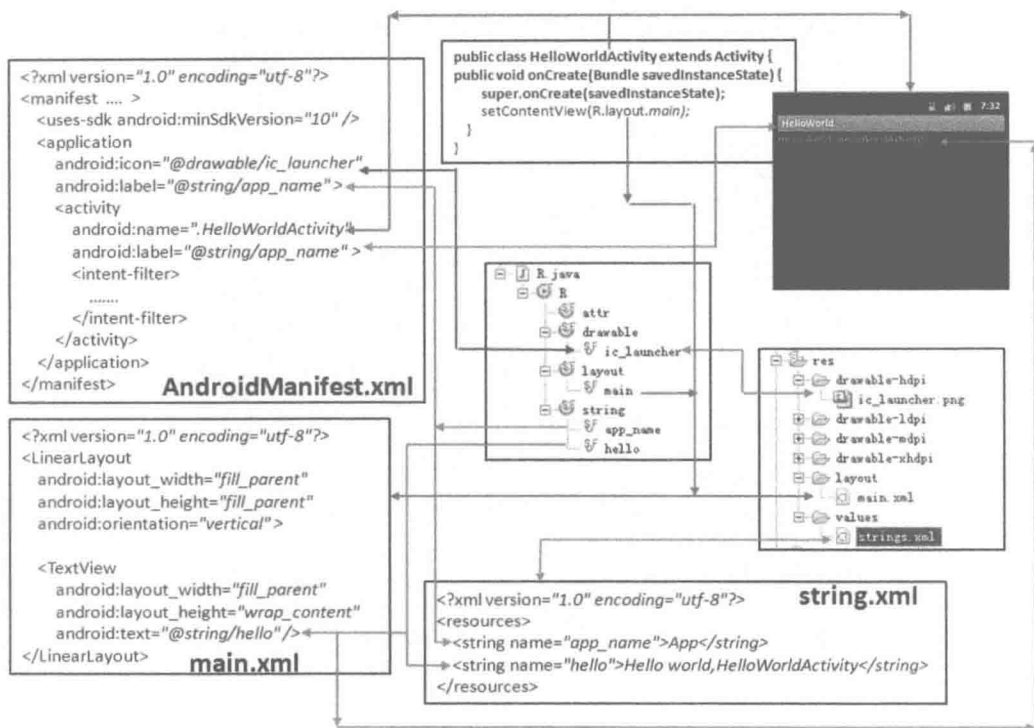


图 1-4 代码、文件、运行图之间的关系

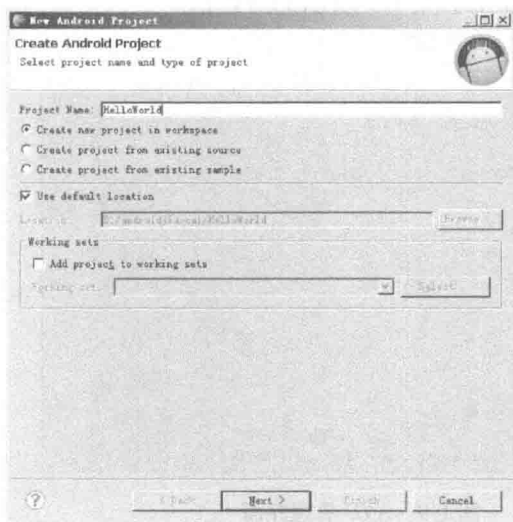


图 1-5 创建工程对话框

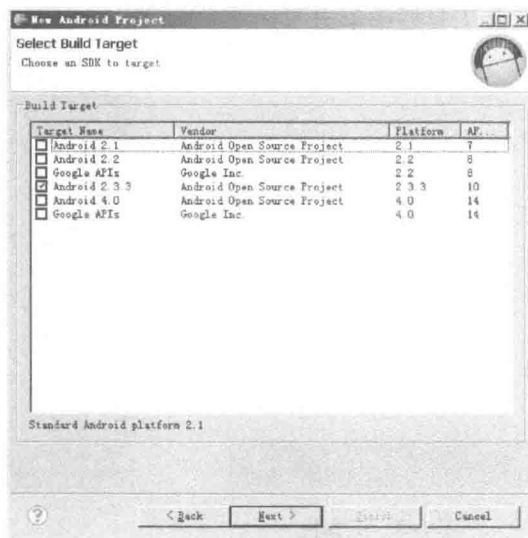


图 1-6 选择 SDK 版本窗口

第三步,在 Package Name 中设置包名为 `cvit.com`,如图 1-7 所示,单击【Finish】按钮完成创建。

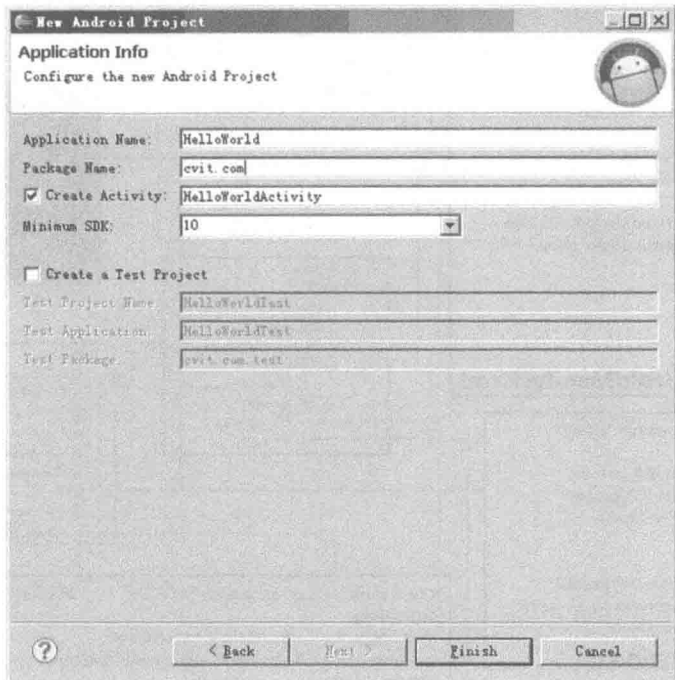


图 1-7 设置 Activity 类信息

## 2. 创建 AVD 模拟器

在工具栏中单击图标 , 如图 1-8 所示, 将弹出如图 1-9 所示的对话框。



