第 5 课
Android 基础控件详解

Android 应用程序的人机交互界面有很多 Android 控件组成。几乎所有的 Android 都会涉及到控件技术，如文本框、编辑框、按钮、列表等控件。这些在 Android 应用程序中随处可见，本课将对 Android 提供的基础控件进行详细的介绍。

本课学习目标：

- ❑ 掌握文本框与编辑框的基本应用
- ❑ 掌握单选按钮和复选按钮的应用
- ❑ 掌握普通按钮和图片按钮的基本应用
- ❑ 掌握列表选择框、列表视图和图像视图的使用方法
- ❑ 掌握日期与时间选择器的基本使用
- ❑ 了解计时器控件的使用方法

5.1 文本框与编辑框

在应用程序中，经常需要编辑和显示文本，在 Android 中提供了文本框（TextView）控件和编辑框（EditText）控件。前者用于在屏幕上显示文本；后者用于在屏幕上显示可编辑的文本框。其中，EditText 是 TextView 的子类。下面将详细介绍这两种控件。

5.1.1 文本框

TextView 控件主要用于在屏幕上显示文本，通过在 XML 布局文件中配置<TextView>标记来使用文本框控件。其基本语法格式如下所示。

```
<TextView
     android:id="@+id/textView1"
     android:layout_width="wrap_content"
     android:layout_height="wrap_content"
     android:text="/hello world" />
```

在以上代码中，id 表示定义了 TextView 的变量名称为 textView1，会自动写进 R.java，在 R.java 文件中会生成内部类 id，可在主程序调用 R.id.textView1 来获取这个控件变量实体。layout_width 和 layout_height 表示 TextView 的高度和宽度，都设置为 wrap_content，表示将完整显示其内部的文本，布局元素将根据内容更改大小。text 属性表示要显示的文本的内容。

TextView 控件有很多属性，根据不同的属性显示 TextView 的效果也不同。常用的 XML 属性如表 5-1 所示。

表 5-1 TextView 支持的 XML 属性

属 性 名 称	描 述
android:autoLink	设置是否当文本为 URL 链接/邮箱/电话号码/map 时，文本显示为可单击的链接。可选值 none、web、email、phone、map、all
android:digits	设置允许输入哪些字符，如“1234567890.+-*/%\n()”
android:drawableBottom	用于在文本框内文本的底端绘制指定图像，该图像可以是放在 res\drawable 目录下的图片，通过“@drawable/文件名”设置
android:drawableLeft	用于在文本框内文本的左侧绘制指定图像，该图像可以是放在 res\drawable 目录下的图片，通过“@drawable/文件名”设置
android:drawablePadding	设置 text 与 drawable(图片)的间隔，与 drawableLeft、drawableRight、drawableTop、drawableBottom 一起使用，可设置为负数，单独使用没有效果
android:drawableRight	用于在文本框内文本的右侧绘制指定图像，该图像可以是放在 res\drawable 目录下的图片，通过“@drawable/文件名”设置
android:drawableTop	用于在文本框内文本的顶部绘制指定图像，该图像可以是放在 res\drawable 目录下的图片，通过“@drawable/文件名”设置
android:gravity	用于设置文本框内文本的对齐方式，可选值有 top、bottom、left、right、fill 等，这些属性也可以同时指定，各个属性之间用竖线隔开。例如，要指定控件靠右下角对齐，可以使用属性值 right|bottom
android:hint	Text 为空时显示的文字提示信息，可通过 textColorHint 设置提示信息的颜色
android:inputType	设置文本的类型，用于帮助输入法显示合适的键盘类型
android:linksClickable	设置链接是否单击链接

续表

属 性 名 称	描　　述
android:marqueeRepeatLimit	在 ellipsize 指定 marquee 的情况下，设置重复滚动的次数，当设置为 marquee_forever 时表示无限次
android:maxLength	限制显示的文本长度，超出部分不显示
android:lines	设置文本的行数，设置两行就显示两行，即使第二行没有数据
android:maxLines	设置文本的最大显示行数，与 width 或者 layout_width 结合使用，超出部分自动换行，超出行数将不显示
android:minLines	设置文本的最小行数，与 lines 类似
android:lineSpacingExtra	设置行间距
android:lineSpacingMultiplier	设置行间距的倍数
android:numeric	如果被设置，该 TextView 有一个数字输入法
android:password	以小点“.”显示文本
android:phoneNumber	设置为电话号码的输入方式
android:privateImeOptions	设置输入法选项
android:scrollHorizontally	设置文本超出 TextView 宽度的情况下，是否出现横拉条
android:selectAllOnFocus	如果文本是可选择的，让它获取焦点而不是将光标移动为文本的开始位置或者末尾位置
android:shadowColor	指定文本阴影的颜色，需要与 shadowRadius 一起使用
android:shadowDx	设置阴影横向坐标开始位置
android:shadowDy	设置阴影纵向坐标开始位置
android:shadowRadius	设置阴影的半径。设置为 0.1 就变成字体的颜色了，一般设置为 3.0 的效果比较好
android:singleLine	设置单行显示。如果和 layout_width 一起使用，当文本不能全部显示时，后面用“…”来表示。如果不设置 singleLine 或者设置为 false，文本将自动换行
android:text	设置显示文本
android:textAppearance	设置文字外观
android:textColor	设置文本颜色
android:textColorHighlight	被选中文字的底色，默认为蓝色
android:textColorHint	设置提示信息文字的颜色，默认为灰色。与 hint 一起使用
android:textColorLink	文字链接的颜色
android:textScaleX	设置文字之间间隔，默认为 1.0f
android:textSize	设置文字大小，推荐度量单位“sp”
android:textStyle	设置字形[bold(粗体)， italic(斜体)，bolditalic(又粗又斜)]，可以设置一个或多个，用“|”隔开
android:typeface	设置文本字体，必须是以下常量值之一：normal、sans、serif 或者 monospace
android:height	设置文本区域的高度，支持度量单位：dp/sp/in/mm(毫米)
android:maxHeight	设置文本区域的最大高度
android:minHeight	设置文本区域的最小高度
android:width	设置文本区域的宽度，支持度量单位：dp/sp/in/mm(毫米)
android:maxWidth	设置文本区域的最大宽度
android:minWidth	设置文本区域的最小宽度

【练习 1】

在 Eclipse 中创建一个 Android 项目，名称为 ch05_01，实现在文本框中显示超链接、邮箱，显示带图片的文本，单行和多行显示文本的功能。

（1）在项目中的 res/layout 目录下修改 activity_main.xml 文件，作为文本框显示的布局文件，

在其中添加 4 个文本框，分别设置属性如下所示。

```
<LinearLayout 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:orientation="vertical" >
<TextView
      android:id="@+id/textView01"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:autoLink="all"
      android:text="@string/text01"
      android:layout_margin="10dp"/>
<TextView
      android:id="@+id/textView02"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:drawableTop="@drawable/logo"
      android:text="@string/text02"
      android:gravity="center"
      android:layout_margin="10dp"/>
<TextView
      android:id="@+id/textView03"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:singleLine="true"
      android:text="@string/text03"/>
<TextView
      android:id="@+id/textView04"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:singleLine="false"
      android:text="@string/text03"
      android:lineSpacingMultiplier="1.5"/>
</LinearLayout>
```

其中第一个文本框添加了属性 autoLink，并且设置值为 all，表示该文本框中所含有的 URL 和邮箱地址，将用超链接显示。第二个文本框中添加了属性 drawableTop，是用来在文本框的上部显示图片，而且该文本在图片下方的正中间。第三个和第四个文本框中都设置了属性 singleLine，其中第三个值设置为 true，代表单行显示文本，多余的将用“…”代替。属性 lineSpacingMultiplier 表示行间距，layout_margin 表示边距。在上述代码中的 text 表示显示的文本，都来自项目的 res/values 中的 strings.xml 文件。

(2) 在项目的 res 目录下的 drawable_ldpi 文件夹中，放入一张名称为 logo.gif 的图片，用以在第二个文本框上部显示。

(3)在项目的 res/values 中的 strings.xml 文件中添加文本框中文本的显示内容,代码如下所示。

```
<string name="text01">汇智学习视频网站: http://www.itzcn.com 邮箱地址为:
admin@itzcn.com</string>
```

```
<string name="text02">带图片的文本框</string>
<string name="text03">Android 是一种基于 Linux 的自由及开放源代码的操作系统，主要使用于移动设备，如智能手机和平板电脑，由 Google 公司和开放手机联盟领导及开发。</string>
```

运行项目后，文本框显示的效果如图 5-1 所示。



图 5-1　文本框的显示效果

5.1.2　编辑框

编辑框主要用于在屏幕上显示文本输入框，便于用户输入信息，通过在 XML 布局文件中配置<EditView>标记来使用编辑框控件，其基本语法格式如下。

```
<EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/editView"
        android:hint="在编辑框内容为空时，显示该内容"
        android:singleLine="true"
        android:inputType="textPersonName"/>
```

