

# 传智播客

## 《Android 移动应用基础教程》

### 教学设计

课程名称： Android 移动应用基础教程

授课年级： 2015 年级

授课学期： 2015 学年第一学期

教师姓名： 某某老师

2014 年 10 月 09 日

课题名称	第3章 Activity	计划学时	8学时
内容分析	在现实生活中，经常会使用手机进行打电话、发短信、玩游戏等，这就需要与手机界面进行交互。在 Android 系统中，用户与程序的交互是通过 Activity 完成的。同时 Activity 也是 Android 四大组件中最常用的一个，本章将针对 Activity 的相关知识进行详细地讲解。		
教学目标	1、要求学生了解如何创建 Activity，以及 Activity 的生命周期； 2、要求学生了解 Android 中的任务栈，掌握 Activity 的四种启动模式； 3、要求学生掌握显式意图和隐式意图的使用，会使用隐式意图开启系统照相机、浏览器等； 4、要求学生掌握 Android 中的数据传递方式以及如何进行回传数据。		
教学重点	Activity 的生命周期、Activity 四种启动模式、数据传递、回传数据。		
教学难点	数据传递、回传数据		
教学方式	教师采用课堂讲授为主，使用教学 PPT 辅助授课。		

## 第一学时

### (Activity 简介、Activity 的创建)

#### 一、情景导入

##### 1、什么是 Activity

不知道大家有没有想过这样一个问题，每个应用程序都有很多界面组成，这些界面由什么管理的呢？同学进行回答，然后老师引出本节课要讲解的 Activity。

#### 二、知识讲解

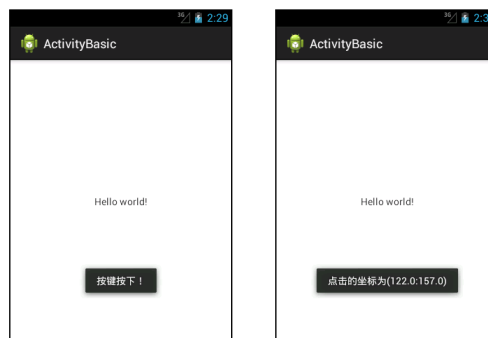
##### 1、Activity 简介 (PPT8-10)

在 Android 系统中，用户与程序的交互是通过 Activity 完成的。Activity 是 Android 应用程序的四大组件之一，它负责管理 Android 应用程序的界面。一个应用程序一般会包含若干个 Activity，每一个 Activity 组件负责一个界面的展现。

下面列举几个 Activity 的常用事件，具体如下：

- `onKeyDown(int keyCode,KeyEvent event)`: 对应按键按下事件
- `onKeyUp(int keyCode,KeyEvent event)`: 对应按键松开事件
- `onTouchEvent(MotionEvent event)`: 对应点击屏幕事件

然后在 MainActivity 中重写上述方法，运行程序进行测试。



##### 2、Activity 的创建 (PPT11)

老师引导，大家对 Activity 有了简单的认识，接下来分步骤演示如何创建 Activity。

- 1) 定义一个类继承自 `android.app.Activity` 或者其子类；
- 2) 在 `res/layout` 目录中创建一个 xml 文件，用于创建 Activity 的布局；
- 3) 在 `AndroidManifest.xml` 文件中注册 Activity；
- 4) 重写 Activity 的 `onCreate()` 方法，并在该方法中使用 `setContentView()` 加载指定的布局文件；

Activity 中的代码如下所示：

```
public class ActivityExample extends Activity {  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
    }  
}
```

```
        setContentView(R.layout.activity_example);  
    }  
}
```



