第 3 章

通信功能的设计及开发

本章的工作目标如下：

(1) 使用 Android 中布局和基本控件实现打电话、发送短信界面。

(2) 使用 Activity 和 Intent 实现拨打电话、发送短信功能。

(3) 完成程序运行与效果测试。

3.1 项 目 分 析

本项目的学习主要关注以下几点。

(1) 认识 Android 中视图的层次结构。

(2) 掌握 Android 中常用布局。

(3) 掌握 Android 中标签、文本框、按钮控件。

(4) 熟悉 Android 中 Intent(意图)的使用。

(5) 掌握 Android 中 Activity 的使用。

(6) 了解 Android 中 permission 的概念。

(7) 综合前面几点实现拨打电话和发送短信功能。

通信功能的设计及开发如图 3-1 所示。

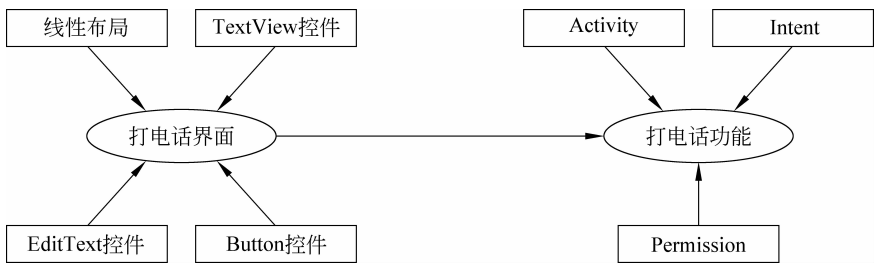

图 3-1 通信功能的设计及开发

1. 打电话界面编写

首先，在界面中加入垂直方向的线性布局，让控件按照从上到下的方式排列。其次，在界面中依次加入 TextView 标签控件显示“请输入手机号码”，加入 EditText 文本框控

件，让用户输入手机号码，加入 Button 按钮控件，显示“拨打”。

2. 打电话功能编写

首先，新建一个 Activity，调用 Activity 的 setContentView()方法与界面建立关联。其次，通过 Intent 意图启动拨打电话功能，最后，在 AndroidManifest. xml 文件中加入 Permission 权限。

3.2 项目界面设计

Android 用户界面的开发包括两个方面：用户界面设计和相应的事件处理。在一个 Android 应用程序中，用户界面由一系列的 View 和 ViewGroup 对象组合而成。Android 有 View 和 ViewGroup 对象，它们都继承自 View 基类；事件处理则包括 button 控件的单击事件等。

1. 用户界面的生成

Android 用户界面的生成有两种：一种是通过 XML 布局文件生成，这也是最简便和最直观的一种方法；另一种是用代码直接生成。对于 XML 生成的布局文件可以通过 ADT 提供的 UI 预览功能预览所创建的用户界面。

2. View

Android 中的 View 与以前理解的“视图”不同。在 Android 中，View 比视图具有更广的含义，它包含用户交互和显示，更像 Windows 操作系统中的 Window。View 对象是 Android 平台中用户界面体现的基础单位。View 类是其称为“Widget(工具)”的子类的基础，它们提供了诸如文本输入框和按钮之类的 UI 对象的完整实现。一个 View 对象是一个数据结构，存储布局参数和屏幕特定区矩形区域的内容。一个 View 会处理自己所在屏幕区域的测量、布局、绘制、焦点改变、滚动和按键手势交互。作为用户交互对象，一个 View 可以作为用户与系统的交互工具，接收事件。作为基类，View 类为 Widget 服务，Widget 是一组用于绘制交互屏幕元素的完全实现子类。Widget 处理自己的测距和绘图，所以可以快速地用它们去构建界面。常用的 Widget 包括 TextView、EditText 及 Button 等。

3. ViewGroup

ViewGroup 是 View 的子类，它具有 View 特性，但它主要用来充当 View 的容器，将其中的 View 视作自己的孩子，对它的子 View 进行管理，当然它的孩子也可以是 ViewGroup 类型。ViewGroup 和它的孩子们(View 和 ViewGroup)形成了一个树形结构，View 类有接收和处理消息的功能，Android 系统所产生的消息会在这些 ViewGroup 和 View 之间传递。

ViewGroup 可以为界面增加结构，并且将复杂的屏幕元素组成一个独立的实体。

ViewGroup 为 Layout 提供服务，Layout 用来提供各种布局结构，包括 linear 线性布局、表格布局和绝对布局等。

图 3-2 所示是一个由 View 和 ViewGroup 布局的活动(Activity)界面。一个 Activity 的界面可以包含多个 ViewGroup 和 View，通过两者的组合使用能够更好地完成更复杂界面的设计。