在上述代码中，定义的编辑框的宽度和高度均为 wrap_content，表示将完整显示其内部的文本，布局元素将根据内容更改大小。hint 属性表示当编辑框内容为空的时候，显示的默认内容。singleLine 属性表示是否单行，设置为“true”表示输入内容为单行。inputType 属性是用于指定当前编辑框输入内容的文本类型。

提示

编辑框的 XML 属性与文本框属性一样，这里就不再一一列举了。

【练习 2】

下面给出一个使用编辑框的实例。在 Eclipse 中创建 Android 项目，名称为 ch05_02，实现会员注册界面。

（1）在项目中的 res/layout 目录下修改 activity_main.xml 文件，作为编辑框显示的布局文件，为了使应用程序布局显示不杂乱，这里添加了一个 TableLayout 表格布局管理器，并添加了 5 个 TableRow 表格行，修改后布局文件的主要代码如下所示。

```
<?xml version="1.0" encoding="utf-8"?>
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
```

```
    android:layout_height="match_parent"
    android:orientation="vertical" >
    <TableRow
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/tableRow1">
<!--在此处添加文本框与编辑框配置属性-->
    </TableRow>
```

由于表格行的代码基本一致，因此在此处省略其他四个表格行的代码，需要注意的是在编写其他几个表格行的时候应将各个表格行的 id 重新设置。

（2）在第一个表格中添加一个用于显示提示信息的文本框和一个用于输入会员昵称的编辑框控件，具体代码如下。

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/textView05"
    android:text="用  户  名"/>
<EditText
    android:layout_width="300dp"
    android:layout_height="wrap_content"
    android:id="@+id/nickName"
    android:hint="@string/edit01"
    android:singleLine="true"
    android:inputType="textPersonName"/>
```

上述代码中，EditView 控件中的 singleLine="true"表示单行输入文本。在 EditView 控件中，inputType="textPersonName"表示输入类型为用户名。

（3）在第二个表格中添加一个用于显示提示信息的文本框和一个用于输入密码的编辑框控件，具体代码如下。

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/textView06"
    android:text="密     码"/>
<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/pwd"
    android:hint="@string/edit02"
    android:password="true"
    android:singleLine="true"/>
```

上述代码中，EditView 控件的 password="true"表示输入的内容为密码，将用“.”代替输入的内容，不过这种方法已经过时，主要用以下方法设置密码。

（4）在第三个表格中添加一个用于显示提示信息的文本框和一个用于输入确认密码的编辑框控件，具体代码如下。

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/textView07"
    android:text="确认密码"/>
<EditText
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:id="@+id/pwd2"
    android:hint="@string/edit02"
    android:inputType="textPassword"
    android:singleLine="true"/>
```

上述代码中，EditView 控件中的 inputType="textPassword"表示输入的内容为密码，将用“.”代替输入的内容。

(5) 在第四个表格中添加一个用于显示提示信息的文本框和一个用于输入电话号码的编辑框控件，具体代码如下。

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/textView08"
    android:text="电话号码"/>
<EditText
    android:layout_width="300dp"
    android:layout_height="wrap_content"
    android:id="@+id/phone"
    android:hint="@string/edit03"
    android:singleLine="true"
    android:inputType="phone"/>
```

上述代码中的 EditView 控件，inputType="phone"表示输入的内容为电话号码，仅仅能输入整数。

(6) 在第五个表格中添加一个用于显示提示信息的文本框和一个用于输入邮箱地址的编辑框控件，具体代码如下。

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/textView09"
    android:text="邮箱地址"/>
<EditText
    android:layout_width="300dp"
    android:layout_height="wrap_content"
    android:id="@+id/email"
    android:hint="@string/edit04"
    android:singleLine="true"
    android:inputType="textEmailAddress"/>
```

上述代码中的 EditView 控件中，inputType="textEmailAddress"表示输入内容为电子邮件，将

会在虚拟键盘中加入字符“@”，用于电子邮件地址的输入。

（7）添加一个水平线性布局管理器，并在该布局管理器中添加两个按钮，具体代码如下。

```
<LinearLayout
    android:orientation="horizontal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/reg"
    android:text="@string/reg"/>
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/rest"
    android:text="@string/rest"/>
</LinearLayout>
```

上述代码中主要是设置了两个按钮控件，按钮控件将会在以后详细介绍，这里就不再说明。

（8）在项目的 res/values 中的 strings.xml 文件中添加编辑框中所用到的字符串内容，代码如下所示。

```
<string name="edit01">请输入用户名</string>
<string name="edit02">请输入密码</string>
<string name="edit03">请输入电话号码</string>
<string name="edit04">请输入邮箱地址</string>
<string name="reg">注册</string>
<string name="rest">重置</string>
```

（9）为了使注册信息能够在控制台下输出，在 com.android.activity 包中的 MainActivity.java 文件中，添加如下代码。

```
protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_editview);
        Button reg = (Button) findViewById(R.id.reg);  //获取注册按钮控件
        reg.setOnClickListener(new regOnClickListener());  //为注册按钮设置监听
    }
    class regOnClickListener implements OnClickListener{   //监听器实现
        public void onClick(View v) {
            EditText nickName = (EditText) findViewById(R.id.nickName);
                                                        //获取会员用户名控件
            EditText pwd = (EditText) findViewById(R.id.pwd);
                                                        //获取会员注册密码控件
            EditText pwd2 = (EditText) findViewById(R.id.pwd2);
                                                        //获取会员确认密码控件
            EditText phone = (EditText) findViewById(R.id.phone);
                                                    //获取会员注册电话号码控件
            EditText email = (EditText) findViewById(R.id.email);
```

```
                                                    //获取会员注册邮箱控件
            Log.i("用户名为: ", nickName.getText().toString());
            Log.i("密码为: ", pwd.getText().toString());
            Log.i("确认密码为: ", pwd2.getText().toString());
            Log.i("电话号码为: ", phone.getText().toString());
            Log.i("邮箱地址为: ", email.getText().toString());
        }
    }
```

上述代码中，findViewById()根据控件的 id 来获取控件对象，由于获取的控件对象为 View 类型，所以要根据需要转换成所想要得到的控件对象。Button 是按钮控件，获取后为按钮使用 setOnClickListener()添加一个监听，当单击该按钮的时候将会执行监听中的 OnClick(View v)方法，Button 控件将会在下个小节详细介绍，这里就不再赘述了。Log.i(String tag,String msg)用于将信息添加到日志文件中，然后在控制台的日志管理中显示。

运行项目后，编辑框的应用效果如图 5-2 所示。在编辑框中输入内容后的效果如图 5-3 所示。



图 5-2 编辑框的应用效果



图 5-3 编辑框输入内容后的效果

在控制台日志中显示内容如图 5-4 所示。

Level	Time	PID	TID	Application	Tag	Text
I	08-06 09:20:45.720	3099	3099	com.android.activity	用户名为:	admin
I	08-06 09:20:45.720	3099	3099	com.android.activity	密码为:	admin888
I	08-06 09:20:45.720	3099	3099	com.android.activity	确认密码为:	admin888
I	08-06 09:20:45.720	3099	3099	com.android.activity	电话号码为:	1593710000
I	08-06 09:20:45.720	3099	3099	com.android.activity	邮箱地址为:	admin@itzcn.com

图 5-4 在日志中显示编辑框中输入的内容

5.2 按钮

在 Android 中提供了普通按钮（Button）和图片按钮（ImageButton）两个控件。这两个控件都可以在应用程序界面显示一个按钮。当用户单击按钮时，将会触发一个 onClick 事件，可以通过为按钮添加单击事件监听器指定所要触发的动作。

5.2.1 普通按钮

Button 控件在布局中的配置在前面也有用到，Button 控件的基本使用方法与 TextView、EditView 并无太大的差别。其在 XML 布局文件中简单的配置如下所示。

```
<Button
    android:id="@+id/button"
```

```
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:text="普通按钮" />
```

在上面的代码中，id 表示定义了 Button 的变量名称为 button，会自动写进 R.java，在 R.java 文件中会生成内部类 id，可在主程序里面调用 R.id.button 来获取这个控件变量实体。layout_width 和 layout_height 表示 Button 的高度和宽度，都设置为 wrap_content，表示布局元素将根据内容更改大小。text 属性表示普通按钮上所显示的内容。

按钮最常用的就是单击事件，Android 提供了两种为按钮添加单击事件监听器的方法，一种是在 Java 代码中完成，具体代码如下所示。