### 三、知识巩固

- 1、总结知识点，使用博学谷系统中的随堂练习题巩固本节课所学知识。
- 2、使用博学谷系统中的测试题给学生布置作业。

## 第二学时

### (Activity 的生命周期、Activity 存活案例)

#### 一、知识回顾

- 1、对上节课留的作业进行答疑。
- 2、回顾总结上节课的内容，引出本节课主题

上节课讲解介绍了 Activity 以及 Activity 的创建，本节将针对 Activity 的生命周期，以及 Activity 存活案例进行讲解。

#### 二、知识讲解

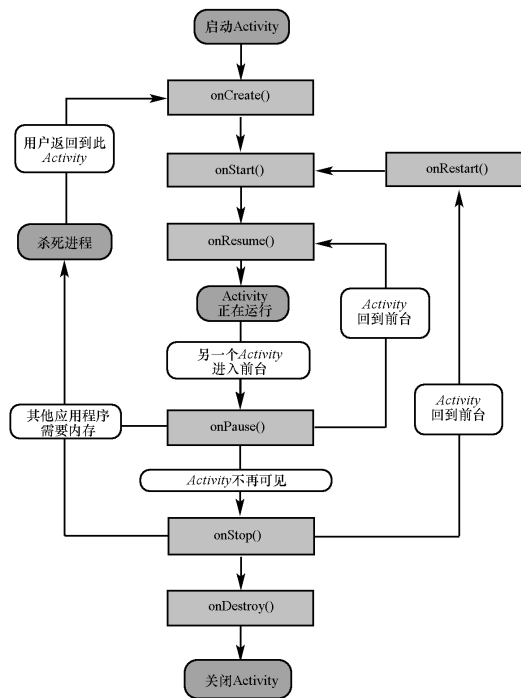
##### 1、Activity 的生命周期 (PPT12-13)

老师引导，程序中的每个对象都有自己的生命周期，也就是从创建到销毁的过程。Activity 也具有自己的生命周期，Activity 的生命周期中分为三种状态，分别是运行状态、暂停状态和停止状态。

- 1、运行状态：当 Activity 在屏幕的最前端时，它是可见的、有焦点的。
- 2、暂停状态：在某些情况下，Activity 对用户来说仍然是可见的，但它不再拥有焦点，即用户对它的操作是没有实际意义的。
- 3、停止状态：当 Activity 完全不可见时，它就处于停止状态，但仍然保留着当前状态和成员信息。然而这些对用户来说都是不可见的。

Activity 从一种状态转变到另一种状态时会触发一些事件，执行一些回调方法来通知状态的变化，具体方法如下所示：

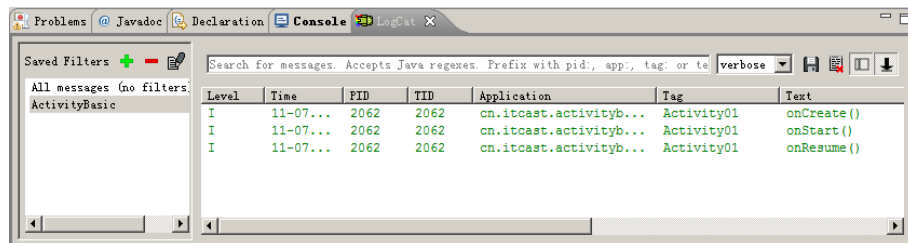
- void onCreate(Bundle savedInstanceState): 创建时执行
- void onStart(): 可见时执行
- void onRestart(): 回到前台，再次可见时执行
- void onResume(): 获取焦点时执行
- void onPause(): 失去焦点时执行
- void onStop(): 用户不可见进入后台时执行
- void onDestroy(): 销毁时执行



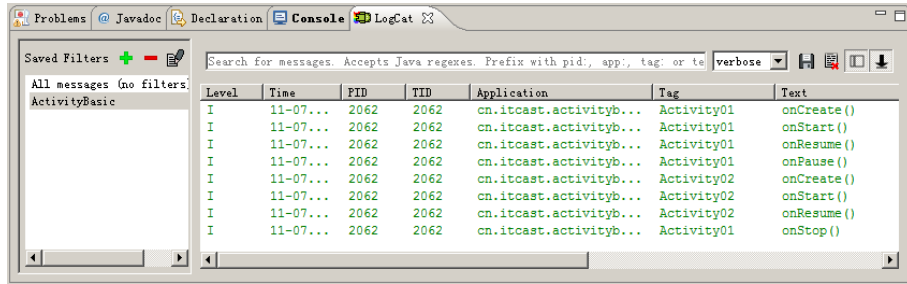
## 2、Activity 存活案例（PPT14-16）

关于 Activity 的生命周期已经讲解完了，想必大家都想来验证一下我讲解的内容是否正确吧！接下来通过案例“Activity 的存活”来演示两个 Activity 之间的跳转时生命周期方法变化的过程。

