## 实验二 组件间通信

指导老师: 李明霞、刘娜

- 一、 实验目的:
- 1、掌握事件的处理方法。
- 2、掌握Activity的生命周期。
- 3、掌握组件间的数据传递方法。
- 4、数据回传的方法。
- 二、 实验内容:
- 1、分别用四种方式实现事件的处理。

```
通过对主页导航栏点击显示底部提示栏的方式对四种事件处理
//方法一
ButtonClick listener = new ButtonClick();
btn1.setOnClickListener(listener);
class ButtonClick implements View.OnClickListener{
   @Override
   public void onClick(View view) {
      Toast.makeText(MainActivity.this,"这是首页按钮",Toast.LENGTH_L
ONG).show();
   }
}
//方法二
btn2.setOnClickListener(new View.OnClickListener(){
   @Override
   public void onClick(View view) {
       Toast.makeText(MainActivity.this,"这是手机界面按钮",Toast.LE
NGTH_LONG();
   }
});
//方法三
btn3.setOnClickListener(this);
public void onClick(View v)
```

```
{
   Toast.makeText(MainActivity.this,"这是食品频道按钮",Toast.LENGTH_L
ONG).show();
}
//方法四
btn4 = (Button)findViewById(R.id.shengxianBottom);
public void onClick(View v)
{
   Toast.makeText(MainActivity.this,"这是数码频道按钮",Toast.LENGTH_L
ONG).show();
}
//实验效果截图
   潮流服饰
        京东到家
                           数码电器
                                       京东秒杀 14 8 00:32:43 京东直播 🕕 梅西20点
  京东秒杀 14 8 00:32:43 京东直播 10 梅西20点
                                                这是食品频道按钮
                         AT MEXAGE
             这是首页按钮
                                               88
           8
                                                       购物车
                   购物车
                           我的
                                                    0
                                                             京东到家
                                                    借钱
                                                         领券
                                                              数码电器
                          数码电器
                                       京东秒杀 14 8 00:32:43 京东直播 10 梅西20点
  京东秒杀 14条 00:32:43 京东直播 111 梅西20点
                                               这是数码频道按钮
           这是手机界面按钮
                                               8
           88
    首页
                  购物车
                                           Δ
                                                    0
                                                             \nabla
                0
```

2、设计验证Activity的生命周期的7个方法。

```
//方法: 设置 Log 属性对各生命周期进行监测
//举例如下:
@Override
protected void onResume() {
    super.onResume();
```

```
Log.i("MainActivity","调用 onResume");
}
//对各情况检测结果如下:
应用程序启动时:
      MainActivity: 调用 onCreate
      MainActivity: 调用 onStart
      MainActivity: 调用 onResume
跳转至下一界面时:
      MainActivity: 调用 onPause
      SecondActivity: 调用 onCreate
      SecondActivity: 调用 onStart
      SecondActivity: 调用 onResume
      MainActivity: 调用 onStop
退回上一界面时:
      SecondActivity: 调用 onPause
      MainActivity: 调用 onRestart
      MainActivity: 调用 onStart
      MainActivity: 调用 onResume
      SecondActivity: 调用 onStop
      SecondActivity: 调用 onDestroy
实验环境截图:
Logcat
  III Emulator Pixel_API_22 Android 5 ▼ cn.dlpu.cs.yulong.app_1025 (15444 ▼ Debug
                                                                            ▼ Q MainActivity|SecondActivity|ThirdActivity
 🝵 10-26 09:23:04.796 1539-2231/system_process I/ActivityManager: START u0 {act=android.intent.action.MAIN cat=[android.i
10-26 09:23:04.815 1539-2854/system_process I/ActivityManager: Start proc 15444:cn.dlpu.cs.yulong.app_1025/u0a61 for a 10-26 09:23:05.186 15444-15444/cn.dlpu.cs.yulong.app_1025 I/MainActivity: 调用onCreate
 ↑ 10-26 09:23:05.188 15444-15444/cn.dlpu.cs.yulong.app_1025 I/MainActivity: 调用onStart
 」 10-26 09:23:05.188 15444-15444/cn.dlpu.cs.yulong.app_1025 I/MainActivity: 调用onResume
10-26 09:23:05.653 1539-1568/system_process I/ActivityManager: Displayed cn.dlpu.cs.yulong.app_1025/.MainActivity: +85 10-26 09:24:32.730 1539-2157/system_process I/ActivityManager: START u0 {cmp=cn.dlpu.cs.yulong.app_1025/.SecondActivity
 = 10-26 09:24:32.741 15444-15444/cn.dlpu.cs.yulong.app_1025 I/MainActivity: 调用onPause
 10-26 09:24:32.764 15444-15444/cn.dlpu.cs.yulong.app_1025 I/SecondActivity: 调用onCreate
    10-26 09:24:32.765 15444-15444/cn.dlpu.cs.yulong.app_1025 I/SecondActivity: 调用onStart
 ☆ 10-26 09:24:32.765 15444-15444/cn.dlpu.cs.yulong.app_1025 I/SecondActivity: 调用onResume
 10-26 09:24:32.970 1539-1568/system_process I/ActivityManager: Displayed cn.dlpu.cs.yulong.app_1025/.SecondActivity: +
    10-26 09:24:33.325 15444-15444/cn.dlpu.cs.yulong.app_1025 I/MainActivity: 调用onStop
 10-26 09:26:47.871 15444-15444/cn.dlpu.cs.yulong.app_1025 I/SecondActivity: 调用onPause
 ■ 10-26 09:26:47.875 15444-15444/cn.dlpu.cs.yulong.app_1025 I/MainActivity: 调用onRestart
    10-26 09:26:47.875 15444-15444/cn.dlpu.cs.yulong.app 1025 I/MainActivity: 调用onStart
 ? 10-26 09:26:47.875 15444-15444/cn.dlpu.cs.yulong.app_1025 I/MainActivity: 调用onResume
     10-26 09:26:48.561 15444-15444/cn.dlpu.cs.yulong.app_1025 I/SecondActivity: 调用onStop
     10-26 09:26:48.562 15444-15444/cn.dlpu.cs.yulong.app_1025 I/SecondActivity: 调用onDestroy
```