```
Button button = (Button) findViewById(R.id.button);
                                                    //通过 Id 来获取布局文件中的按钮
button.setOnClickListener(new buttonOnClickListener());
                                                    //为按钮添加单击事件监听器
class buttonOnClickListener implements OnClickListener{//监听器实现接口
        @Override
        public void onClick(View v) {
        //编写需要执行的动作代码
        }
    }
```

另一种方法是在 Activity 中写一个包含 View 类型的参数的方法，其主要代码如下。

```
public void loginClick(View v) {
        //编写需要执行的动作代码
    }
```

那么就可以在布局文件中通过给按钮添加属性 android:onClick="loginClick"语句为按钮添加单击事件监听器。

5.2.2 图片按钮

图片按钮与普通按钮的使用方法基本相同，只不过图片按钮可以选择自定义的图片来作为按钮，在布局文件中加载图片按钮的基本代码如下所示。

```
<ImageButton
    android:id="@+id/imageButton"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:src="@drawable/imagebutton" />
```

上述代码中，图片按钮的宽和高设置为 match_parent，表示填满整个父窗口，src 属性表示图片的路径。

提示

和普通按钮一样，使用图片按钮的时候也需要为按钮添加单击事件监听器，具体方法与普通按钮方法一致，这里就不再赘述。

【练习 3】

在 Eclipse 中创建一个 Android 项目，名称为 ch05_03。在布局中添加一个普通按钮和图片按

钮，并为两个按钮设置单击事件监听器，单击按钮时触发事件。

（1）在项目 res 目录下的 drawable_ldpi 文件夹中，放入一张名称为 imagebutton.bmp 的图片，用于作为图片按钮的图片。

（2）在项目中的 res/layout 目录下修改 activity_main.xml 文件，改为线性布局。添加一个文本框控件，用于显示当前正在单击的按钮控件，主要代码如下。

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/textView01"
    android:text="@string/textShow"/>
```

当没有单击按钮的时候，该文本框显示的内容为项目下 res/values 的 strings.xml 文件中的属性为 textShow 的字符串，当单击按钮后内容将发生改变。

（3）添加一个水平线性布局管理器，并在该布局管理器中添加一个普通按钮和一个图片按钮，主要代码如下。

```
<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal" >
    <Button
        android:id="@+id/button"
        android:layout_width="match_parent"
        android:layout_height="fill_parent"
        android:text="@string/button" />
    <ImageButton
        android:id="@+id/imageButton"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:contentDescription="@string/button"
        android:src="@drawable/imagebutton" />
</LinearLayout>
```

上述代码中，Button 控件中的 text 属性表示该 Button 上所显示的内容；ImageButton 控件中的 src 表示该图片控件的图片来源。

（4）添加两个文本框控件，用于统计单击每个按钮的次数，主要代码如下。

```
<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:id="@+id/textView02"
android:text="@string/text01"/>
<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:id="@+id/textView03"
android:text="@string/text02/>
```

（5）在项目的 res/values 中的 strings.xml 文件中添加文本框文本的显示内容，代码如下所示。

```
<string name="button">登录</string>
<string name="textShow">您没有单击按钮</string>
<string name="text01">您没有单击普通按钮</string>
<string name="text02">您没有单击图片按钮</string>
```

（6）在 com.android.activity 包中的 MainActivity.java 文件中，添加按钮单击事件监听器和设置文本框显示内容，其主要代码如下。

```
private TextView textShow = null;
private int count1 = 0;                                    //统计普通按钮单击次数
private int count2 = 0;                                    //统计图片按钮单击次数
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    Button button = (Button) findViewById(R.id.button);//获取普通按钮对象
    ImageButton imageButton = (ImageButton) findViewById(R.id.imageButton);
                                                           //获取图片按钮
    button.setOnClickListener(new buttonOnClickListener());
                                                           //为普通按钮添加事件监听器
    imageButton.setOnClickListener(new imageButtonOnClickListener());
                                                           //为图片按钮添加事件监听器
}
class buttonOnClickListener implements OnClickListener{
    public void onClick(View v) {
        TextView buttonCount = (TextView) findViewById(R.id.textView02);
                                                           //获取文本框对象
        count1 ++ ;
        textShow = (TextView) findViewById(R.id.textView01);
        textShow.setText("您正在单击普通按钮");   //设置文本框显示内容
        buttonCount.setText("您一共单击了"  + count1 + "次普通按钮");
                                                           //设置文本框显示内容
    }
}
class imageButtonOnClickListener implements OnClickListener{
    public void onClick(View v) {
        TextView buttonCount = (TextView) findViewById(R.id.textView03);
        count2 ++ ;
        textShow = (TextView) findViewById(R.id.textView01);
        textShow.setText("您正在单击图片按钮");
        buttonCount.setText("您一共单击了"  + count2 + "次图片按钮");
    }
}
```

上述代码中 findViewById()根据控件的 id 来获取控件对象，由于获取的控件对象为 View 类型，所以要根据需要转换成所想要得到的控件对象。Button 是按钮控件，获得后为按钮使用 setOnClickListener()添加一个事件监听器，当单击该按钮的时候将会执行监听中的 OnClick(View v)方法。每执行一次 onClick(View v)方法，都会将单击次数加 1，这样就将按钮的单击次数统计出来，

并在文本框内显示。

运行项目后，按钮显示效果如图 5-5 所示。当单击按钮后，效果如图 5-6 所示。



图 5-5　按钮显示效果



图 5-6　单击按钮后效果

5.3 单选按钮与复选框

单选按钮和复选框都继承了普通按钮。因此它们都可以直接使用普通按钮支持各种属性和方法。同时，单选按钮控件和复选框控件提供了是否选中的功能。下面分别对单选按钮和复选框进行详细的介绍。

5.3.1 单选按钮

单选按钮用于多选一的状态，一般情况下，单选按钮默认为一个圆形的图片，在图标旁有该单选按钮的说明文字。在程序中，一般将同类单选按钮放在同一个单选按钮组中。在同一个单选按钮组中，当某个单选按钮被选中的时候，其他单选按钮则自动取消被选中的状态。

单选按钮用 RadioButton 表示，RadioButton 是 Button 的子类，所以支持 Button 所支持的属性。在屏幕上添加单选按钮可以通过在 XML 布局文件中使用<RadioButton>标记添加，其基本语法格式如下所示。

```
<RadioButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="男"
    android:checked="true"
    android:id="@+id/radio"/>
```

上述代码中，text 属性表示单选按钮的标签文字；checked 表示该单选按钮默认是否处于选中状态，当为 true 时表示该单选按钮处于选中状态；为 false 时，表示取消选中，默认为 false。

多数情况下，在使用单选按钮的时候都会使用到单选按钮组，将单选按钮放在单选按钮组中。单选按钮组在 XML 中通过添加<RadioGroup>标记来添加单选按钮组，基本语法格式如下所示。

```
<RadioGroup
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:id="@+id/radioGroup">
<!--添加单选按钮控件-->
</RadioGroup>
```

在上述代码中，orientation="horizontal"表示单选按钮组中的单选按钮水平放置。

与普通按钮的使用方法相同，在使用单选按钮的时候也要给单选按钮添加事件监听器，不过在这里是给单选按钮组添加监听，其主要代码如下所示。

```
    RadioGroup radioGroup = (RadioGroup) findViewById(R.id.radioGroup);
    radioGroup.setOnCheckedChangeListener(new RadioGroup.OnCheckedChange-
    Listener() {
        public void onCheckedChanged(RadioGroup group, int checkedId) {
            RadioButton radioButton = (RadioButton) findViewById(checkedId);
            radioButton.getText();              //获取被选中的单选按钮的值
        }
    });
```

上述代码中，在获取单选按钮的值时，首先通过单选按钮组的 id 来获取单选按钮组，然后为其添加 OnCheckedChangeListener 监听器，并在 onCheckedChanged()方法中根据参数 checkedId 获取被选中的单选按钮，并通过 getText()方法获取该单选按钮所对应的值。

5.3.2 复选框

复选框用于在多个选择中选择一个或多个的情况。在默认的情况下，复选框显示为一个方块图标，并在该图标旁有该复选框的说明文字。每一个复选框都有“选中”和“未选中”两种状态。

复选框在 Android 中用 CheckBox 表示，CheckBox 也是 Button 的子类，所以也支持 Button 所支持的属性。在屏幕上添加单选按钮可以通过在 XML 布局文件中使用<CheckBox>标记添加，其基本语法格式如下所示。

```
    <CheckBox
       android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       android:text="游泳"
       android:id="@+id/checkBox"/>
```

上述代码中，text 属性表示 CheckBox 的标签文字。id="@+id/checkBox"表示一个 id 为 checkBox 的 CheckBox 控件。

