



《移动终端开发技术》 电子教案

第三单元 Activity 的生命周期

所属专业（教研室）： 计算机软件技术

制定人： 陈媛媛

合作人：

制定时间： 2018年2月



单元标题	Activity 的生命周期	单元教学学时	2 课时
		在整体设计中的位置	第 11 次
授课班级		上课地点	一体化教室
上课时间	周 月 日第 节		
教学目标	能力目标	知识目标	素质目标
	能够熟练使用 Activity 的生命周期中的方法	1、了解 Activity 的生存期 2、掌握 Activity 的生命周期中方法的使用	1、养成积极主动学习意识； 2、养成勤于动手的习惯。
教学重点、难点	<p>教学重点：Activity 的生命周期中方法的使用</p> <p>教学难点：Activity 的生命周期中方法的使用</p>		
教学方法	<p>采用反转课堂教学模式，课前学生学习微课了解知识点，课上采用教师引导、演示，学生分组练习、讨论等教学方法。</p> <p>运用多媒体、AndroidStudio 开发环境、实训助手、教学平台等辅助授课。</p>		
课前需掌握的知识点	<p>void onCreate(Bundle savedInstanceState): 创建时执行</p> <p>void onStart(): 由不可见变为可见时执行</p> <p>void onRestart(): 由停止状态变成运行状态之前调用</p> <p>void onResume(): 准备和用户进行交互的时候调用</p> <p>void onPause(): 失去焦点时执行，系统准备去启动或者恢复另一个活动时调用。</p> <p>void onStop(): 活动完全不可见进入后台时执行</p> <p>void onDestroy(): 活动在销毁之前被调用</p>		
教学任务分解	<p>任务一、理解 Activity 的状态</p> <p>任务二、体验 Activity 的生存期</p>		
教学总结			



一、情景导入

1、引入 Activity 的生存期

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。

2、引出本单元相关学习内容

本节课我们将一起学习认识 Activity 的生存期，并通过案例体验 Activity 的生命周期。

二、课前测试

在网络课程平台上做以下练习：

1、下列哪个不是 Activity 的生命周期方法之一（）

- A. onCreate（） B. startActivity（）
C. onStart（） D. onResume（）

答案 B

2、下列哪个方法是在活动由不可见变为可见的时候调用（）。

- A. onCreate（） B. onPause（）
C. onStart（） D. onResume（）

答案 C

3、完整生存期是活动在 onCreate（）方法和 onDestroy（）方法之间所经历的。（）

答案 正确

三、知识讲解

任务一、理解 Activity 的状态

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。

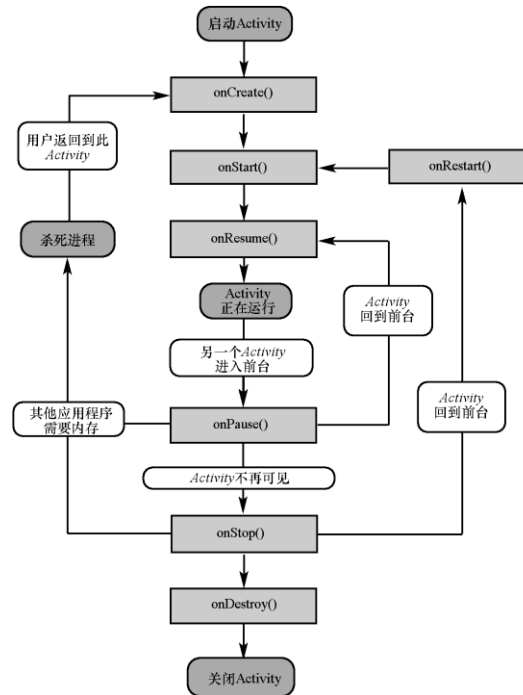
4) 销毁状态：当一个活动从返回栈中移除后就变成了销毁状态。

任务二、体验 Activity 的生存期

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(): 由不可见变为可见时执行
- void onResume(): 由停止状态变成运行状态之前调用

- void onResume(): 准备和用户进行交互的时候调用
- void onPause(): 失去焦点时执行，系统准备去启动或者恢复另一个活动时调用。
- void onStop(): 活动完全不可见进入后台时执行
- void onDestroy(): 活动在销毁之前被调用



案例、体验 Activity 的生命周期

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。

创建项目 AcitivityLifeCycleTest

再创建两个子活动 NormalActivity（普通活动）和 DialogActivity（对话框式的活动）

修改 AndroidManifest.xml 的<activity>标签配置

```
<activity android:name=".DialogAcitivity"
    android:theme="@style/Theme.AppCompat.Dialog">
</activity>
```

在 acitivity_main.xml 中加入两个按钮，一个 start_normalActivity 用于启动 NormalActivity，一个 start_dialogActivity 用于启动 DialogActivity。

最后修改 MainActivity 中的代码

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    Log.d(TAG, "onCreate");
}
```



