



上海建桥学院

SHANGHAI JIAN QIAO UNIVERSITY

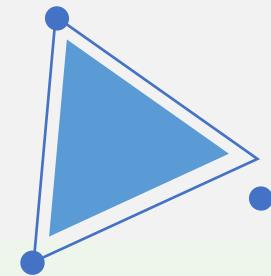


Activity数据回传

主讲：万永权



Activity之间数据回传



案例1

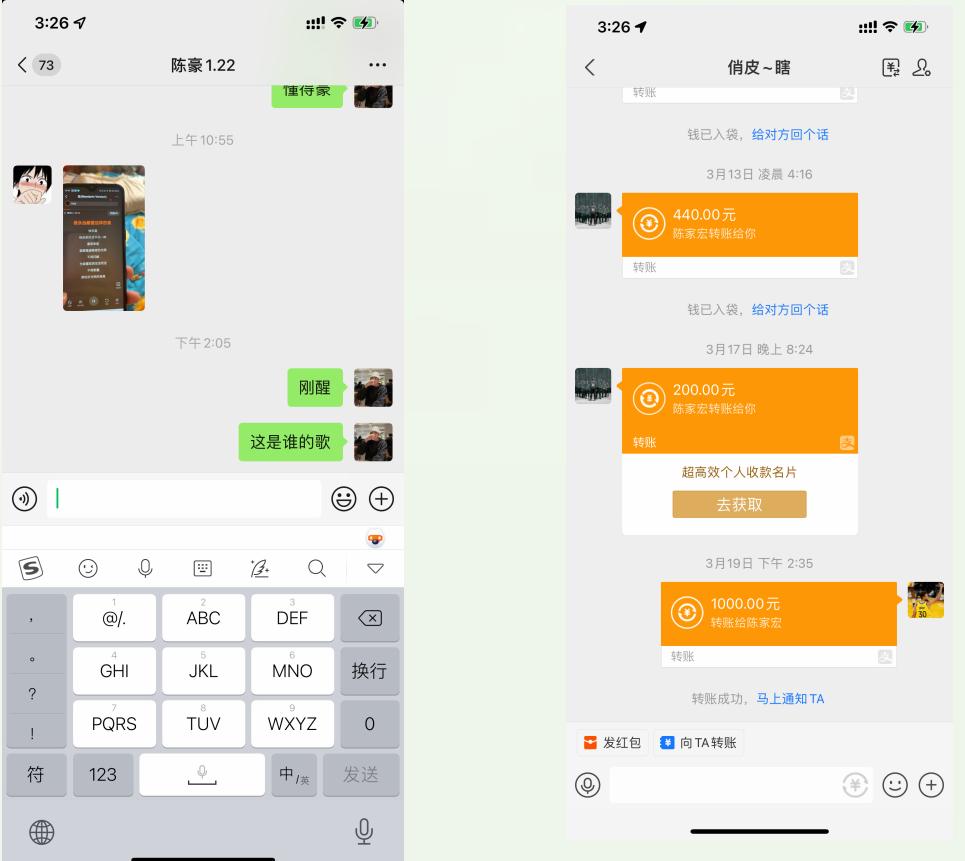
■如上图所示，在用户下单之前需要先选定下单的门店，提货方式等信息，之后再选定产品，在用户即将支付时，会弹出一个界面提示用户确定下单门店和提货方式是否无误，在这两个界面里，传递的数据为：下单门店“建桥学院店”和提货方式“自提”。



本案例由计科B19-5 同学：郝昀、王磊、毛鸿杰、金恒笑 提供

反例

■注意：以下情况有数据的传递，比如微信互相发消息 支付宝互相转账都是数据交换，但**不是**本章所讲的Activity之间的数据传递。



本案例由计科B19-5 同学：黄思宇、植冠儒、刘宇辰、莫斯捷提供



4.4.2 Activity之间的数据回传





>>> 4.4.2 Activity之间的数据回传

Activity之间进行数据回传时包含3个方法，分别是startActivityForResult()方法、
setResult()方法和onActivityResult()方法。

(1) startActivityForResult()方法

用于开启一个Activity，当开启的Activity销毁时，会从销毁的Activity中返回数据。

startActivityForResult(Intent intent, int requestCode)