由于在使用复选框的时候可以同时选中多项，所以为了确定用户是否选中了该项，则需要给每一个复选框选项都添加一个事件监听器，为复选框添加监听器的主要代码如下所示。

```
    private CheckBox checkBox = null;
        protected void onCreate(Bundle savedInstanceState) {
            //省略部分代码
            checkBox = (CheckBox) findViewById(R.id.checkBox); //获取复选框对象
            checkBox.setOnCheckedChangeListener(new
CheckBoxOnCheckedChangedListener());
        }
        class CheckBoxOnCheckedChangedListener implements OnCheckedChange-
        Listener{
            @Override
            public void onCheckedChanged(CompoundButton arg0, boolean arg1) {
                if (checkBox.isChecked()) {         //判断该复选框是否被选中
                    checkBox.getText();             //获取选中复选框的值
```

```
            }
        }
    }
```

在上述代码中，获取复选框的值的时候，首先通过复选框的 id 来获取复选框对象，然后为其添加 OnCheckedChangeListener 监听器，并在 onCheckedChanged()方法中根据方法 isChecked()来获取该复选框是否被选中，如果被选中，则通过 getText()方法获取该复选框所对应的值。

【练习 4】

在 Eclipse 中创建一个 Android 项目，名称为 ch05_04，实现一个性别和兴趣的选择界面。

（1）在项目中的 res/layout 目录下修改 activity_main.xml 文件，首先添加一个线性布局，用于添加一个文本框控件和一个单选按钮组，其中在单选按钮组中添加两个单选按钮控件，主要代码如下所示。

```
<LinearLayout
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:orientation="vertical" >
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/sex"/>
<RadioGroup
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:id="@+id/sex">
    <RadioButton
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/man"
        android:id="@+id/man"/>
<!-- 省略性别为女的单选按钮 -->
</RadioGroup>
</LinearLayout>
```

在布局文件中的 orientation="vertical"表示在其中的控件将垂直放置；单选按钮组中的 orientation="horizontal"表示在单选按钮组中的两个单选按钮将水平放置；由于两个单选按钮的布局文件一致，这里就省略了性别为女的单选按钮的配置，其中性别为女的单选按钮的 text="@string/woman"，id="@+id/woman"。

（2）在上述步骤完成后，进行复选框和普通按钮的布局文件。添加一个文本框控件、五个复选框控件和一个普通按钮控件，主要代码如下所示。

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/fav"/>
<CheckBox
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
```

```
        android:text="@string/like1"
        android:id="@+id/like1"/>
    <!-- 省略其他四个兴趣复选框 -->
    <Button
        android:layout_height="wrap_content"
        android:layout_width="wrap_content"
        android:text="@string/button"
        android:id="@+id/button"/>
```

在上述代码中，复选框控件中的 text="@string/like1"表示该复选框显示的内容，由于这 5 个选框控件配置相同，这里就省去其他四个复选框的配置，其中其他四个复选框的内容分别为 text="@string/like2 、 text="@string/like3 、 text="@string/like4 、 text="@string/like5 ， id 分别为 id="@+id/like2"、id="@+id/like3"、id="@+id/like4"、id="@+id/like5"。

（3）在项目 res/values 中的 strings.xml 文件中添加文本框文本的显示内容，代码如下所示。

```
<string name="sex">请选择您的性别</string>
<string name="fav">请选择您的兴趣</string>
<string name="man">男</string>
<string name="woman">女</string>
<string name="button">确定</string>
<string name="like1">体育</string>
<string name="like2">美术</string>
<string name="like3">看电影</string>
<string name="like4">上网</string>
<string name="like5">购物</string>
```

（4）在 com.android.activity 包中的 MainActivity.java 文件中，添加了单选按钮组、复选框控件和普通按钮控件的监听器，其主要代码如下所示。

```
private RadioGroup sex = null;                          //声明单选按钮组对象
private CheckBox like1 = null;                          //声明复选框控件对象
//省略其他四个复选框对象的声明
private RadioButton radio = null;                       //声明单选按钮对象
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    Button button = (Button) findViewById(R.id.button);
    button.setOnClickListener(new buttonOnClickListener());
                                                        //为普通按钮添加监听器
    sex = (RadioGroup) findViewById(R.id.sex);          //根据 id 属性获取单选按钮组
    like1 = (CheckBox) findViewById(R.id.like1);        //根据 id 属性获取复选框
    like1.setOnCheckedChangeListener(new checkBoxListener());
                                                        //为复选框设置监听
//省略其他四个复选框的获取和监听器的添加
    sex.setOnCheckedChangeListener(new RadioGroup.OnCheckedChangeListener() {
        public void onCheckedChanged(RadioGroup group, int checkedId) {
            radio = (RadioButton) findViewById(checkedId);
        }
    });
}
```

```
class checkBoxListener implements OnCheckedChangeListener{
    public void onCheckedChanged(CompoundButton buttonView,
            boolean isChecked) {
    }
}
class buttonOnClickListener implements OnClickListener{
public void onClick(View v) {
    if (radio != null) {                          //单选按钮对象是否为空
        Log.i("您选择的性别是: ", radio.getText().toString());
    }else{
        Log.i("您选择的性别是: ", "无");
    }
    String like = "";
    if (like1.isChecked()) {                      //复选框是否被选中
        like = like + like1.getText().toString() + " ";
    }
//省略其他四个复选框的判断
        if ("".equals(like.trim())) {             //没有复选框被选中
            Log.i("您的兴趣是: ", "无");
        } else {
            Log.i("您的兴趣是: ", like);
        }
    }
}
```

上述代码中，在获取单选按钮的值时，首先通过单选按钮组的 id 来获取单选按钮组，然后为其添加 OnCheckedChangeListener 监听器，并在 onCheckedChanged()方法中根据参数 checkedId 获取被选中的单选按钮，并将声明的 RadioButton 对象实例化。如果 radio 为空，表示未选中该单选按钮组中的任何一个单选按钮；若不为空，则通过 radio.getText().toString()方法将单选按钮的内容输出，并用 Log.i(String tag,String msg)添加到日志文件中。

在获取复选框值的时候，首先通过复选框的 id 来获取复选框对象，然后为其添加 OnCheckedChangeListener 监听器，并在 onCheckedChanged()方法中根据方法 isChecked()来获取该复选框是否被选中。如果被选中，则通过 like = like + like1.getText().toString() + " "将获取的复选框所对应的值拼接到字符串 like 中。最后再根据"".equals(like.trim())来判断该 like 字符串是否为空，为空则表示未选中任何一个复选框。

运行该项目后，未选择性别和兴趣时效果如图 5-7 所示。选择性别和兴趣后效果如图 5-8 所示。

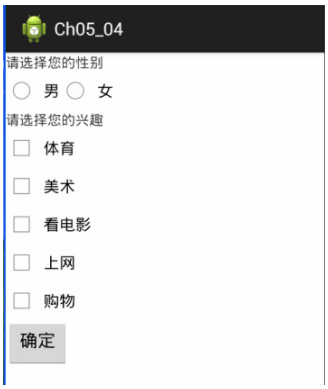
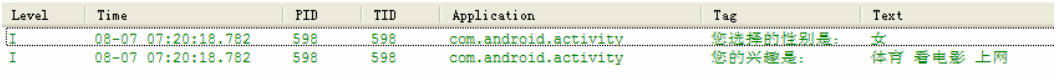


图 5-7　未选择性别和兴趣



图 5-8　选择性别和兴趣

单击【确定】按钮后，在日志控制台中输出选中的性别和兴趣如图 5-9 所示。



Level	Time	PID	TID	Application	Tag	Text
I	08-07 07:20:18.782	598	598	com.android.activity	您选择的性别是:	女
I	08-07 07:20:18.782	598	598	com.android.activity	您的兴趣是:	体育 看电影 上网

图 5-9　在日志控制台中输出选中的性别和兴趣

5.4 列表选择框

列表选择框（Spinner）控件与网页中的下拉列表框类似，通常是提供一些可选择的列表项以提供用户选择。

在使用列表选择框的时候，需要在 XML 布局文件中加入<Spinner>标记来实现，其基本语法格式如下所示。

```
<Spinner
    android:entries="@array/type"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/spinner"/>
```

在上述代码中，entries 为可选属性，用于指定列表项。通常是在项目下的 res\values 中新建一个 XML 文件，用于添加列表项的数组资源文件，其代码格式如下所示。

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string-array name="type">
        <item>水果</item>
        <item>蔬菜</item>
        <item>衣服</item>
        <item>电器</item>
        <item>其他</item>
    </string-array>
</resources>
```

其中 string-array name="type"表示该数组资源名称，也是在 Spinner 中 entries 属性所引用的内容。如果 entries 属性不指定时，也可以在 Java 代码中通过为其指定适配器的方式来指定数组资源。

