第 7 次课　　第 3 章　多个用户界面的程序设计

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：　菜单与对话框
教学过程：　需注意同学们实践中对环境的理解与掌握，带领学生完成实践过程非常重要，特别需要注意，对常期未从事开发工作的同学的培养，去除其恐惧心理。加强创新思维理解与培养，在多用户界面设计中时该体现用户需求与人机工学特点，减少繁杂的交互，为用户着想，提升自身创新能力。

【例 3-3】选项菜单应用示例。

```
package com.ex03_03;
import android.app.Activity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.TextView;
public class MainActivity extends Activity
 {
    TextView txt;
    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
     txt = (TextView)findViewById(R.id.TextView1);
    }
    @Override
    public boolean onCreateOptionsMenu(Menu menu)
    {
      // 调用父类方法来加入系统菜单
      super.onCreateOptionsMenu(menu);
      // 添加菜单项
      menu.add(
                1,          //组号
                1,          //唯一的 ID 号
                1,          //排序号
                "菜单项 1"); //标题
       menu.add( 1, 2, 2,  "菜单项 2");
      menu.add( 1, 3, 3,   "菜单项 3");
      menu.add( 1, 4, 4,   "菜单项 4");
      return true;
   }
    @Override
   public boolean onOptionsItemSelected(MenuItem item)
   {
     String title = "选择了" + item.getTitle().toString();
     switch (item.getItemId())
      { //响应每个菜单项(通过菜单项的 ID)
        case 1:
             txt.setText(title);
             break;
        case 2:
             txt.setText(title);
             break;
        case 3:
             txt.setText(title);
             break;
        case 4:
             txt.setText(title);
             break;
         default:
              //对没有处理的事件，交给父类来处理
             return super.onOptionsItemSelected(item);
       }
      return true;
    }
 }
```

【例 3-4】上下文菜单应用示例。

```
  package com.ex3_4;
  import android.app.Activity;
  import android.os.Bundle;
  import android.view.ContextMenu;
  import android.view.ContextMenu.ContextMenuInfo;
  import android.view.Menu;
  import android.view.MenuItem;
  import android.view.View;
  import android.widget.ListView;
 import android.widget.TextView;
 public class MainActivity extends Activity
 {
```

```
TextView txt1, txt2, txt3;
private static final int item1 = Menu.FIRST;
  private static final int item2 = Menu.FIRST+1;
  private static final int item3 = Menu.FIRST+2;
String   str[] = {"  令狐冲", "杨    过", "萧    峰  "   };
@Override
public void onCreate(Bundle savedInstanceState)
{
  super.onCreate(savedInstanceState);
  setContentView(R.layout.activity_main);
  txt1=(TextView)findViewById(R.id.textView1);
  txt2=(TextView)findViewById(R.id.textView2);
  txt3=(TextView)findViewById(R.id.textView3);
  txt1.setText(str[0].toString());
  txt2.setText(str[1].toString());
  txt3.setText(str[2].toString());
  registerForContextMenu(txt1);
  registerForContextMenu(txt2);
  registerForContextMenu(txt3);
}
//上下文菜单，本例会通过长按条目激活上下文菜单
@Override
public void onCreateContextMenu(ContextMenu menu, View view,
ContextMenuInfo menuInfo) {
     menu.setHeaderTitle("人物简介");
     //添加菜单项
     menu.add(0, item1, 0, "武功");
     menu.add(0, item2, 0, "战斗力");
     menu.add(0, item3, 0, "经典语录");
 }
//菜单单击响应
@Override
public boolean onContextItemSelected(MenuItem item){
  //获取当前被选择的菜单项的信息
  switch(item.getItemId())
  {
    case item1:
        //在这里添加处理代码
        break;
    case item2:
        //在这里添加处理代码
        break;
    case item3:
        //在这里添加处理代码
```

```
            break;
        }
        return true;
    }
  }
```

【例 3-5】消息对话框应用示例。

(1) 主界面

2 个按钮

(2) 登录对话框的界面布局文件 login.xml

```
<?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="wrap_content"
        android:layout_height="wrap_content"
        android:text="用户名:"
        android:id="@+id/user"
        android:textSize="18sp" />

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/editText" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="密码："
        android:id="@+id/textView"
        android:textSize="18sp" />

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/paswdEdit" />
</LinearLayout>
```