↓

意图对象

表示请求码，用于标识请求的来源

(2) setResult() 方法

用于携带数据进行回传，该方法的语法格式如下：

setResult(int resultCode, Intent intent)

↓

表示返回码，用于标识返回的数据来自哪一个Activity



>>> 4.4.2 Activity之间的数据回传

(3) onActivityResult()方法

用于接收回传的数据，该方法的语法格式如下：

```
onActivityResult(int requestCode, int resultCode, Intent data)  
    ↓      ↓      ↓  
  请求码  回传码  回传数据
```

程序会根据传递的参数requestCode与resultCode来识别数据的来源。



// 流程概括

- 第一个活动通过startActivityForResult()方法来启动（关键：唯一的请求码）

- 第二个活动 setResult() 设置数据返回处理结果

- 第一个活动onActivityResult() 与startActivitiyResult() 对应，来获取返回的数据。



>>> 4.4.2 Activity之间的数据回传

```
Intent intent = new Intent(MainActivity.this,SecondActivity.class);
startActivityForResult(intent,1); → 开启SecondActivity
```

```
Intent intent = new Intent();
intent.putExtra("data","Hello MainActivity");
 setResult(2,intent); → 在SecondActivity 中添加返回数据
finish();
```

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == 1) { SecondActivity被销毁之后在MainActivity
        String acquiredData = data.getStringExtra("data"); 中回调onActivityResult()方法，接收回传的
        Toast.makeText(MainActivity.this,acquiredData,toast.LENGTH_SHORT).show();
    }
}
```

4.4.3 实战演练—小猴子摘桃

本节我们将通过一个小猴子摘桃的案例来演示Activity之间是如何进行数据回传的，本案例的界面效果如下图所示。