提示

通常情况下，如果列表选择框中要显示的列表项是已知的，那么可以将其内容保存在数组资源中，然后通过数组资源来为列表选择框指定列表项。

【练习 5】

在 Eclipse 中创建一个 Android 项目，名称为 ch05_05，实现列表选择框加载已知的数组资源。

（1）在项目中的 res/layout 目录下修改 activity_main.xml 文件，并添加一个列表选择框控件，其代码如下所示。

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

```
    android:layout_height="match_parent"
    android:orientation="vertical">
    <Spinner
        android:entries="@array/type"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/spinner"/>
</LinearLayout>
```

在上述代码中，为 Spinner 添加了一个名称为 type 的字符串数组。

（2）在项目中的 res\values 目录中新建数组资源文件 arrays.xml，在该文件中添加一个字符串数组，名称为 type，具体代码如下。

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string-array name="type">
        <item>水果</item>
        <item>蔬菜</item>
        <item>衣服</item>
        <item>电器</item>
        <item>其他</item>
    </string-array>
</resources>
```

（3）在 com.android.activity 包中的 MainActivity.java 文件中，为 Spinner 控件添加选择列表项事件监听器，以此来获取列表选择框所选定的内容，其主要代码如下所示。

```
protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Spinner spinner = (Spinner) findViewById(R.id.spinner);
                                                    //获取 Spinner 对象
        spinner.setOnItemSelectedListener(new spinnerOnItemSelected-
        Listener());//添加监听器
    }
    class spinnerOnItemSelectedListener implements OnItemSelectedListener{
        public void onItemSelected(AdapterView<?> arg0, View arg1, int arg2,long
        arg3)
        {
            Log.i("您选择了: ", arg0.getItemAtPosition(arg2).toString());
                                                    //添加到日志
            //Log.i("您选择了: ", arg0.getSelectedItem().toString());
        }
        public void onNothingSelected(AdapterView<?> arg0) {
        }
    }
```

在上述代码中，为选择列表框添加了 OnItemSelectedListener 事件监听器，当选择选择列表框中的某一项时会触发监听，通过使用 getSelectedItem()方法获取到选择项的值，并输出在 Log 控制台。

运行该项目，Spinner 控件的运行效果如图 5-10 所示。选择列表选择框的效果如图 5-11 所示。

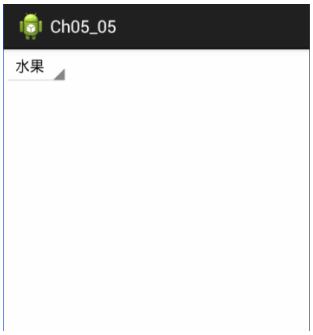


图 5-10　Spinner 控件的运行效果



图 5-11　选择列表框中选项的效果

选中选择列表框时，在控制台输出的日志信息如图 5-12 所示。



图 5-12　控制台输出选择列表框选中的内容

上述练习的前提是选择列表框中所要显示的列表项是已知的，下面将讲述如果不在 XML 布局文件中配置 Spinner 控件的 entries 属性，通过在 Java 代码中定义数组来实现。

【练习 6】

在 Eclipse 中创建一个 Android 项目，名称为 ch05_06，通过为选择列表框加载适配器来实现选择列表框的显示。

（1）在项目中的 res/layout 目录下修改 activity_main.xml 文件，并添加一个列表选择框控件，其代码如下所示。

```
<Spinner
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/spinner"/>
```

由于要通过指定适配器来实现选择列表框，这里就不再指定数组资源。

（2）在 com.android.activity 包中的 MainActivity.java 文件中，为 Spinner 控件添加选择列表项事件监听器，并为其指定适配器，以此来获取列表选择框所选定的内容，其主要代码如下。

```
//省略部分代码
Spinner spinner = (Spinner) findViewById(R.id.spinner);
String type[] = new String[]{"水果","蔬菜","衣服","电器","其他"};
ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,android.R.layout.
simple_spinner_item, type);
adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_it
em);//为适配器设置列表框下拉的选项样式
spinner.setAdapter(adapter);                    //适配器与选择列表框关联
spinner.setOnItemSelectedListener(new spinnerOnItemSelectedListener());
//省略部分代码
```

上述代码中的选择列表项事件监听器与练习 5 中的一样。在 Java 文件中使用字符串数组创建适配器，首先要创建一个一维数组，用于保存下拉列表框显示的内容，然后使用 ArrayAdapter 类的构造方法 ArrayAdapter(Context context,int textViewResourceId,T[] objects) 实例化一个 ArrayAdapter 类的实例。之后通过 setDropViewResource()方法为适配器设置下拉列表框的选项样式，然后使用 setAdapter()将适配器与选择列表框关联。

运行该项目，Spinner 控件的运行效果如图 5-13 所示。选择列表选择框的效果如图 5-14 所示。



图 5-13　Spinner 控件的运行效果



图 5-14　选择列表框中选项的效果

选中选择列表框时，在控制台输出的日志信息如图 5-15 所示。



图 5-15　控制台输出选择列表框选中的内容

5.5 列表视图

列表视图（ListView）在 Android 中以垂直列表的形式列出需要显示的列表项。在 Android 中可以使用两种方法向屏幕中添加列表视图。一种是直接使用 ListView 控件在 XML 布局文件中配置；另一种是让 Activity 继承 ListActivity 来实现。下面将详细介绍这两种实现方法。

5.5.1　使用 ListView 控件创建列表视图

在 XML 布局文件中添加<ListView>标记来实现使用 ListView 创建列表视图，这与其他控件的配置基本相同，其基本语法格式如下。

```
<ListView
    android:entries="@array/type"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/listView"/>
```

与列表选择框的语法格式一样，在上述代码中，entries 为可选属性，用于指定列表项，也可以不设置该属性，在 Java 代码中通过为其指定适配器的方式来指定数组资源。

ListView 支持的 XML 属性如表 5-2 所示。

表 5-2　ListView 支持的 XML 属性

属 性 名 称	描　　述
android:divider	用于为列表视图设置分隔条，既可以用颜色分割，也可以用 Drawable 资源分离
android:dividerHeight	用于设置分隔条的高度
android:entries	用于通过数组资源为 ListView 指定列表项
android:footerDividersEnabled	用于设置是否在 footer View 之前绘制分隔条，默认为 true，设置为 false 时，表示不绘制。使用该属性时，需要通过 ListView 控件提供的 addFooterView()方法为 ListView 设置 footer View
android:headerDividersEnabled	用于设置是否在 footer View 之后绘制分隔条，默认为 true，设置为 false 时，表示不绘制。使用该属性时，需要通过 ListView 控件提供的 addHeaderView()方法为 ListView 设置 header View

ListView 指定的外观形式通常有以下几个。

- **simple_list_item_1**　每个列表项都是一个普通的文本。
- **simple_list_item_2**　每个列表项都是一个普通的文本（字体略大）。
- **simple_list_item_checked**　每个列表项都有一个选中的列表项。
- **simple_list_item_multiple_choice**　每个列表项都是带复选框的文本。
- **simple_list_item_single_choice**　每个列表项都是带单选按钮的文本。

在使用列表视图时，与列表选择框一样，如果没有在布局文件中为 ListView 指定要显示的内容，也可以通过为其设置 Adapter 来指定所需要显示的列表项。

【练习 7】

在 Eclipse 中创建一个 Android 项目，名称为 ch05_07，使用 ListView 控件来创建列表视图。

（1）在项目中的 res/layout 目录下修改 activity_main.xml 文件，并添加一个列表视图控件，其代码如下所示。

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
    <ListView
        android:entries="@array/type"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/listView"/>
</LinearLayout>
```

在上述代码中，为 ListView 添加了一个名称为 type 的字符串数组。

（2）在项目中的 res\values 目录中新建数组资源文件 arrays.xml，在该文件中添加一个字符串数组，名称为 type，这里与练习 5 中（2）一致，这里就省略了。

（3）在 com.android.activity 包中的 MainActivity.java 文件中，为 ListView 控件添加事件监听器，以此来获取列表视图所选定的内容，其主要代码如下。

```
ListView listView = (ListView) findViewById(R.id.listView);
listView.setOnItemClickListener(new ListView.OnItemClickListener() {
    public void onItemClick(AdapterView<?>arg0,View arg1,int arg2,long arg3) {
```

```
            Log.i("您选择了: ", arg0.getItemAtPosition(arg2).toString());
        }
    });
```

在上述代码中，为列表视图添加了 setOnItemClickListener 事件监听器，当选择选择列表框中的某一项时会触发监听，通过使用 getItemAtPosition(int position)方法获取到选择项的值，并输出在 Log 控制台。