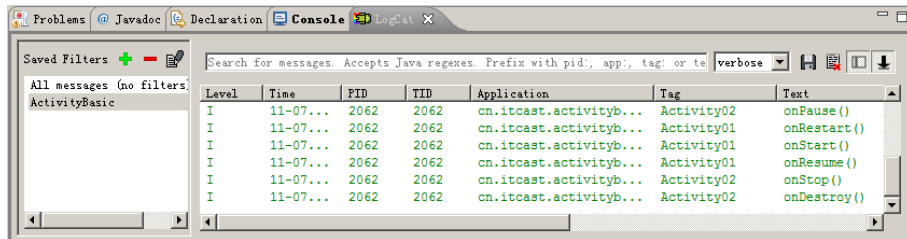
Activity01 创建时执行的方法：



Activity02 创建时，Activity01 执行的方法：



点击 Activity02 返回键时，Activity01 执行的方法：



### 三、知识巩固

- 1、总结知识点，使用博学谷系统中的随堂练习题巩固本节课所学知识。
- 2、使用博学谷系统中的测试题给学生布置作业。

## 第三学时

### （Android 中的任务栈、Activity 的四种启动模式）

#### 一、知识回顾

- 1、对上节课留的作业进行答疑。
- 2、回顾总结上节课的内容，引出本节课主题

上节课讲解了 Activity 的生命周期并通过 Activity 存活的案例进行演示，接下来本节将针对 Android 中的任务栈、Activity 四种启动模式进行讲解。

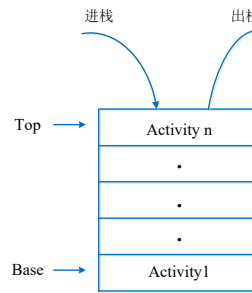
#### 二、知识讲解

##### 1、Android 中的任务栈（PPT17）

老师引导，大家想一下前面讲解 Activity 生命周期时，先后开启了两个 Activity，那么 Android 系统是如何管理这两个 Activity 的呢？

实际上，Android 系统采用任务栈（Task）的方式来管理 Activity 的实例，当启动一个应用时，Android 就会为之创建一个任务栈。先启动的 Activity 压在栈底，后启动的 Activity 放在栈顶，通过启动模式可以控制 Activity 在任务栈中的加载方式。

Android 系统中的任务栈，类似于一个容器，用于管理所有的 Activity 实例。在存放 Activity 时，满足“先进后出（First-In/Last-Out）”的原则。



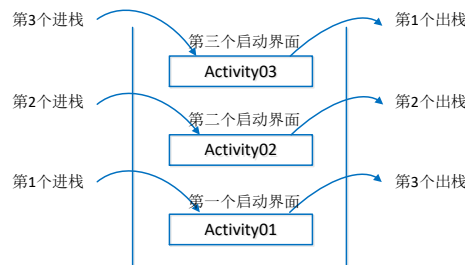
## 2、Activity 的四种启动模式（PPT18-22）

教师引导，前面讲解了 Android 是采用任务中的形式来管理 Activity 的。实际开发中，应根据特定的需求为每个 Activity 指定恰当的启动模式。

Activity 的启动模式有四种，分别是 standard、singleTop、singleTask 和 singleInstance。在 AndroidManifest.xml 中，通过<activity>标签的 android:launchMode 属性可以设置启动模式。

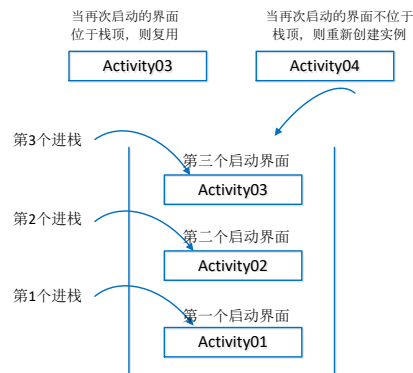
### 1) standard 标准模式

standard 是 Activity 默认的启动模式。在 standard 模式下，每当启动一个新的 Activity，它就会进入任务栈，并处于栈顶的位置，对于使用 standard 模式的 Activity，系统不会判断该 Activity 在栈中是否存在，每次启动都会创建一个新的实例。



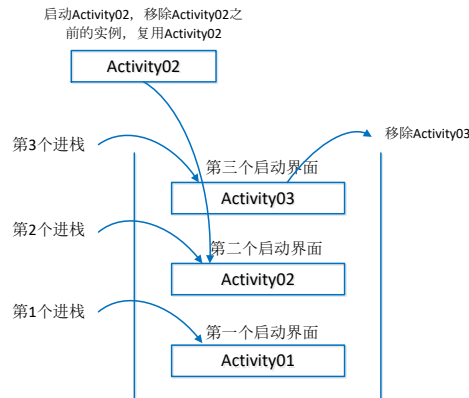
### 2) singleTop 启动模式

singleTop 启动模式与 standard 类似，不同的是，当启动的 Activity 已经位于栈顶时，则直接使用它不创建新的实例。如果启动的 Activity 没有位于栈顶时，则创建一个新的实例位于栈顶。



