**02\_四大应用组件之Activity**

🖊 人不光是靠他生来就拥有一切，而是靠他从学习中所得到的一切来造就自己。 —— 歌德

**相关课程资料：**

鼠标放置在正文左侧，点击“**+**”工具栏中的“**高亮块**”来插入高亮块。

|  |
| --- |
| 科目：Android  日期：03/21  主题：02\_四大应用组件之Activity  课前阅读： @ 插入相关文档 |

**Activity的理解**

* Activity，直译为活动，它是Android定义的四大应用组件之一，也是最重要的，也是用的最多的
* 作用：Activity用来提供一个能够让用户操作并与之交互的界面
* 组件的特点：
* 它的类必须实现特定接口或继承特定类
* 需要在配置文件中配置其全类名
* 它的对象不实际通过new来创建的，而是系统自动创建的
* 它的对象具有一定的声明周期，它的类中有对应的声明周期回调方法
* 哪些地方用到了反射？
* 配置文本中配置全类名
* 布局文件定义标签
* 显示意图：Intent（Context context，Class c）

**Intent的理解**

* 意图：信使（Activity，Service，BroadcastReceiver三个组件间通信的信使）
* 分类：
* 【显示】：操作 【当前】 应用自己的组件
* 【隐式】：操作 【其他】 应用自己的组件

**Intent的使用**

* 创建：
* 显示：Intent（Content context，Class activityClass）
* 隐式：Intent（String action） // 与Activity与<intent-filter>的action匹配
* 携带数据
* 额外：putExtra（String key，Xxx value）内部使用map容器保存
* 有特定前缀：setData（Uri data） // tel:12123,smsto:123123
* 读取数据
* 额外：getXxxExtra（String key）
* 有特定前缀： Uri getData（）

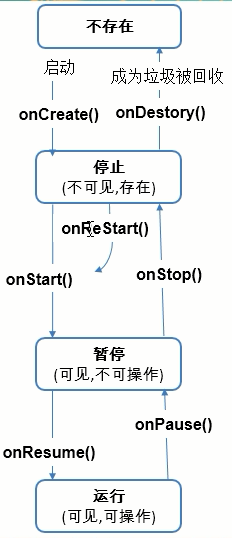
**Activity的使用**

* 定义
* 定义一个类extends Activity，并重写声明周期方法
* 在功能清单文件中使用<activity>注册
* 启动
* 一般： startActivity(Intent intent)
* 带回调启动：
* startActivityForResult（Intent intent，int requestCode）
* 重写：onActivityResult（int requestCode，int resultCode,Intent data）
* 结束
* 一般：finish（）
* 带结果的：setResult（int resultCode，Intent data）

**Activity界面的四种状态**

* 运行状态：可见也可操作
* 暂停状态：可见但不可操作
* 停止状态：不可见，但对象存在
* 死亡状态：对象不存在
* **Activity生命周期与状态**

1. 界面从 死亡 ---》运行
2. 创建对象 ---> onCreate() ---> onStart() --->onResume()
3. 界面从 运行 ---》 死亡
4. onPause() ---> onStop() ---> onDestory()
5. 界面从 运行 ---》 停止
6. onPause() ---> onStop()
7. 界面从 停止 ---》运行
8. onRestart() ---> onStart() ---> onResume()
9. 界面从 运行 ---》暂停
10. onPause()
11. 界面从 暂停 ---》 运行
12. onResume()



* onCreate（）: 加载布局和初始化的工作
* onResume（）: 只有经历次方法，才进入运行状态
* onDestroy（）: 在对象死亡之前，做一些收尾或清理的工作

**TaskStack和lauchMode**

1. TaskStack **任务堆栈**
2. 在Android中，系统用Task Stack（Back Stack）结构来存储启动的Activity对象
3. 一个应用启动，系统就会为其创建一个对应的Task Stack来存储并管理该应用的Activity对象
4. 只有最上面的任务栈的栈顶的Activity才能显示在窗口中
5. lauchMode：启动模式
6. standard：标准模式，每次调用startActivity()方法就会产生一个新的实例
7. singleTop：如果已经有一个实例位于Activity栈的顶部时，就不产生新的实例；如果不位于栈顶，就会产生一个新的实例。
8. singleTask：只有一个实例，默认在当前Task中
9. singleInstance：只有一个实例，创建时会新建一个栈，且此栈中不能有其它的对象

**测试程序**

1. 界面布局
2. 实现Activity的功能
3. 定义所需要操作的视图对象并初始化
4. 给视图设置监听
5. 在回调方法中实现逻辑
6. 实现一般启动
7. 定义好界面二
8. 定义Activity类
9. 配置
10. 重写onCreate(),并加载布局
11. 启动界面二
12. 创建Intent对象（显示）
13. 通过Intent携带额外数据
14. 显示到EditText
15. 实现一般返回
16. 在显示Second界面是，Main界面起始在，知识被盖住了
17. 关闭当前界面

**应用练习：打电话和发短信**



1. 界面布局
2. 分析：垂直的LinearLayout ---》 水平的LinearLayout
3. 编码
4. 在Activity中初始化需要操作的视图对象
5. 给button设置点击监听
6. 给button设置长按监听
7. 点击打电话进入拨号界面
8. 创建一个Intent(隐式)
9. 携带数据
10. startActivity(intent)
11. 长按打电话进入打电话的界面
12. 创建一个Intent（隐式
13. 携带数据
14. startActivity（intent）
15. 点击发短信进入短信编辑界面
16. 创建一个Intent（隐式）
17. 携带数据（号码 / 内容）
18. startActivity（intent）
19. 长按发短信直接发短信
20. 得到SMSManager的对象
21. 发送文本信息（短信）

**总结**

1. 实现一个简单功能的应用的步骤：
2. 外观：分析界面的组成，定义布局文件
3. 行为：编写Activity的实现
4. 在OnCreate（）中加载布局文件：setContentView（layoutId）R.layout.xxx
5. 调用findViewById得到需要操作的所有视图对象并保存成为成员变量
6. 给视图对象设置就监听器（点击/长按），在回调方法实现响应逻辑
7. 使用隐式意图启动系统应用的界面
8. 如何找到对应的Action字符串？
9. 添加ActivityManager的LOG日志，利用系统应用源码找到对应的Activity的配置
10. 权限：
11. 当调用一些系统比较重要的功能的时需要声明

**常见的异常**

1. NollPointException
2. 原因：调用对象的方法 / 属性，但对象为null
3. ClassCastException
4. 原因：类型转换异常
5. ActivityNotFoundException
6. 原因：没有注册Activity
7. 注册不正确
8. 基本常见异常的一般分析步骤：
9. 在logcat中从下往下找，尽量找到causeBy（会显示哪种异常导致的）
10. 找到出异常的类及行号，点击进入对应的行