```
Button sn=(Button) findViewById(R.id.start_normalActivity);
Button sd=(Button) findViewById(R.id.start_dialogActivity);
sn.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View view) {
        Intent intent=new Intent(MainActivity.this,NormalActivity.class);
        startActivity(intent);
    }
});
sd.setOnClickListener(new View.OnClickListener() {

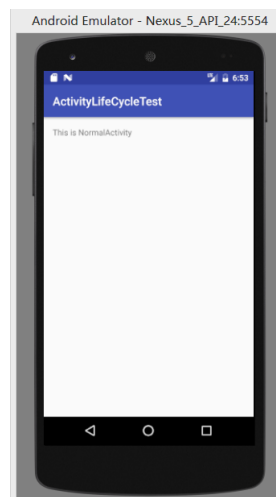
    @Override
    public void onClick(View view) {
        Intent intent=new Intent(MainActivity.this,DialogAcitvity.class);
        startActivity(intent);
    }
});
protected void onStart() {
    super.onStart();
    Log.d(TAG, "onStart");
}
protected void onResume() {
    super.onResume();
    Log.d(TAG, "onResume");
}
protected void onPause() {
    super.onPause();
    Log.d(TAG, "onPause");
}
protected void onStop() {
    super.onStop();
    Log.d(TAG, "onStop");
}
protected void onDestroy() {
    super.onDestroy();
    Log.d(TAG, "onDestroy");
}
protected void onRestart() {
    super.onRestart();
    Log.d(TAG, "onRestart");
}
}
```

运行后效果:



```
Run: AVD: Nexus_5_API_24 app
D/MainActivity: onCreate
D/MainActivity: onStart
D/MainActivity: onResume
I/OpenGLRenderer: Initialized EGL, version 1.4
D/OpenGLRenderer: Swap behavior 1
E/OpenGL_emulation: tid 3316: eglSurfaceAttrib(1146): error 0x3009 (EGL_BAD_MATCH)
W/OpenGLRenderer: Failed to set EGL_SWAP_BEHAVIOR on surface 0x95c6dc80, error=EGL_BAD_MATCH
```

点击第一个按钮

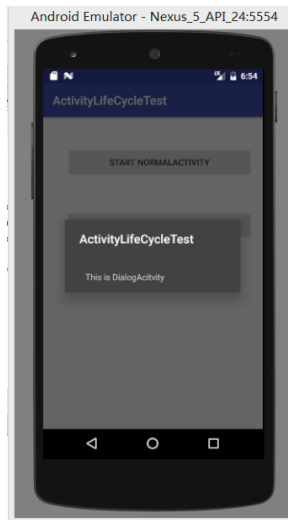


```
Run: AVD: Nexus_5_API_24 app
E/OpenGL_emulation: tid 3316: eglSurfaceAttrib(1146): error 0x3009 (EGL_BAD_MATCH)
W/OpenGLRenderer: Failed to set EGL_SWAP_BEHAVIOR on surface 0x95c6dc80, error=EGL_BAD_MATCH
D/MainActivity: onPause
E/OpenGL_emulation: tid 3316: eglSurfaceAttrib(1146): error 0x3009 (EGL_BAD_MATCH)
W/OpenGLRenderer: Failed to set EGL_SWAP_BEHAVIOR on surface 0x95c11da0, error=EGL_BAD_MATCH
D/OpenGLRenderer: endAllActiveAnimators on 0xa1bcf700 (RippleDrawable) with handle 0x95c142b0
D/MainActivity: onStop
```

点击返回按钮

```
D/MainActivity: onRestart
D/MainActivity: onStart
D/MainActivity: onResume
E/OpenGL_emulation: tid 3316: eglSurfaceAttrib(1146): error 0x3009 (EGL_BAD_MATCH)
W/OpenGLRenderer: Failed to set EGL_SWAP_BEHAVIOR on surface 0x952506a0, error=EGL_BAD_MATCH
```

点击第二个按钮



```
Run: AVD: Nexus_5_API_24 app
D/MainActivity: onPause
E/EGL_emulation: tid 3316: eglSurfaceAttrib(1146): error 0x3009 (EGL_BAD_MATCH)
W/OpenGLRenderer: Failed to set EGL_SWAP_BEHAVIOR on surface 0x95250520, error=EGL_BAD_MATCH
```

点击返回按钮，返回到 MainActivity

```
D/MainActivity: onResume
E/EGL_emulation: tid 3316: eglSurfaceAttrib(1146): error 0x3009 (EGL_BAD_MATCH)
W/OpenGLRenderer: Failed to set EGL_SWAP_BEHAVIOR on surface 0x952506a0, error=EGL_BAD_MATCH
```

返回到 MainActivity 中，点击返回按钮

```
D/MainActivity: onStop
D/MainActivity: onDestroy
```

三、作品展示

各小组展示作品，教师与其他小组评价作品，各小组课后完善作品。

四、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。

拓展
作业

- 1、教学平台中的测试题；
- 2、学习微课《Activity 的启动模式》。

教学
后记