

智能手机应用软件开发技术教案

第 2 次课 第 2 章 Android 用户界面设计

授课班级： 计算机 21-1

教学方式： PPT 教学+ 课堂实例教学 + 学生课堂练习 同步方式

教学要点： 学习 **Android** 基本组件的设计与点击事件的设计，学习 **Android** 界面布局方式，学习设计计算器及事件处理，完成拼图游戏的界面设计

教学过程： 需注意同学们实践中对环境的理解与掌握，带领学生完成拼图游戏中界面的设计实践。

实践源代码见 **PinTu.zip** 代码分享给同学。

【例 2-1】设计一个文本标签组件程序。

(1) 设计资源文件：strings.xml

```
<resources>
    <string name="app_name">ex2_1</string>
    <string name="hello">
        \n      荷塘月色
        \n  剪一段时光缓缓流淌，
        \n  弹一首小荷淡淡的香，
        \n  美丽的琴音就落在我身旁。
    </string>
</resources>
```

(2) 界面布局文件 activity_main.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.ex2_1.MainActivity">

    <TextView
        android:layout_width="wrap_content"
```

```

        android:layout_height="wrap_content"
        android:text="@string/hello"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="59dp"
        android:textSize="22sp"
        android:id="@+id/textView" />
</RelativeLayout>

```

(3) 控制文件 MainActivity.java

```

package com.example.ex2_1;

import android.graphics.Color;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {
    TextView txt;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        txt = (TextView)findViewById(R.id.textView);
        txt.setTextColor(Color.BLUE);
    }
}

```

【例 2-2】 编写程序，当点击按钮命令后，页面标题及文本组件的文字内容发生变化。

(1) 布局文件： activity_main.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.ex2_2.MainActivity">

    <TextView

```

```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World, 这是 ex2_2 的界面!"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="36dp"
        android:textSize="18sp"
        android:id="@+id/textView" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="点击我! "
    android:id="@+id/button"
    android:layout_below="@+id/textView"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="85dp"
    android:textSize="20sp" />
</RelativeLayout>

```

(2) 资源文件: strings.xml

```

<resources>
    <string name="app_name">ex2_2</string>
    <string name="newStr">改变了文本标签的内容</string>
</resources>

```

(3) 控制文件: MainActivity.java

```

package com.example.ex2_2;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

import org.w3c.dom.Text;

public class MainActivity extends AppCompatActivity {
    //首先声明与界面相同的组件
    TextView txt;
    Button btn;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }
}

```

```

        setContentView(R.layout.activity_main);
        //建立控制层与表现层（布局文件）组件之间的关联
        txt = (TextView)findViewById(R.id.textView);
        btn = (Button)findViewById(R.id.button);
        //注册监听接口
        btn.setOnClickListener(new mClick());
    }
    //定义实现监听接口的内部类
    class mClick implements View.OnClickListener{
        //实现接口的方法
        @Override
        public void onClick(View v) {
            MainActivity.this.setTitle("改变标题");
            txt.setText(R.string.newStr);    //把要改变的文字内容设置到 strings.xml 文件
        }
    }
}

```

中

【例 2-3】编写程序，点击按钮命令，改变文本组件的文字及背景颜色。

(1) 布局文件 activity_main.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.ex2_3.MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        android:layout_alignParentTop="true"
        android:layout_alignParentStart="true"
        android:layout_marginStart="104dp"
        android:textSize="18sp"
        android:id="@+id/textView" />

    <Button

```

```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/button"
        android:id="@+id/button"
        android:layout_below="@+id/textView"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="54dp"
        android:textSize="16sp" />
</RelativeLayout>

```

(2) 资源文件: strings.xml

```

<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="hello">Hello World, MainActivity!</string>
    <string name="app_name">Ex2_3</string>
    <string name="button">点击我，改变文字背景颜色</string>
</resources>

```

(3) 控制文件: MainActivity.java

```

package com.example.ex2_3;

import android.graphics.Color;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {
    //声明与界面布局相同的组件
    TextView txt;
    Button btn;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        //建立与界面布局中组件的关联
        txt = (TextView)findViewById(R.id.textView);
        btn = (Button)findViewById(R.id.button);
        btn.setOnClickListener(new mClick());
    }
    //定义实现监听接口的内部类
    class mClick implements View.OnClickListener{

```

```

@Override
public void onClick(View v) {
    int BLACK = 0xffccccc;
    txt.setText("改变了文字及背景颜色");
    txt.setTextColor(Color.YELLOW); //设置文字的颜色我黄色
    txt.setBackgroundColor(BLACK); //设置文本标签背景颜色
}
}
}

```

【例 2-4】设计一个密码验证程序。

(1) 设计布局文件 activity_main.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.ex2_4.MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="请输入密码: "
        android:layout_alignParentTop="true"
        android:layout_alignParentStart="true"
        android:textSize="16sp"
        android:id="@+id/textView" />

    <EditText
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:id="@+id/editText"
        android:layout_below="@+id/textView"
        android:layout_alignParentStart="true"
        android:layout_marginTop="31dp" />

    <Button
        android:layout_width="wrap_content"

```

```

        android:layout_height="wrap_content"
        android:text="确定"
        android:id="@+id/button"
        android:layout_below="@+id/editText"
        android:layout_alignParentStart="true"
        android:layout_marginTop="37dp"
        android:textSize="16sp" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/textView2"
    android:layout_centerVertical="true"
    android:layout_alignParentStart="true"
    android:textSize="16sp" />
</RelativeLayout>