3、分别使用Intent和Bundle实现组件间的数据传递。

## //设置数据传递(Bundle 方法写在了回传中)

```
protected void onCreate(@Nullable Bundle savedInstanceState) {
   super.onCreate(savedInstanceState);
   setContentView(R.layout.layout_gouwuche);
   Log.i("SecondActivity","调用 onCreate");
   bt xd.setOnClickListener(new View.OnClickListener() {
       @Override
       public void onClick(View view) {
          string = redo();
                               //字符串的制作
          intent = new Intent (SecondActivity.this, ThirdActivity.class
);
          intent.putExtra("string",string);
          startActivity(intent);
       }
   });
}
//获取传递数据
protected void onCreate(@Nullable Bundle savedInstanceState) {
   super.onCreate(savedInstanceState);
   setContentView(R.layout.xdlayout);
   Log.i("ThirdActivity","调用 onCreate");
   intent = getIntent();
   String str = intent.getStringExtra("string");
   TextView textView = (TextView)findViewById(R.id.str);
   textView.setText(str);
}
实验验证:
✓ 棉拖鞋
                                    您选择的商品是
□ 高领毛衣
二 洗脸巾
                                     棉拖鞋
✓ 鞋
                                      鞋
□ 教材
```

4、实现组件间数据的回传。

```
//设置回传数据(写在 Activity 的 onStop 生命周期中)
@Override
protected void onStop() {
   super.onStop();
   Log.i("ThirdActivity","调用 onStop");
   str1 = "在"+getTime()+"时, 您进行了一次消费。\n";
   reIntent = new Intent(this, SecondActivity.class);
   Bundle bundle = reIntent.getExtras();
   bundle.putString("str1",str1);
   reIntent.putExtras(bundle);
   this.setResult(1,reIntent);
   this.finish();
//接收回传数据(写入onRestart生命周期中)
@Override
protected void onRestart() {
   super.onRestart();
   Log.i("SecondActivity","调用 onRestart");
   reIntent = new Intent(this, ThirdActivity.class);
   startActivityForResult(reIntent,1);
}
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent
reIntent) {
   super.onActivityResult(requestCode, resultCode, reIntent);
   if (reIntent!=null)
   {
       if(requestCode==1)
       {
          if (resultCode==1)
          {
              Bundle bundle = reIntent.getExtras();
              stra = bundle.getString("str1");
              Toast.makeText(this, stra, Toast.LENGTH_SHORT).show();
          }
       }
```



## 三、实验总结

通过本次实验重新复习了在Android开发过程中对Activity中事件的处理方法,可以通过四种方法设置事件启动,并引导进行相应操作。同时复习了Activity的生命周期,并通过设置Log方法在控制台中进行监测。此外、通过本次实验对数据的传递和回传进行复习和编写,完成预期效果。