运行该项目，列表视图运行效果如图 5-16 所示。当选中某一列表，如选中衣服时，控制台输出信息如图 5-17 所示。



图 5-16　列表视图运行效果

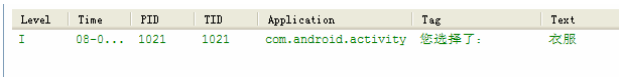


图 5-17　控制台输出列表视图选中的内容

注意

在列表视图布局文件中如果不设置数组资源，不配置 ListView 控件的 entries 属性，那么也可以通过在 Java 代码中定义数组来实现。与列表选择框一样，也是通过设置适配器来实现。

5.5.2　Activity 继承 ListActivity 实现列表视图

如果程序的窗口仅仅需要显示一个列表，那么就可以使用 Activity 继承 ListActivity 来实现。继承了 ListActivity 的类不需要调用 setContentView()方法来显示界面，而是可以直接为其设置适配器，从而显示一个列表。

【练习 8】

在 Eclipse 中创建一个 Android 项目，名称为 ch05_08，通过让 Activity 继承 ListActivity 来实现列表视图。

在 com.android.activity 包中的 MainActivity.java 文件中修改 MainActivity 类，使其继承 ListActivity，然后为列表视图添加适配器，并使用 setListAdapter()方法将其添加到列表中。为了单击 ListView 各个列表时获取到选择项的值，需要重写父类的 OnListItemClick()方法，MainActivity 的主要方法如下。

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    String type[] = new String[]{"水果","蔬菜","衣服","电器","其他"};
    ArrayAdapter<String> adapter = new ArrayAdapter<String>(this, android.R.
    layout.simple_spinner_item,type);
        setListAdapter(adapter);
    }
    protected void onListItemClick(ListView l, View v, int position, long id) {
        super.onListItemClick(l, v, position, id);
        Log.i("您选择了: ", l.getItemAtPosition(position).toString());
    }
```

在上述代码中，为列表视图添加了 setOnItemClickListener 事件监听器，当选择选择列表框中的某一项时会触发监听，通过使用 getItemAtPosition(int position)方法获取到选择项的值，并输出在 Log 控制台。

运行该项目，列表视图运行效果如图 5-18 所示。当选中某一列表，例如选中水果时，控制台输出信息如图 5-19 所示。



图 5-18　列表视图运行效果

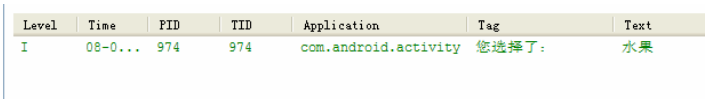


图 5-19　控制台输出列表视图选中的内容

5.6 图像视图

图像视图（ImageView）用于在屏幕中显示 Drawable 对象，通常用来显示图片。在使用 ImageView 控件为程序加载图片时，是通过在 XML 布局文件中添加<ImageView>标记来实现。

在使用 ImageView 控件显示图像时，通常先将要显示的图片放置在 res/drawable 目录下，然后将其显示在布局管理器中，其基本语法如下。

```
<ImageView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:src="@drawable/logo"
    android:id="@+id/imageView01"/>
```

在上述代码中，android:src="@drawable/logo"表示 ImageView 所显示的 Drawable 对象的 ID 为 logo。

ImageView 常用的 XML 属性如表 5-3 所示。

表 5-3　ImageView 支持的 XML 属性

属 性 名 称	描　　述
android:adjustViewBounds	用于设置 ImageView 是否调整自己的边界来保持所需要显示图片的长度变化
android:baseline	在视图中的偏移
android:baselineAlignBottom	如果值为 true，图像的视图将基线对齐基于其底部边缘
android:cropToPadding	如果设置为 true，图像将被裁剪以适合其填充
android:maxHeight	设置 ImageView 的最大高度（android:adjustViewBounds 属性值为 true 时起作用）
android:maxWidth	设置 ImageView 的最大宽度（android:adjustViewBounds 属性值为 true 时起作用）
android:scaleType	用于设置所显示的图片如何缩放或移动以适应 ImageView 的大小
android:src	用于设置 ImageView 所显示的 Drawable 对象的 ID
android:tint	设置图像着色的颜色

【练习 9】

在 Eclipse 中创建一个 Android 项目，名称为 ch05_09，通过使用 ImageView 控件来显示图片。

（1）在项目 res 目录下的 drawable_ldpi 文件夹中，放入两张名称分别为 logo.gif 和 background.gif 的图片，用于作为显示的图片和背景图片。

（2）在项目中的 res/layout 目录下修改 activity_main.xml 文件，改为线性布局。在其中添加一个 ImageView 控件，其代码如下。

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="@drawable/background">
    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/logo"
        android:id="@+id/imageView01"
        android:layout_margin="5dp"/>
</LinearLayout>
```

上述代码中，android:background 属性为该程序指定了背景图片，android:src 属性指定了所要显示图片，android:layout_margin 表示该控件的边距。

（3）在布局管理器中添加一个 ImageView 控件，并设置该控件的最大高度和宽度，具体代码如下。

```
<ImageView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:src="@drawable/logo"
    android:id="@+id/imageView02"
    android:layout_margin="5dp"
    android:adjustViewBounds="true"
    android:maxWidth="75dp"
    android:maxHeight="50dp"/>
```

在上述代码中，android:adjustViewBounds 用于设置 ImageView 是否调整自己的边界来保持所需显示图片的长度变化，为 true 时表示调整自己的边界来保持所需显示图片的长度变化；android:maxWidth 和 android:maxHeight 分别表示 ImageView 的最大宽度和最大高度。

（4）在布局管理器中添加一个 ImageView 控件，实现保持纵横比缩放图片，直到该图片能完全显示在 ImageView 控件中，并让该图片显示在 ImageView 控件的右下角，具体代码如下。

```
<ImageView
    android:layout_width="100dp"
    android:layout_height="100dp"
    android:src="@drawable/logo"
    android:id="@+id/imageView03"
    android:scaleType="fitEnd"
    android:layout_margin="5dp"
    android:background="#666666"/>
```

上述代码中，android:scaleType 用于设置所显示的图片如何缩放或移动以适应 ImageView 的大小，fitEnd 表示保持纵横比缩放图片，直到该图片能完全显示在 ImageView 中，缩放完成后，该图片放在 ImageView 的右下角。

（5）在布局管理器中添加一个 ImageView 控件，实现为显示在 ImageView 控件中的图像着色的功能，具体代码如下。

```
<ImageView
    android:layout_width="100dp"
    android:layout_height="100dp"
    android:src="@drawable/logo"
    android:id="@+id/imageView04"
    android:tint="#77889900"
    android:layout_margin="5dp"
    android:background="#666666"/>
```

在上述代码中，android:tint 用于给图片着色。

运行该项目，使用 ImageView 显示图像的效果如图 5-20 所示。



图 5-20　应用 ImageView 显示图像

5.7 日期与时间选择器

为了让用户选择设置日期和时间，Android 提供了日期/时间选择器，分别是 DatePicker 和 TimePicker 控件。这两个控件的使用比较简单，可以在 Eclipse 的可视化界面设计器中选择对应的控件放在布局文件中。下面将详细介绍这两种控件。

5.7.1　日期选择器

在使用 DatePicker 控件显示日期时，通常是使用<DatePicker>标记在 XML 布局文件中配置，其基本语法如下。

```
<DatePicker
    android:id="@+id/datePicker"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"/>
```

为了可以在程序中获取到用户选择的日期，需要为 DatePicker 控件添加事件监听器。其事件监听器是 OnDateChangedListener，监听器配置的主要代码如下。

```
DatePicker date = (DatePicker) findViewById(R.id.datePicker);
date.init(year, month, day, new dateOnDateChangetdListener());.
//省略部分代码
class dateOnDateChangetdListener implements OnDateChangedListener{
    public void onDateChanged(DatePicker view, int year, int monthOfYear, int
    dayOfMonth) {
        MainActivity.this.year = year;
        MainActivity.this.month = monthOfYear;
        MainActivity.this.day = dayOfMonth;
    }
}
```

5.7.2 时间选择器

在使用 TimePicker 控件显示时间时，通常是使用<TimePicker>标记在 XML 布局文件中配置，其基本语法如下。

```
<TimePicker
   android:id="@+id/timePicker"
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"/>
```

与日期选择器一样，使用时间选择器时也要添加事件监听器，TimePicker 控件的事件监听器是 OnTimeChangedListener。实现的主要代码如下。

```
TimePicker time = (TimePicker) findViewById(R.id.timePicker);
time.setOnTimeChangedListener(new timeOnTimeChangedListener());
//省略部分代码
class  timeOnTimeChangedListener implements OnTimeChangedListener{
    public void onTimeChanged(TimePicker view, int hourOfDay, int minute) {
        MainActivity.this.hour = hourOfDay;
        MainActivity.this.minute = minute;
    }
}
```