```

(2) 设计控制文件 MainActivity.java

```

package com.example.ex2_4;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {
    EditText edit;
    TextView txt;
    Button mBtn;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        edit = (EditText)findViewById(R.id.editText);
        txt = (TextView)findViewById(R.id.textView2);
        mBtn = (Button)findViewById(R.id.button);
        mBtn.setOnClickListener(new mClick());
    }
    class mClick implements View.OnClickListener{
        @Override
        public void onClick(View v) {
            String passwd;

```

```

        passwd = edit.getText().toString(); //获取文本编辑框中的文本内容
        if(passwd.equals("abc123")) //判断两个字符串是否相等
            txt.setText("欢迎进入快乐大本营! ");
        else
            txt.setText("非法用户, 请立刻离开! ");
    }
}
}

```

【例 2-5】线性布局应用示例。

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      android:layout_width="fill_parent"
4      android:layout_height="fill_parent"
5      android:orientation="vertical" >
6      <!-- android:orientation="horizontal" -->
7      <Button
8          android:id="@+id/mButton1"
9          android:layout_width="60px"
10         android:layout_height="wrap_content"
11         android:text="按钮 1" />
12     <Button
13         android:id="@+id/mButton2"
14         android:layout_width="60px"
15         android:layout_height="wrap_content"
16         android:text="按钮 2" />
17     <Button
18         android:id="@+id/mButton3"
19         android:layout_width="60px"
20         android:layout_height="wrap_content"
21         android:text="按钮 3" />
22     <Button
23         android:id="@+id/mButton4"
24         android:layout_width="60px"
25         android:layout_height="wrap_content"
26         android:text="按钮 4" />
27 </LinearLayout>

```

【例 2-6】帧布局应用示例。

将事先准备的图像文件 `img.png` 复制到 `res\drawable` 目录下。

(1) 布局文件 activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <ImageView
        android:id="@+id/mlImageView"
        android:layout_width="60px"
        android:layout_height="wrap_content"
    />
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="快乐大本营"
        android:textSize="18sp"
    />
</FrameLayout>
```

(2) 控制文件 MainActivity.java

```
package com.ex2_6;
import android.app.Activity;
import android.os.Bundle;
import android.widget.ImageView;
public class MainActivity extends Activity
{
    ImageView imageview;
    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        imageview = (ImageView) this.findViewById(R.id.mlImageView);
        imageview.setImageResource(R.drawable.img);
    }
}
```

【例 2-7】 表格布局应用示例。设计一个 3 行 4 列的表格布局。

(1) 表格布局的布局文件 activity_main.xml

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

```

        android:layout_height="fill_parent">
<TableRow>    <!-- 第 1 行 -->
<ImageView    android:id="@+id/mlImageView1"
        android:layout_width="wrap_content "
        android:layout_height="wrap_content"
        android:src="@drawable/img1" />
<ImageView    android:id="@+id/mlImageView2"
        android:layout_width="wrap_content "
        android:layout_height="wrap_content"
        android:src="@drawable/img2" />
</TableRow>

```

```

<TableRow>    <!-- 第 2 行 -->
<TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
<ImageView    android:id="@+id/mlImageView3"
        android:layout_width="wrap_content "
        android:layout_height="wrap_content"
        android:src="@drawable/img3" />
<ImageView    android:id="@+id/mlImageView4"
        android:layout_width=" wrap_content "
        android:layout_height="wrap_content"
        android:src="@drawable/img4" />
</TableRow>

```

```

<TableRow>    <!-- 第 3 行 -->
<TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
<TextView
        android:id="@+id/textView3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
<TextView
        android:id="@+id/textView4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
<ImageView    android:id="@+id/mlImageView5"
        android:layout_width="wrap_content "
        android:layout_height="wrap_content"
        android:src="@drawable/img5" />
</TableRow>

```

</TableLayout>

(2) 控制文件 MainActivity.java

```
package com.ex02_07;
import android.app.Activity;
import android.os.Bundle;
import android.widget.ImageView;

public class MainActivity extends Activity
{
    ImageView img1, img2, img3, img4, img5;
    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        img1 = (ImageView) this.findViewById(R.id.mImageView1);
        img2 = (ImageView) this.findViewById(R.id.mImageView2);
        img3 = (ImageView) this.findViewById(R.id.mImageView3);
        img4 = (ImageView) this.findViewById(R.id.mImageView4);
        img5 = (ImageView) this.findViewById(R.id.mImageView5);
        img1.setImageResource(R.drawable.img1);
        img2.setImageResource(R.drawable.img2);
        img3.setImageResource(R.drawable.img3);
        img4.setImageResource(R.drawable.img4);
        img5.setImageResource(R.drawable.img5);
    }
}
```

【例 2-8】应用相对布局设计一个组件排列如图 2.13 所示的应用程序。

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <TextView
        android:id="@+id/label"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:textSize="24sp"
        android:text="相对布局"/>
    <EditText
        android:id="@+id/edit"
        android:layout_width="fill_parent"
```

```
        android:layout_height="wrap_content"
        android:background="@android:drawable/editbox_background"
        android:layout_below="@id/label"/>
<Button
    android:id="@+id/ok"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/edit"
    android:layout_alignParentRight="true"
    android:layout_marginLeft="10dip"
    android:text="OK" />
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_toLeftOf="@id/ok"
    android:layout_alignTop="@id/ok"
    android:text="Cancel" />
</RelativeLayout>
```