### 3) singleTask 模式

如果希望 Activity 在整个应用程序中只存在一个实例，可以使用 singleTask 模式，当 Activity 的启动模式指定为 singleTask，每次启动该 Activity 时，系统首先会检查栈中是否存在该活动的实例，如果发现已经存在则直接使用该实例，并将当前 Activity 之上的所有 Activity 出栈，如果没有发现则创建一个新的实例。

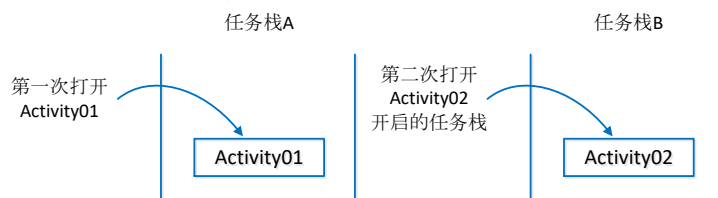


### 4) singleInstance 模式

在程序开发中，如果需要 Activity 在整个系统中都只有一个实例，这时就需要用到 singleInstance 模式，不同于上述三种模式，指定为 singleInstance 模式的 Activity 会启动一个新的任务栈来管理这个 Activity。

singleInstance 模式加载 Activity 时，无论从哪个任务栈中启动该 Activity，只会创建一个 Activity 实例，并且会使用一个全新的任务栈来装载该 Activity 实例。采用这种模式启动 Activity 会分为以下两种情况，具体如下：

第一种：如果要启动的 Activity 不存在，系统会先创建一个新的任务栈，再创建该 Activity 的实例，并把该 Activity 加入栈顶。



第二种：如果要启动的 Activity 已经存在，无论位于哪个应用程序或者哪个任务栈中。系统都会把该 Activity 所在的任务栈转到前台，从而使该 Activity 显示出来。

## 三、知识巩固

- 1、总结知识点，使用博学谷系统中的随堂练习题巩固本节课所学知识。
- 2、使用博学谷系统中的测试题给学生布置作业。

## 第四学时

(Intent 介绍、显式意图和隐式意图、打开系统照相机)



## 一、知识回顾

- 1、对上节课留的作业进行答疑。
- 2、回顾总结上节课的内容，引出本节课主题

上节课讲解了 Android 中的任务栈、Activity 的四种启动模式，接下来将针对程序中的 Intent 介绍、显式意图和隐式意图、打开系统照相机进行讲解。

## 二、知识讲解

### 1、Intent 介绍 (PPT23)

教师引导，很久以前通信技术不发达时，人们只能通过信件的方式互相通信，这中间一定会用到“邮递员”，这个邮递员就起到了传递信息的作用。那么大家思考一下，在 Android 系统中，组件之间怎么进行数据传递呢？

在 Android 系统中，使用的是 Intent 进行数据传递。Intent 中文翻译为“意图”，它是 Android 程序中各个组件进行信息传递的一种重要方式，它不仅可以指定当前组件要执行的动作，还可以在不同组件之间进行数据传递。

Intent 一般用于启动 Activity、启动服务、发送广播等，承担了 Android 应用程序三大核心组件相互间的通信功能。

开启 Activity 的示例代码如下：

```
startActivity (Intent intent)
startActivityForResult (Intent intent)
```

### 2、显式意图和隐式意图 (PPT24-26)

教师引导，Android 中 Intent 寻找目标组件的方式分为两种，一种是显式意图和一种是隐式意图。

#### 1) 显式意图

通过 Intent 启动 Activity 时，需要明确指定激活组件的名称。

```
Intent intent = new Intent(this, Activity02.class);
startActivity(intent);
```

#### 2) 隐式意图

没有明确指定组件名的 Intent 称为隐式意图。Android 系统会根据隐式意图中设置的动作 (action)、类别 (category)、数据 (Uri 和数据类型) 找到最合适的组件。

```
<activity android:name="com.itcast.intent.Activity02">
    <intent-filter>
        <action android:name="cn.itscast.xxx"/>
        <category android:name="android.intent.category.DEFAULT"/>
    </intent-filter>
</activity>
```

使用隐式意图开启 Activity 的示例代码如下所示：

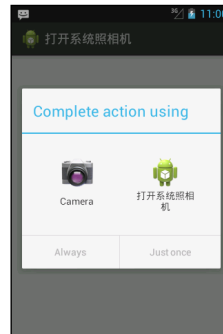
```
Intent intent = new Intent();
// 设置动作和清单文件一样
```

```
intent.setAction("cn.itscast.xxx");  
startActivity(intent);
```