(3) 主控制程序

```
public class MainActivity extends Activity {
    ProgressDialog mydialog;
    Button btn1,btn2;
    LinearLayout login;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        btn1=(Button)findViewById(R.id.button);
        btn2=(Button)findViewById(R.id.button2);
        btn1.setOnClickListener(new mClick());
        btn2.setOnClickListener(new mClick());
    }
    class mClick implements View.OnClickListener
    {
        AlertDialog.Builder dialog=new AlertDialog.Builder(MainActivity.this);
        @Override
        public void onClick(View arg0)
        {
            if(arg0 == btn1)
            {
                //设置对话框的标题
                dialog.setTitle("警告");
                //设置对话框的图标
                dialog.setIcon(R.drawable.icon1);
                //设置对话框显示的内容
                dialog.setMessage("本项操作可能导致信息泄漏！ ");
                //设置对话框的“确定”按钮
                dialog.setPositiveButton("确定", new okClick());
                //创建对象框
                dialog.create();
                //显示对象框
                dialog.show();
            }
            else   if(arg0 == btn2)
            {
                login = (LinearLayout)getLayoutInflater().inflate(R.layout.login, null);

                dialog.setTitle("用户登录").setMessage("请输入用户名和密码")
```

```
                    .setView(login);
            dialog.setPositiveButton("确定", new loginClick());
            dialog.setNegativeButton("退出", new exitClick());
            dialog.setIcon(R.drawable.icon2);
            dialog.create();
            dialog.show();
        }
    }
}
/*   普通对话框的“确定”按钮事件 */
class okClick implements DialogInterface.OnClickListener
{
    @Override
    public void onClick(DialogInterface dialog, int which)
    {
        dialog.cancel();
    }
}
/*   输入对话框的“确定”按钮事件    */
class loginClick implements   DialogInterface.OnClickListener
{
    EditText txt;
    @Override
    public void onClick(DialogInterface dialog, int which)
    {
        txt = (EditText)login.findViewById(R.id.paswdEdit);
        //取出输入编辑框的值与密码“admin”比较
        if((txt.getText().toString()).equals("admin"))
            Toast.makeText(getApplicationContext(),
                    "登录成功", Toast.LENGTH_SHORT).show();
        else
            Toast.makeText(getApplicationContext(),
                    "密码错误", Toast.LENGTH_SHORT).show();
        dialog.dismiss();
    }
}
/*   输入对话框的“退出”按钮事件    */
class exitClick implements DialogInterface.OnClickListener
{
    @Override
    public void onClick(DialogInterface dialog, int which)
    {
        MainActivity.this.finish();
    }
    }

}
```

【例 3-6】进度及日期、时间对话框示例。

```
   package com.example.ex3_6;
   import android.app.Activity;
   import android.app.DatePickerDialog;
   import android.app.ProgressDialog;
   import android.app.TimePickerDialog;
   import android.app.DatePickerDialog.OnDateSetListener;
   import android.app.TimePickerDialog.OnTimeSetListener;
   import android.os.Bundle;
   import android.view.View;
  import android.view.View.OnClickListener;
  import android.widget.Button;
  import android.widget.DatePicker;
  import android.widget.TimePicker;

  public class MainActivity extends Activity
  {
      Button btn1,btn2,btn3;
    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        btn1=(Button)findViewById(R.id.button1);
        btn2=(Button)findViewById(R.id.button2);
        btn3=(Button)findViewById(R.id.button3);
        btn1.setOnClickListener(new mClick());
        btn2.setOnClickListener(new mClick());
        btn3.setOnClickListener(new mClick());
    }
    class mClick implements OnClickListener
    {
        int m_year = 2012;
        int m_month = 1;
        int m_day = 1;
        int m_hour = 12,     m_minute = 1;
        @Override
    public void onClick(View v)
  {
   if(v == btn1)
{
       ProgressDialog   d=new ProgressDialog (MainActivity.this);
       d.setTitle("进度对话框");
       d.setIndeterminate(true);
       d.setMessage("程序正在 Loading...");
       d.setCancelable(true);
       d.setMax(10);
       d.show();
     }
  else   if(v == btn2)
     {
        //设置日期监听器
        OnDateSetListener dateListener = new OnDateSetListener()
         {
             @Override
              public void onDateSet(DatePicker view, int year,
                                    int monthOfYear, int dayOfMonth)
             {
                 m_year = year;
                 m_month = monthOfYear;
                 m_day = dayOfMonth;
              }
         };
         //创建日期对话框对象
         DatePickerDialog date = new DatePickerDialog(MainActivity.this,
                  dateListener, m_year, m_month, m_day);
         date.setTitle("日期对话框");
         date.show();
   }
  else   if(v == btn3)
  {     //设置时间监听器
     OnTimeSetListener timeListener = new OnTimeSetListener()
      {
          @Override
         public void onTimeSet(TimePicker view, int hourOfDay, int minute)
          {
             m_hour = hourOfDay;
             m_minute = minute;
          }
       };
      TimePickerDialog d = new TimePickerDialog(MainActivity.this,
                            timeListener, m_hour, m_minute, true);
```

```
                d.setTitle("时间对话框");
                d.show();
        }
          }
    }
}
```