**《移动应用开发》**

**实验报告**

**班级：软件161**

**姓名：胡启明**

**学号：164547**

**实验三 Android生命周期**

**一、实验目的**

1. 巩固使用Eclipse开发Android应用程序方法和步骤。

2．加深了解和体验Activity的生命周期中各状态的变化关系。

3．掌握Android应用程序的调试方法和工具。

**二、实验内容**

1.建立在Android生命周期各阶段添加日志点的Android工程，工程名称：ActivityLifeCycle，包名称：edu.scse.ActivityLifeCycle，Activity名称：ActivityLifeCycle。

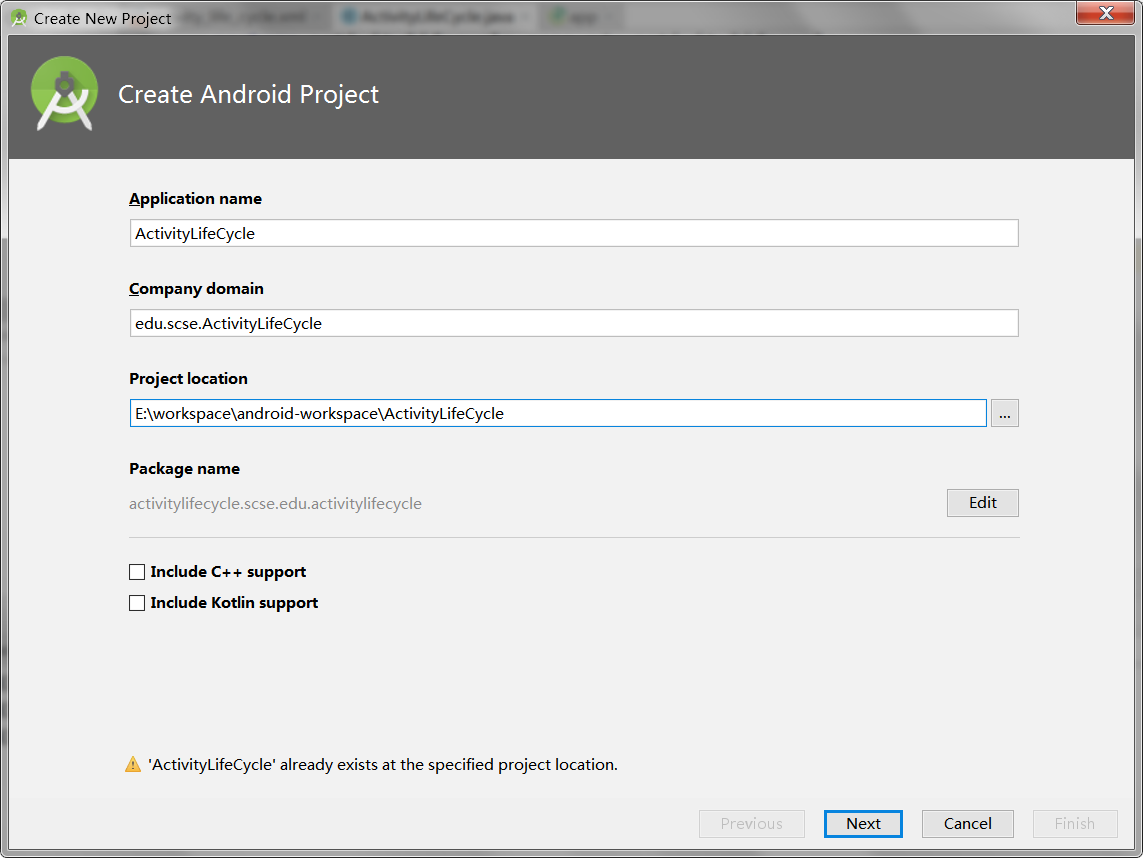
**三、实验仪器、设备**

硬件：PC 微型计算机、1G以上内存，40G以上硬盘

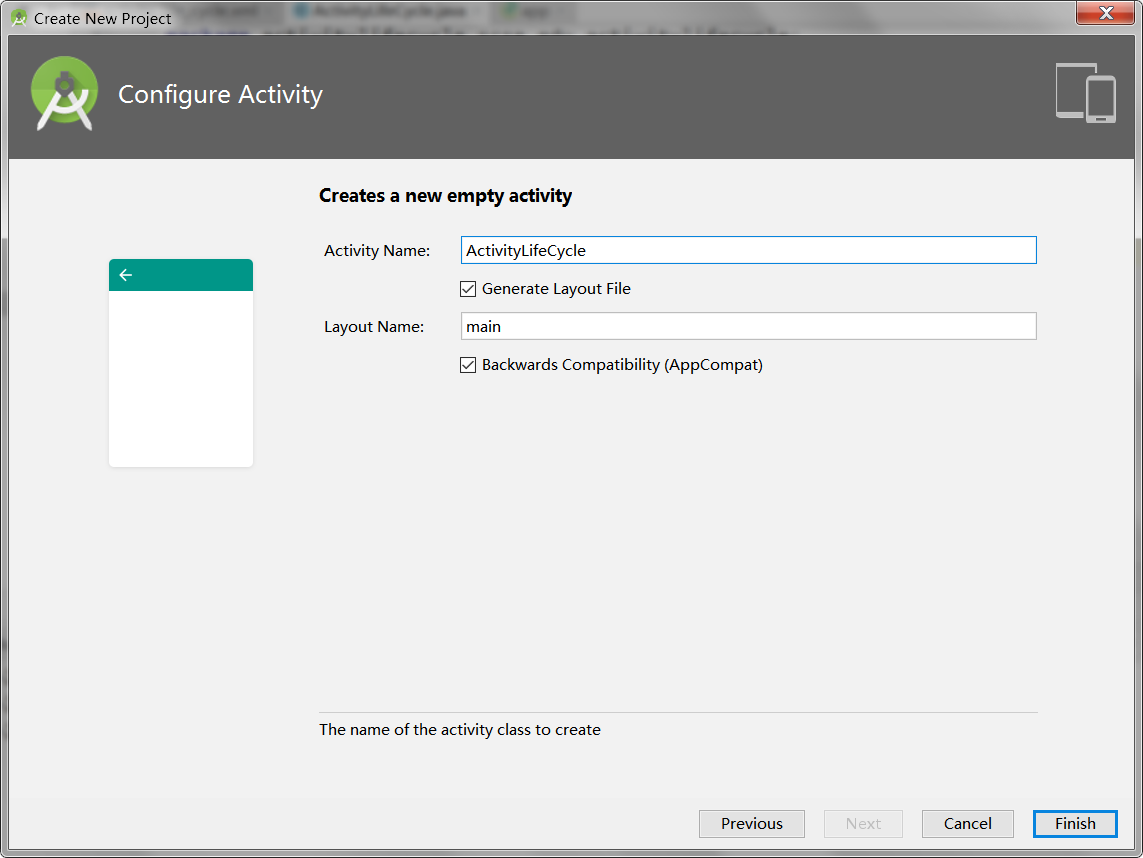
软件：Windows XP，Eclipse , JDK , Android SDK