图 1-8 工具栏

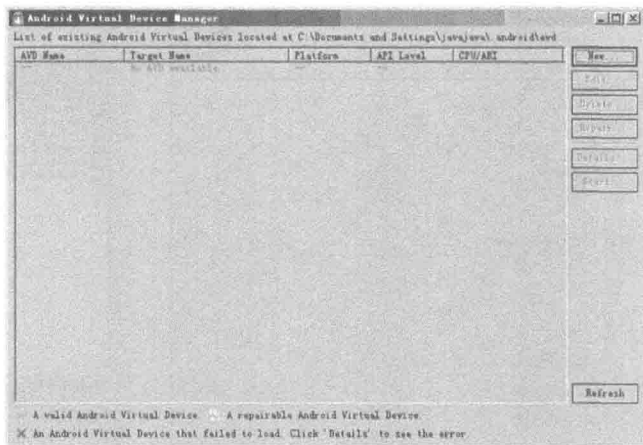


图 1-9 AVD 管理界面

单击图 1-9 中的【New】按钮, 弹出如图 1-10 所示的对话框。在图 1-10 中输入 AVD 的名字并选择需要的 Android 版本; 设置 SD Card 大小; 选择 Skin(屏幕)大小。然后单击【Create AVD】按钮, 则 AVD 创建成功并出现在列表中。

AVD 创建成功界面如图 1-11 所示。



图 1-10 创建 AVD 界面



图 1-11 AVD 创建成功界面

### 3. 运行程序

选择 HelloWorld 工程,单击右键,在弹出的菜单中选择 Run As > Android Application 运行工程,运行结果如图 1-11 所示。