【练习 10】

在 Eclipse 中创建一个 Android 项目，名称为 ch05_10，在屏幕中添加日期和时间选择器，在改变日期和时间时，能够得到改变后的日期和时间。

（1）在项目中的 res/layout 目录下修改 activity_main.xml 文件，然后添加一个 TextView 控件、一个 DatePicker 控件和 TimePicker 控件，其代码如下所示。

```
<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/text"
```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"/>
    <DatePicker
        android:id="@+id/datePicker"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"/>
    <TimePicker
        android:id="@+id/timePicker"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"/>
</LinearLayout>
```

上述代码中，TextView 是用来显示日期和时间。

（2）在 com.android.activity 包中的 MainActivity.java 文件中，为 DatePicker 和 TimePicker 控件添加事件监听器，来获取到用户选择的日期和时间，其主要代码如下。

```
private int year = -1;
private int month = -1;
private int day = -1;
private int hour = -1;
private int minute = -1;
private TextView text = null;
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
DatePicker date = (DatePicker) findViewById(R.id.datePicker);
                                                    //获取日期选择器控件
TimePicker time = (TimePicker) findViewById(R.id.timePicker);
                                                    //获取时间选择器控件
text = (TextView) findViewById(R.id.text);
time.setIs24HourView(true);                         //设置时间为 24 小时制式
Calendar calendar = Calendar.getInstance();         //创建日历对象
year = calendar.get(Calendar.YEAR);                 //获取当前年份
month = calendar.get(Calendar.MONTH);               //获取当前月份
day = calendar.get(Calendar.DAY_OF_MONTH);          //获取当前日
hour = calendar.get(Calendar.HOUR_OF_DAY);          //获取当前小时
minute = calendar.get(Calendar.MINUTE);             //获取当前分钟
date.init(year, month, day, new dateOnDateChangetdListener());
                                                    //为日期添加监听器
time.setOnTimeChangedListener(new timeOnTimeChangedListener());
                                                    //为时间添加监听器
text.setText("现在是: " + year+"年"+(month+1)+"月"+day+"日"+hour+"时"+minute +"
分");//设置文本框内容
}
class dateOnDateChangetdListener implements OnDateChangedListener{
    public void onDateChanged(DatePicker view, int year, int monthOfYear, int
    dayOfMonth) {
        MainActivity.this.year = year;              //改变 year 的值
        MainActivity.this.month = monthOfYear;      //改变 month 的值
```

```
            MainActivity.this.day = dayOfMonth;          //改变 day 的值
            show(year, month, day, hour, minute);        //在文本框内显示日期和时间
        }
    }
    class  timeOnTimeChangedListener implements OnTimeChangedListener{
    public void onTimeChanged(TimePicker view, int hourOfDay, int minute) {
        MainActivity.this.hour = hourOfDay;              //改变 hour 的值
        MainActivity.this.minute = minute;               //改变 minute 的值
        show(year, month, day, hour, minute);            //在文本框内显示日期和时间
    }
    }
    private void show(int year,int month,int day,int hour,int minute){
                                                         //用于获取选择的日期和时间
        text.setText("您选择的日期是: " + year+"年"+(month+1)+"月"+day+"日"+hour+"时
"+minute +"分");
    }
```

注意

由于通过 DatePicker 对象获取到的月份是 0~11 月，所以需要将获取到的月份加 1，才能代表真正的月份。

运行该项目后，屏幕上显示的效果如图 5-21 所示，当改变日期时，屏幕上显示的效果如图 5-22 所示，当改变时间时，屏幕上显示的效果如图 5-23 所示。



图 5-21　项目运行效果图



图 5-22　改变日期时效果图



图 5-23　改变时间时效果图

5.8 计时器

计时器（Chronometer）控件可以显示从某个起始时间开始，一共过去了多长时间。由于该控件继承 TextView，所以它以文本的形式显示内容。一般在使用该控件的时候，会调用以下几个方法。

- **setBase(long base)**　设置计时器计时的基准（开始）时间。
- **setFormat(String format)**　设置用于格式化显示格式的字符串，计时器将用“MM:SS”或“H:MM:SS”形式的值替换格式化字符串中的第一个“%s”。

- **setOnChronometerTickListener(OnChronometerTickListener listener)** 设置计时器变化时调用的监听器。
- **start()** 开始计时，该操作不会影响到由 setBase(long)设置的基准（开始）时间，仅影响显示的视图。
- **stop()** 停止计时，不会影响用 setBase(long)方法设置的基准（开始）时间，只影响视图的显示。

在使用计时器控件时，需要在 XML 布局文件中添加<Chronometer>标记，其主要代码如下。

```
<Chronometer
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/chronometer"
  />
```

【练习 11】

在 Eclipse 中创建一个 Android 项目，名称为 ch05_11，在屏幕中添加一个计时器。

（1）在项目中的 res/layout 目录下修改 activity_main.xml 文件，然后添加一个 Chronometer 控件和一个普通按钮控件，其代码如下所示。

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <Chronometer
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/chronometer"/>
    <Button
           android:layout_width="wrap_content"
           android:layout_height="wrap_content"
           android:id="@+id/rest"
           android:text="@string/rest"
           android:visibility="gone"/>
</LinearLayout>
```

上述代码中，Button 控件的 android:visibility 属性用来设置该控件的可见与不可见的状态，设置为“gone”表示不可见并且不占用空间。

（2）在 com.android.activity 包中的 MainActivity.java 文件中，获取计时器控件，并设置起始时间和显示时间的格式，开启计时器等，其代码如下。

```
private Chronometer chronometer = null;
private Button rest = null;
protected void onCreate(Bundle savedInstanceState) {
//省略部分代码
    rest = (Button) findViewById(R.id.rest);            //获取按钮控件
    chronometer = (Chronometer) findViewById(R.id.chronometer);
                                                        //获取计时器控件
    chronometer.setBase(SystemClock.elapsedRealtime());//设置起始时间
    chronometer.setFormat("已用时间: %s");               //设置显示时间的格式
    chronometer.start();                                //开启计时器
```

```
        chronometer.setOnChronometerTickListener(new chronometerListener());
        rest.setOnClickListener(new restOnClickListener());
    }
    class restOnClickListener implements OnClickListener{
        public void onClick(View v) {
            chronometer.setBase(SystemClock.elapsedRealtime());    //设置起始时间
            chronometer.start();                    //开启计时器
            rest.setVisibility(View.GONE);          //设置 Button 控件不可见
        }
    }
    class chronometerListener implements OnChronometerTickListener{
        public void onChronometerTick(Chronometer chronometer) {
        if (SystemClock.elapsedRealtime() - chronometer.getBase() >= 10000) {
            chronometer.stop();                     //停止计时器
            rest.setVisibility(View.VISIBLE);       //设置 Button 控件可见
            }
        }
    }
```

上述代码中，先获取到计时器控件，然后使用 setBase(SystemClock.elapsedRealtime())方法为计时器设置起始时间，setFormat()为计时器设置格式，然后使用 start()方法开启计时器。在监听器中判断如果(SystemClock.elapsedRealtime() - chronometer.getBase() >= 10000)为真，则使用 stop()方法来停止计时器，并且使用 setVisibility()方法设置 Button 控件可见。在按钮控件的监听器中为计时器添加了起始时间和开启计时器的方法，在单击按钮的时候，计时器将重新开始，同时 Button 控件设置为不可见。

运行该项目，计时器显示效果如图 5-24 所示。当计时器停止时如图 5-25 所示。



图 5-24　计时器运行时效果



图 5-25　计时器停止时效果

5.9 实例应用：设计用户注册界面

5.9.1 实例目标

根据本课所讲内容，设计一个用户注册界面，在其中要使用到本课所讲的基础控件，如文本框、编辑框、按钮、复选框等控件。

5.9.2 技术分析

首先在布局文件中使用控件的标记来配置所需要的各个控件，然后在主 Activity 中获取到该控件，给其添加监听器来监听其操作，最后在控制台输出所操作的内容。

5.9.3 实现步骤

在 Eclipse 中创建 Android 项目，名称为 ch05。设计一个用户注册界面，在其中要使用到文本框、编辑框、按钮、单选按钮、复选框、列表选择框、列表视图、图片视图等控件。

（1）在项目 res 目录下的 drawable_ldpi 文件夹中，放入两张名称分别为 logo.gif 和 background.gif 的图片，用于作为显示的 logo 图片和背景图片。

（2）在项目的 res\values 目录中新建数组资源文件 arrays.xml，在该文件中添加两个字符串数组，名称分别为 type 和 care，具体代码如下。