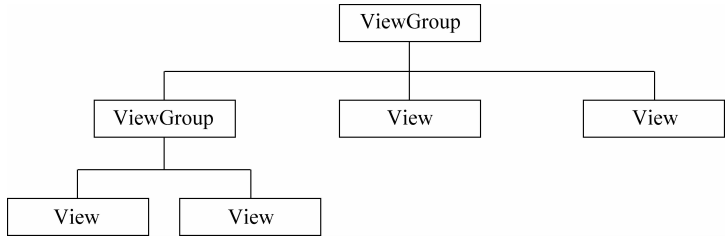

图 3-2 View 与 ViewGroup 组合使用布局的 Activity 界面

一个新的 Activity 被创建时是一个空白屏幕，可以把自己的用户界面放到上面。要设置用户界面，可以调用 setContentView()，并传入需要显示的 View 实例(通常是一个布局)。由于空白屏幕不是我们想要的，所以在创建一个新的 Activity 时，在 onCreate()处理程序中总是采用 setContentView()的方法设置我们需要显示的用户界面。具体使用 Activity，请参照本章相关实例。

3.2.1 知识准备

1. *布局*

Android 的界面是由布局和组件协同完成的，布局好比是建筑里的框架，而组件则相当于建筑里的砖瓦。组件按照布局的要求依次排列，就组成了用户所看见的界面。Android 的五大布局分别是 RelativeLayout(相对布局)、LinearLayout(线性布局)、FrameLayout(单帧布局)、TableLayout(表格布局)和 AbsoluteLayout(绝对布局)。前两种布局在项目实战中使用较多，后两种布局在项目中使用较少，本书所有项目重点使用的是前两种布局，在 QQ 项目中使用 FrameLayout 布局。

(1) RelativeLayout

RelativeLayout 按照各子元素之间的位置关系完成布局。在此布局中的子元素里，与位置相关的属性将生效。例如，android：layout_below、android:layout_above 等。子元素就通过这些属性和各自的 ID 配合指定位置关系。在指定位置关系时，引用的 ID 必须在引用之前先被定义，否则将出现异常。不能在 RelativeLayout 容器本身和它的子元素之间产生循环依赖，比如，不能将 RelativeLayout 的高设置成 WRAP_CONTENT 时将子元素的高设置成 ALIGN_PARENT_BOTTOM。

RelativeLayout 常用的位置属性如下。

- android:layout_above：将该控件置于给定 ID 的控件之上。
- android:layout_below：将该控件置于给定 ID 控件之下。

- android:layout_toLeftOf：将该控件置于给定 ID 的控件之左。
- android:layout_toRightOf：将该控件置于给定 ID 的控件之右。
- android:layout_alignBaseline：该控件基线对齐给定 ID 的基线。
- android:layout_alignBottom：该控件于给定 ID 的控件底部对齐。
- android:layout_alignLeft：该控件于给定 ID 的控件左对齐。
- android:layout_alignRight：该控件于给定 ID 的控件右对齐。
- android:layout_alignTop：该控件于给定 ID 的控件顶对齐。
- android:layout_alignParentLeft：如果为 True，该控件位于父控件的左部。
- android:layout_alignParentRight：如果为 True，该控件位于父控件的右部。
- android:layout_alignParentTop：如果为 True，该控件位于父控件的顶部。
- android:layout_alignParentBottom：如果为 True，该控件位于父控件的底部。
- android:layout_centerHorizontal：如果为 True，该控件将被置于水平方向的中央。
- android:layout_centerInParent：如果为 True，该控件将被置于父控件水平方向和垂直方向。
- android:layout_centerVertical：如果为 True，该控件将被置于垂直方向的中央。

RelativeLayout 是 Android 五大布局结构中最灵活的一种布局结构，比较适合一些复杂界面的布局，效果如图 3-3 所示。

图 3-3　RelativeLayout 布局

开发步骤。

第一步：新建 Android 工程。在 eclipse 左上角单击 File → New → Android Application Project，在弹出的对话框中填写应用名称（Application Name）、项目名称（Project Name）和包名（Package Name），如图 3-4 所示。

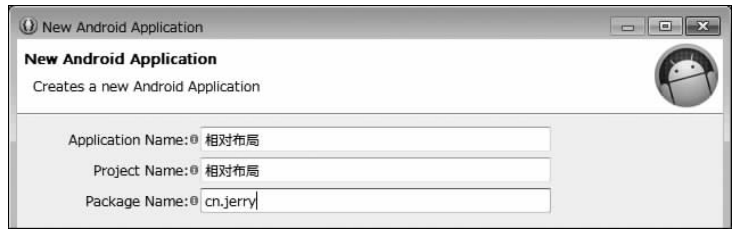

图 3-4 填写相应内容

一直单击 Next 按钮，直到单击 Finish 按钮结束，Android 工程创建成功。

第二步：添加布局代码。在项目中双击 res→layout 目录下的 Activity_main. xml，添加如下代码。

```
<TextView
    android:id="@+id/textView1"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="请输入用户名:" />
<EditText
    android:id="@+id/editText1"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/textView1">
    <requestFocus />
</EditText>
<Button
    android:id="@+id/button1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/editText1"
    android:layout_alignParentRight="true"
    android:text="OK"
     />
<Button
    android:id="@+id/button2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/editText1"
    android:layout_toLeftOf="@id/button1"
    android:layout_marginRight="20px"
    android:text="Cancle" />
```

代码解释如下。

① android:layout_below="@id/textView1"：这行代码指定文本框在 id 为 textView1 的 TextView 控件下方。

② android:layout_alignParentRight="true"：这行代码指定按钮位于手机屏幕的右边。

③ android:layout_toLeftOf="@id/button1"：这行代码指定 Button 按钮位于 id 为