① 创建PickPeach程序

② 导入界面图片

③ 放置界面控件

1 搭建首页界面布局：④ 去掉默认标题栏

① 创建桃园界面

② 导入界面图片

2 搭建桃园界面布局：③ 放置界面控件

① 实现首页界面的显示效果

3 实现小猴子摘桃的功能：② 实现桃园界面的摘桃效果

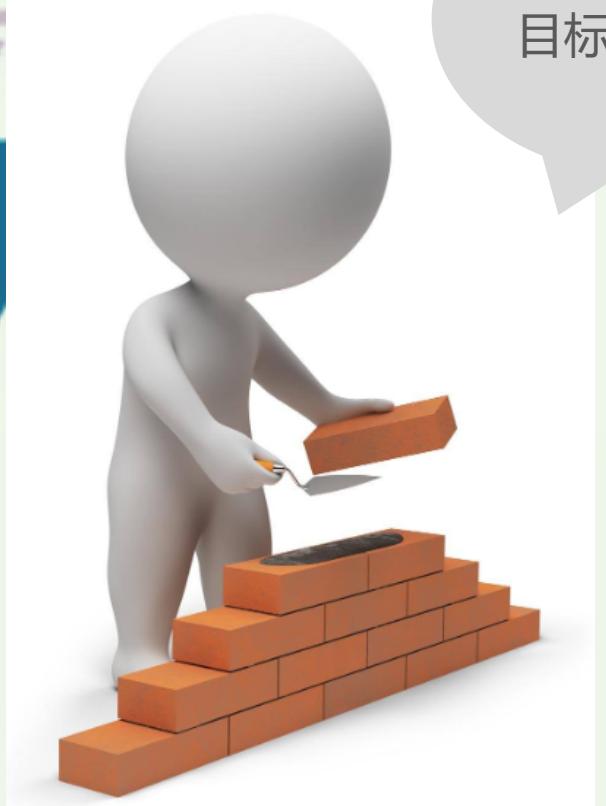




4.5

任务栈和启动模式

4.5 任务栈和启动模式



- 熟悉Android中的任务栈，能够归纳Activity在任务栈中的存放情况
- 熟悉Activity的启动模式，能够解释Activity的4种启动模式

4.5.1 Android中的任务栈

任务栈：一种用来存放Activity实例的容器。

特点：“先进后出”

操作：压栈和出栈



4.5.2 Activity的启动模式

Activity启动模式有四种，分别是standard、singleTop、singleTask和singleInstance模式。

(1) standard模式

standard模式是Activity的默认启动方式，每启动一个Activity就会在栈顶创建一个新的实例。



4.5.2 Activity的启动模式

(2) singleTop模式

singleTop模式会判断要启动的Activity实例是否位于栈顶，如果位于栈顶则直接复用，否则创建新的实例。



4.5.2 Activity的启动模式

(3) singleTask模式

singleTask模式下每次启动该Activity时，系统首先会检查栈中是否存在当前Activity实例，如果存在则直接使用，并把当前Activity之上的所有实例全部出栈。

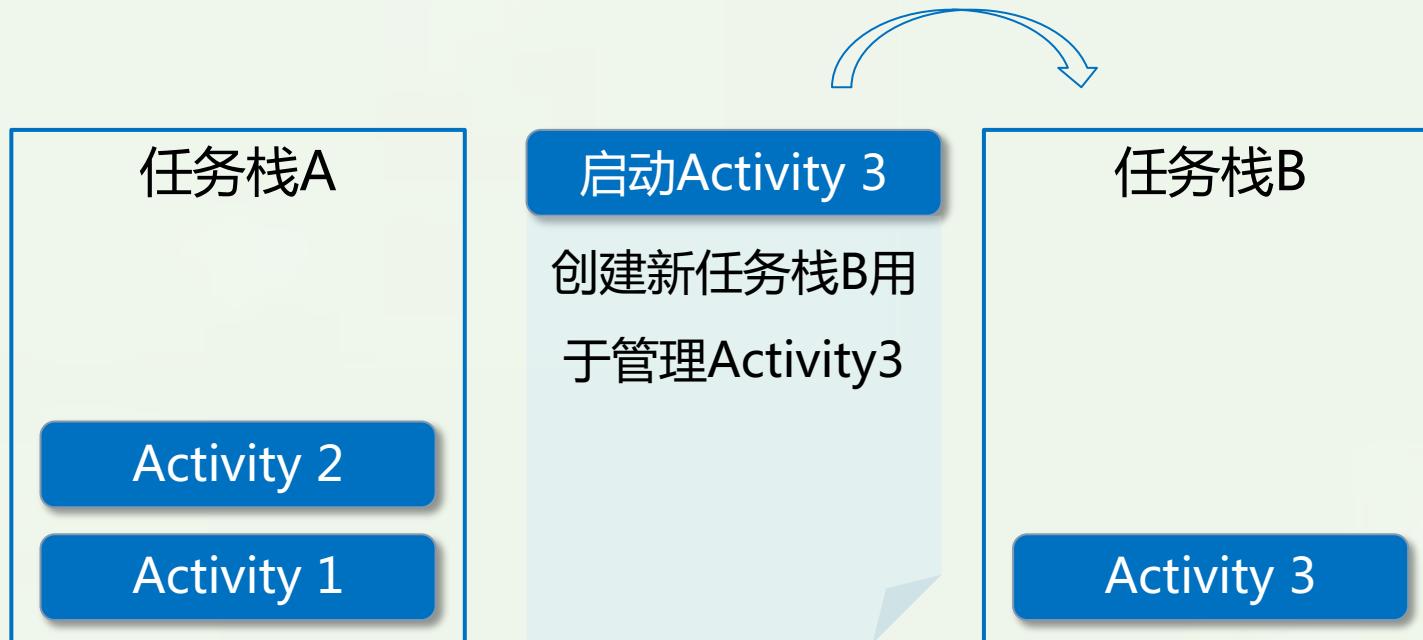
任务栈中有Activity 2，则直接使用，并将Activity 2之上的所有Activity移除



4.5.2 Activity的启动模式

(4) singleInstance模式

singleInstance模式会启动一个新的任务栈来管理Activity实例，无论从哪个任务栈中启动该Activity，该实例在整个系统中只有一个。





》》》 本章小结

本 章 小 结

本章主要介绍了Activity的相关知识，包括了Activity的生命周期，如何创建、启动和关闭单个Activity、Intent和IntentFilter、Activity之间的跳转与数据传递和Activity的启动模式，以及Fragment的使用。在Android程序中用到最多的就是Activity以及Activity之间数据的传递，因此要求读者必须掌握这部分内容。