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string-array name="type">
        <item>学生</item>
        <item>老师</item>
        <item>白领</item>
        <item>工程师</item>
        <item>其他</item>
    </string-array>
    <string-array name="care">
        <item>1.保护用户个人信息是汇智的一项基本原则</item>
        <item>2.用户在本网站上不得发布违法信息</item>
        <item>3.保护好自己的账号和密码安全</item>
        <item>4.本网站所有权和解释权归本网站拥有</item>
    </string-array>
</resources>
```

（3）在项目的 res\values 目录中的 strings.xml 中添加如下代码。

```
<string name="edit01">请输入用户名</string>
<string name="edit02">请输入密码</string>
<string name="reg">注册</string>
<string name="rest">重置</string>
<string name="sex">请选择您的性别</string>
<string name="fav">请选择您的兴趣</string>
<string name="man">男</string>
<string name="woman">女</string>
<string name="checkText">我同意上述条款</string>
```

（4）在项目的 res/layout 目录下修改 activity_main.xml 文件，首先将界面整体布局改为表格布局，并设置背景，之后添加一个图像视图作为 logo 图像显示，代码如下。

```
<?xml version="1.0" encoding="utf-8"?>
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
```

```
    android:orientation="vertical"
    android:background="@drawable/background" >
    <ImageView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:src="@drawable/logo"
    android:id="@+id/imageView02"
    android:layout_margin="5dp"
    android:adjustViewBounds="true"
    android:maxWidth="75dp"
    android:maxHeight="50dp"/>
</TableLayout>
```

在上述代码中，android:adjustViewBounds 用于设置 ImageView 是否调整自己的边界来保持所需显示图片的长度变化，为 true 时表示调整自己的边界来保持所需显示图片的长度变化；android:maxWidth 和 android:maxHeight 分别表示 ImageView 的最大宽度和最大高度。

（5）添加 3 个 TableRow 表格行，并在其中添加 3 个文本框和编辑框控件，用来显示和填写用户名、密码和确认密码，这与练习 2 中的步骤（2）代码一样，在这里就省略了。

（6）添加一个线性布局，在其中添加一个文本框控件和一个单选按钮组，其中在单选按钮组中添加两个单选按钮控件，这与练习 4 中的步骤（1）代码一样，在这里省略，其中线性布局的 android:orientation 属性设置为"horizontal"。

（7）添加一个线性布局，在其中添加一个文本框控件和一个列表选择框控件，线性布局的 android:orientation 属性设置为"horizontal"，列表选择框控件代码与练习 5 中步骤（1）代码一样，在此省略，其中列表选择框的 android:entries 属性为"@array/care"。

（8）添加一个文本框控件、列表视图控件、复选框控件和一个普通按钮控件，代码如下。

```
<TextView
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:text="服务条款"
   android:gravity="center_horizontal"
   />
<!—省略列表视图控件的布局代码-->
<CheckBox
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:id="@+id/checkBox"
   android:text="@string/checkText"/>
<Button
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:id="@+id/reg"
   android:text="@string/reg"
   android:gravity="center_horizontal"
   android:visibility="invisible"/>
```

在上述代码中，文本框控件的 android:gravity="center_horizontal"表示文本框控件在屏幕中水平放置，android:textSize 表示字体大小；普通按钮中的 android:visibility 属性表示该控件是否可见，

在这里设置的是不可见。

（9）在 com.android.activity 包中的 MainActivity.java 文件中，获取到复选框控件、普通按钮控件、单选按钮组控件和列表选择框控件，并为它们添加监听器，其主要代码如下。

```
    //省略其他控件的定义
    private Button reg = null;                              //定义一个按钮控件
    private int position = -1;
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    //省略其他控件的获取
        reg = (Button) findViewById(R.id.reg);              //获取普通按钮控件
       ArrayAdapter<CharSequence> adapter = ArrayAdapter.createFromResource(this,
R.array.care, android.R.layout.simple_spinner_item);
        listView.setAdapter(adapter);                       //适配器与列表视图关联
        checkBox.setOnCheckedChangeListener(new checkBoxOnCheckedChange-
        Listener());//为复选框控件添加监听器
        sex.setOnCheckedChangeListener(new RadioGroup.OnCheckedChangeListener()
        {//为单选按钮组添加监听器
            public void onCheckedChanged(RadioGroup group, int checkedId) {
                 radio = (RadioButton) findViewById(checkedId);
            }
        });
        spinner.setOnItemSelectedListener(new spinnerOnItemSelectedListener());
                                                            //为列表选择框控件添加监听器
        reg.setOnClickListener(new regOnClickListener());
    }
    class checkBoxOnCheckedChangeListener implements OnCheckedChangeListener{
                                                            //复选框控件监听器
        public void onCheckedChanged(CompoundButton buttonView,
                boolean isChecked) {
            if (isChecked) {                                //复选框控件是否被选中
                reg.setVisibility(View.VISIBLE);            //注册按钮设置可见
            } else {
                reg.setVisibility(View.INVISIBLE);          //注册按钮设置不可见
            }
        }
    }
    class spinnerOnItemSelectedListener implements OnItemSelectedListener{
                                                          //下拉列表框控件监听器
        public void onItemSelected(AdapterView<?> arg0, View arg1, int arg2,long
        arg3)
        {
            position = arg2;                              //获取下拉列表框控件选中的位置
        }
        public void onNothingSelected(AdapterView<?> arg0) {
        }
    }
    class regOnClickListener implements OnClickListener{   //注册按钮监听器
```

```
        public void onClick(View v) {
            Log.i("您输入的用户名为: ", nickName.getText().toString());
            Log.i("您输入的密码为: ", pwd.getText().toString());
            Log.i("您输入的确认密码为: ", pwd2.getText().toString());
            if (radio != null) {
                Log.i("您选择的性别是: ", radio.getText().toString());
            }else{
                Log.i("您选择的性别是: ", "无");
            }
            Log.i("您选择的身份是: ",spinner.getItemAtPosition(position).
            toString());
        }
    }
```

在上述代码中，通过下拉列表框控件的监听器来获取所选内容的位置，然后赋值给 position 变量；在复选框控件的监听器中，如果该复选框被选中，则注册按钮显示可见，否则不可见。

运行该项目，效果如图 5-26 所示。填写信息后的效果如图 5-27 所示。单击注册后控制台输出的信息如图 5-28 所示。

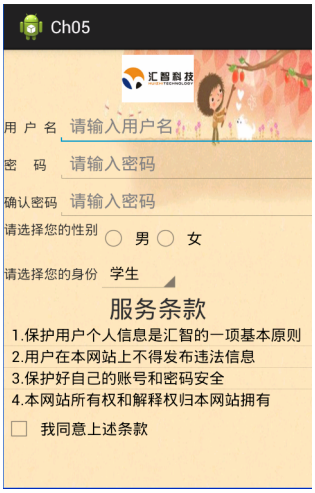


图 5-26　项目运行效果



图 5-27　填写信息后效果

Level	Time	PID	TID	Application	Tag	Text
I	08-1...	2020	2020	com.android.activity	您输入的用户名为:	admin
I	08-1...	2020	2020	com.android.activity	您输入的密码为:	admin888
I	08-1...	2020	2020	com.android.activity	您输入的确认密码为:	admin888
I	08-1...	2020	2020	com.android.activity	您选择的性别是:	男
I	08-1...	2020	2020	com.android.activity	您选择的身份是:	工程师

图 5-28　控制台输出信息

5.10 扩展训练

拓展训练 1：实现图标在上、文字在下的 ListView

根据本节所学内容，创建一个 Android 项目，要求实现图标在上、文字在下的 ListView。

5.11 课后练习

一、填空题

1. 在 Android 的布局中，用于设置文本框内文本的对齐方式属性的是_____。
2. 在编辑框中需要输入的文本为密码时，需要将 inputType 属性值设置为_____。
3. 在 Android 的布局中，_____是编辑框为空时显示的文字提示信息的属性。
4. 在 Android 的布局中，_____是设置该控件显示或者不显示的属性。

二、选择题

1. 下列 Android 控件中，不属于基础控件的是_____。
   A. TextView
   B. Button
   C. ProgressBar
   D. Chronometer
2. 在使用下列 Android 控件时，_____不需要添加监听器。
   A. TextView
   B. Button
   C. ListView
   D. RadioButton
3. 在使用 DatePicker 对象获取到月份 month 时，需要将 month_____才能获取到真正的月份。
   A. 减 1
   B. 加 1
   C. 不操作
   D. 加 2
4. 在使用属性 typeface 设置文本字体的时候，下列_____不是该属性的值。
   A. normal
   B. sans
   C. serif
   D. nospace

三、简答题

1. 在屏幕中添加控件时，除了使用 XML 布局文件中添加标记，是否有其他方法呢？请详细说明。
2. 在使用 Android 基础控件时，必须添加监听器吗？如果不是，请给出例子说明。
3. 分别使用 ListView 控件五种指定的外观形式来显示列表视图，并说出这几种样式的区别。