### 3、打开系统照相机案例（PPT27）

老师引导，在实际开发中，避免不了要调用其他应用程序的组件。例如，在开发新浪微博时，需要启动系统的相机功能。

然后通过案例“打开系统照相机”向大家演示如何使用隐式意图。



## 三、知识巩固

- 1、总结知识点，使用博学谷系统中的随堂练习题巩固本节课所学知识。
- 2、使用博学谷系统中的测试题给学生布置作业。

## 第五学时

### （数据传递方式、用户注册案例）

## 一、知识回顾

- 1、对上节课留的作业进行答疑。
- 2、回顾总结上节课的内容，引出本节课主题

上节课讲解了 Intent、显式意图和隐式意图、打开系统照相机案例，接下来将针对数据传递以及用户注册案例进行讲解。

## 二、知识讲解

### 1、数据传递方式（PPT28-29）

在 Android 开发中，经常要在 Activity 之间传递数据。使用 Intent 传递数据只需调用 putExtra()方法将想要存储的数据存在 Intent 中即可。当启动了另一个 Activity 后，再把这些数据从 Intent 中取出即可。

在 Activity01 中将数据存储存在 Intent 中

```
String data = "Hello Activity02"
```

```
Intent intent = new Intent(this,Activity02.class);
intent.putExtra("extra_data",data);
startActivity(intent);
```

从 Activity02 中取出传递过来的数据

```
Intent intent = getIntent();
String data = intent.getStringExtra("extra_data");
Log.i("Activity02", data);
```

还有一种传递数据的方式是调用 `putExtras()`方法传递数据，该方法传递的是 `Bundle` 对象。调用 `putExtras()`方法传递数据可以使用如下代码：

```
Bundle bundle = new Bundle();
bundle.putString("name", "Linda");
bundle.putInt("age", 20);
Intent intent = new Intent(this,Activity02.class);
intent.putExtras(bundle);
startActivity(intent);
```

如果想要在 Activity02 中取出上述方式传递的数据，可以使用如下代码：

```
Intent intent = getIntent();
Bundle bundle = intent.getExtras();
String stuName = bundle.getString("name");
int stuAge = bundle.getString("age");
```

## 2、用户注册案例（PPT30）

老师引导，为了让初学者掌握 Activity 中的数据传递，接下来通过“用户注册”的案例来演示 Activity 中的数据传递。然后按照教材中的步骤一步步编写程序，实现效果如下：



## 三、知识巩固

- 1、总结知识点，使用博学谷系统中的随堂练习题巩固本节课所学知识。
- 2、使用博学谷系统中的测试题给学生布置作业。

第六学时

（回传数据、装备选择案例）

## 一、知识回顾

- 1、对上节课留的作业进行答疑。
- 2、回顾总结上节课的内容，引出本节课主题

上节课讲解了数据传递方式以及用户注册案例，接下来将针对程序中的回传数据以及装备选择案例进行讲解。

## 二、知识讲解

### 1、回传数据（PPT31）

老师引导，大家用过新浪微博 APP 吗？有没有发现在微博发布页面选择图片后，会回到微博发布页面并回传了图片选择页面的图片信息。其实这种效果是通过 Andorid 中的 `startActivityForResult()` 方法实现的。

`startActivityForResult()` 示例代码如下：

```
Intent intent = new Intent(this,Activity02.class);
startActivityForResult(intent,1);
```

在 Activity02 中添加数据返回的示例代码，具体如下所示：

```
Intent intent = new Intent();
intent.putExtra("extra_data","Hello Activity01");
setResult(1,intent);
finish();
```

由于使用了 `startActivityForResult()` 方法启动 Activity02，因此会在 Activity01 页面回调 `onActivityResult()` 方法，需要在 Activity01 中重写该方法来获取返回的数据

```
protected void onActivityResult(int requestCode, int resultCode,
Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (resultCode == 1) {
        String data = data.getStringExtra("extra_data");
        Log.i("Activity01",data);
    }
}
```

### 2、装备选择案例（PPT32）

老师引导，回传数据已经讲解完了，接下来通过一个装备选择的案例进行演示回传数据功能。



### 三、知识巩固

- 1、总结知识点，使用博学谷系统中的随堂练习题巩固本节课所学知识。
- 2、使用博学谷系统中的测试题给学生布置作业。

### 第七学时、第八学时

#### (上机练习)

- 1、完成教材中所有教学案例以及课后习题中的上机练习。
- 2、使用博学谷系统中的测试题给学生布置上机练习，或者老师自行安排上机练习。

思考和练习	见教材第3章课后习题
教 学 后 记	