## 1.5 拓展项目

题目:我的第一个手机程序。

要求:在 HelloWorld 工程的基础上进行修改,实现如图 1-12 所示的内容。



图 1-12 运行结果

## 1.6 课后练习

1. 一个 Activity 就是代表一个手机\_\_\_\_\_。
2. 开发 Android 应用程序主要使用的集成开发环境是\_\_\_\_\_。
3. AVD 表示\_\_\_\_\_。
4. res 目录主要存放\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_。
5. 每个应用程序都会对应一个\_\_\_\_\_清单文件,为 Android 系统提供了这个应用的基本信息。
6. \_\_\_\_\_元素是 AndroidManifest.xml 的根元素。
7. \_\_\_\_\_文件用来存放当前应用程序中使用到的字符串——以“键值对”的形式存放。
8. R.java 文件是\_\_\_\_\_,不允许开发者进行更改。
9. src 目录中存放项目的\_\_\_\_\_。
10. \_\_\_\_\_目录用于存放项目相关的资源文件,例如文本文件等。

## 项目2 设计用户登录界面

### 学习目标

知识目标:

1. 了解 Android 中布局类型;
2. 理解各种布局设计模式及布局相关属性;
3. 掌握布局中属性单位;
4. 理解布局文件和 Activity 类之间的关系。

技能目标:

1. 能够利用单一类型的布局进行界面设计;
2. 能够利用多种布局相互嵌套进行界面设计。

### 2.1 项目描述

本项目主要设计登录界面,实现组件正常显示和被按下相关组件时变换不同的图片,达到美化的效果。设计的登录界面如图 2-1 所示。

### 2.2 项目分析

#### 1. 布局设计分析

本项目整体采用 LinearLayout 嵌套布局,难点是如图 2-2 所示的登录输入界面的设计。图 2-2 的构成如图 2-3 所示,主要由三行构成:第一行由 EditText 可编辑文本框和 Button 命令按钮组件组成,用来模仿下拉列表框;第二行是一个 ImageView 图片框组件;第三行是一个 EditText 可编辑文本框组件。每一行由一个 LinearLayout 控制。这三行再由一个 LinearLayout 控制,设置其背景图片,就能达到图 2-2 所示的显示效果。



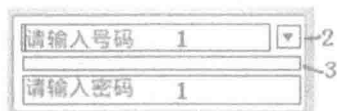
图 2-1 登录界面



图 2-2 登录输入界面

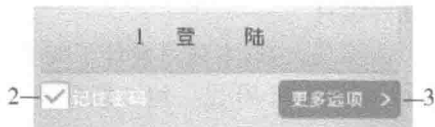


图 2-4 所示是登录辅助部分,由两行组成:第一行是 Button 命令按钮组件;第二行由 CheckBox 复选框和 Button 命令按钮组件构成。每一行由一个 LinearLayout 控制。



1. EditText 组件
2. Button 组件
3. ImageView 组件

图 2-3 登录输入界面分解



- 1和3: Button 组件
- 2: CkeechBox 组件

图 2-4 登录辅助部分分解

## 2. 组件图片效果切换设计分析

命令按钮组件被设计了正常显示的背景图和被按下时显示的背景图。当按钮被按下时,命令按钮的图片设置为被按下的图片;按钮没被按下时,命令按钮的图片设置为正常的背景图,从而实现图片切换。

对复选框组件设计了没有选中图片、没有选中被按下图片、选中图片、选中被按下四副图片,根据复选框的状态实现图片的切换。

对于图片的切换,在 Android 中通过使用选择器来实现。以实现【登录】按钮的效果为例来说明选择器的应用,其步骤如下:

第一步,在 res 文件夹下创建一个 drawable 文件夹;

第二步,在 drawable 文件夹下创建 login\_button\_select.xml 文件,以 selector 标记符作为根元素;

第三步,在 login\_button\_select.xml 文件中书写下列代码:

```
<?xml version="1.0" encoding="utf-8"? >
<selector xmlns:android="http://schemas.android.com/apk/res/android" >
    <! -- 命令按钮被按下时,显示被按下的图片 -- >
    <item android:state_pressed="true"
        android:drawable="@drawable/login_button_press"
    />
    <! -- 命令按钮没有获得焦点和没有被按下时,显示被按下的图片 -- >
    <item android:state_pressed="false"
        android:drawable="@drawable/login_button_nor"
    />
</selector >
```

第四步,在 main.xml 文件中,Button 命令标签的 background 属性关联选择器文件。

```
android:background="@drawable/login_button_select"
```

selector 标记符表示选择器,其 item 标记符的常用属性如下:

android:state\_focused 表示组件是否获得焦点。

android:state\_selected 表示组件是否被选中。

android:state\_pressed 表示组件是否被按下。