**四、实验步骤**

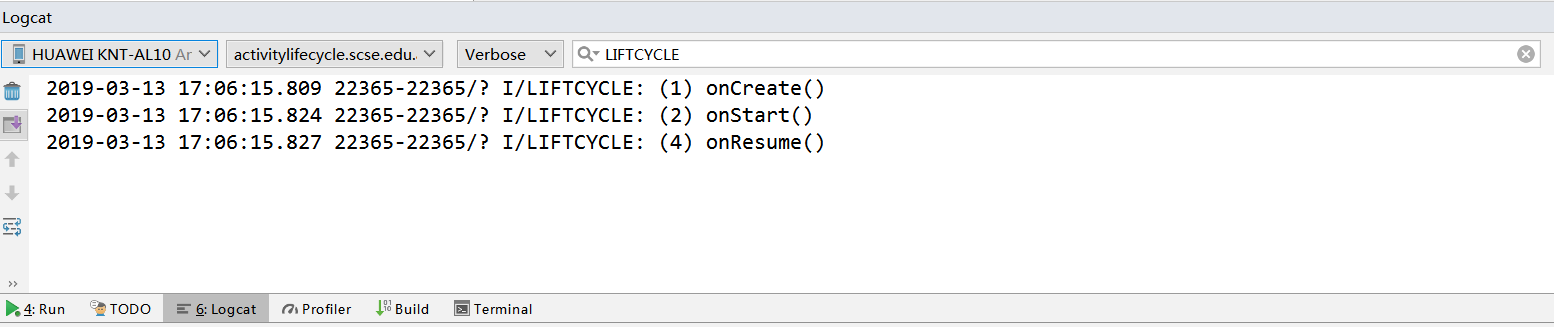
1). 创建名称为ActivityLifeCycle的项目



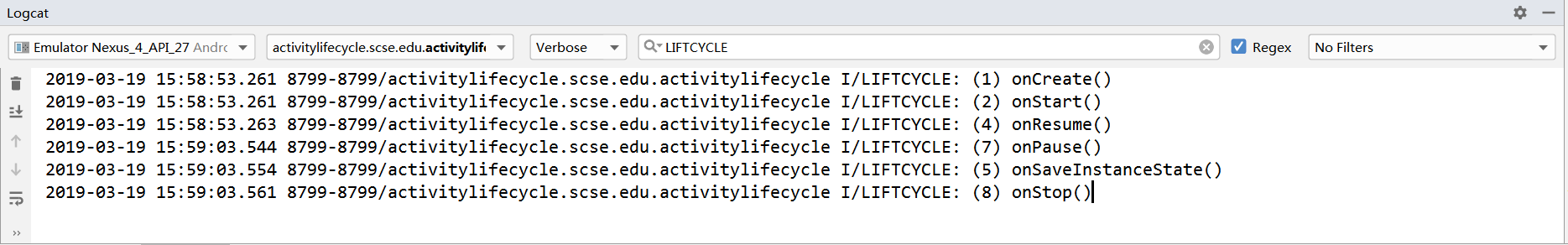
2). 指定Activity的名称和对应的布局文件名称



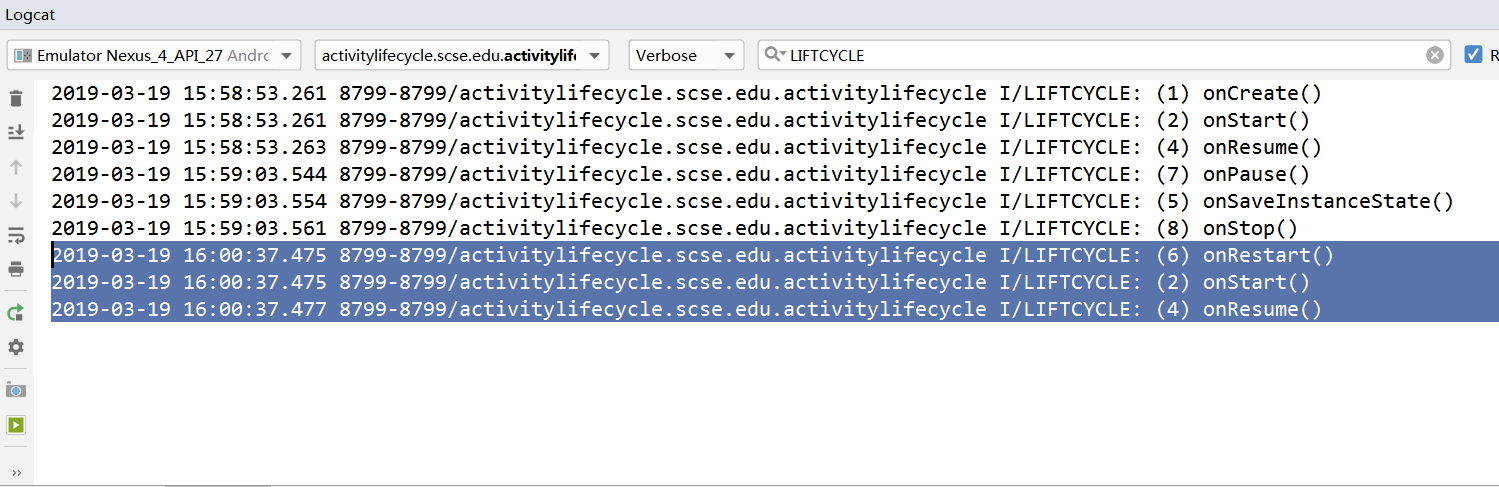
3). 打开应用之后LogCat中的显示



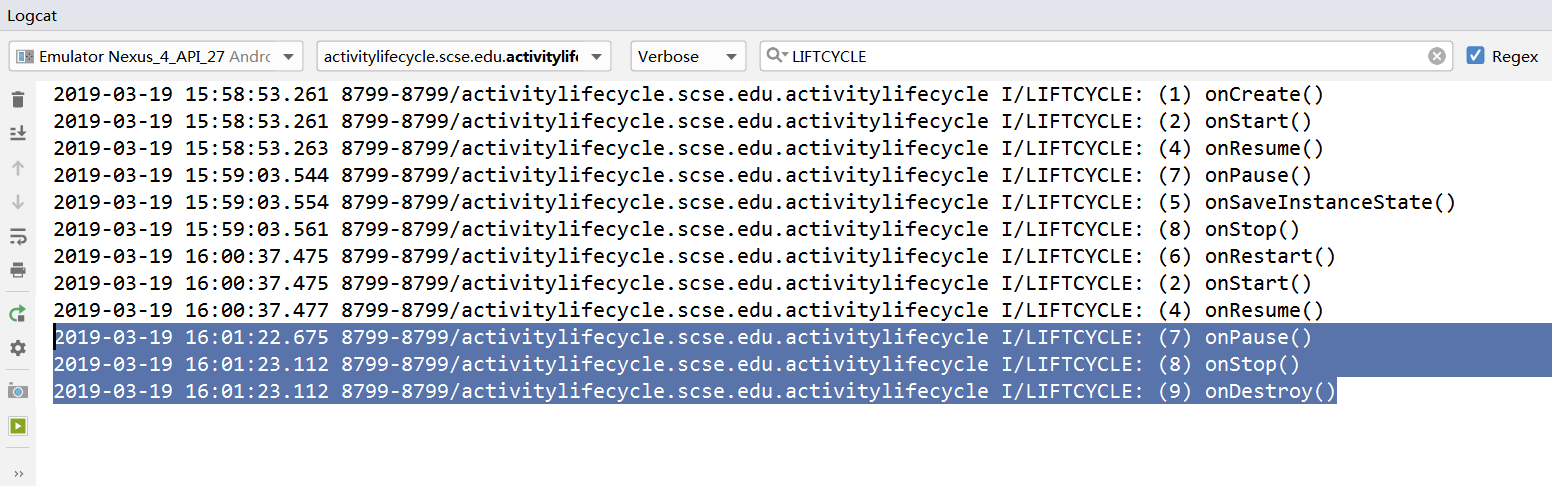
4). 切出到桌面后的LogCat中的显示



5). 切回应用后LogCat中的显示



6). 关闭应用之后的LogCat中的显示



7). 代码实现如下：

ActivityLifeCycle.java

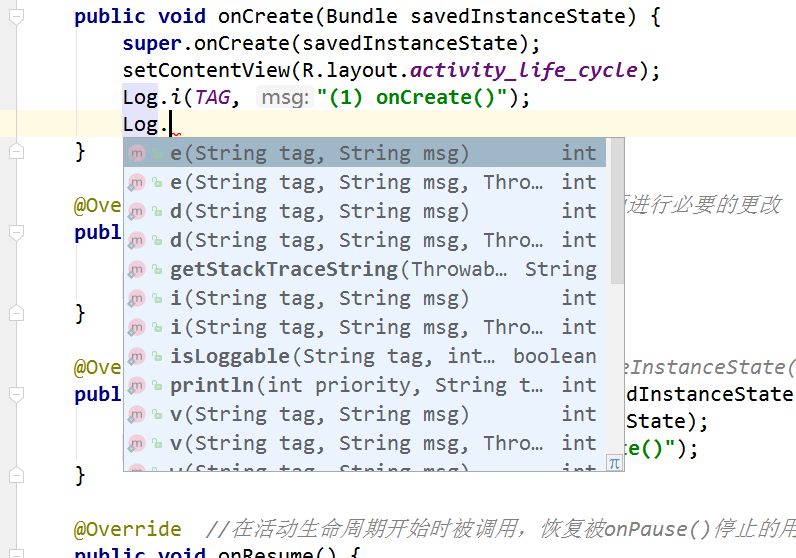
**package** activitylifecycle.scse.edu.activitylifecycle;  
  
**import** android.os.Bundle;  
**import** android.app.Activity;  
**import** android.util.Log;  
  
**public class** ActivityLifeCycle **extends** Activity {  
 **private static** String *TAG* = **"LIFTCYCLE"**;  
  
 @Override *//完全生命周期开始时被调用，初始化Activity* **public void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_life\_cycle***);  
 Log.*i*(*TAG*, **"(1) onCreate()"**);  
 }  
  
 @Override *//可视生命周期开始时被调用，对用户界面进行必要的更改* **public void** onStart() {  
 **super**.onStart();  
 Log.*i*(*TAG*, **"(2) onStart()"**);  
 }  
  
 @Override *//在onStart()后被调用，用于恢复onSaveInstanceState()保存的用户界面信息* **public void** onRestoreInstanceState(Bundle savedInstanceState) {  
 **super**.onRestoreInstanceState(savedInstanceState);  
 Log.*i*(*TAG*, **"(3) onRestoreInstanceState()"**);  
 }  
  
 @Override *//在活动生命周期开始时被调用，恢复被onPause()停止的用于界面更新的资源* **public void** onResume() {  
 **super**.onResume();  
 Log.*i*(*TAG*, **"(4) onResume()"**);  
 }  
  
 @Override *// 在onResume()后被调用，保存界面信息* **public void** onSaveInstanceState(Bundle savedInstanceState) {  
 **super**.onSaveInstanceState(savedInstanceState);  
 Log.*i*(*TAG*, **"(5) onSaveInstanceState()"**);  
 }  
  
 @Override *//在重新进入可视生命周期前被调用，载入界面所需要的更改信息* **public void** onRestart() {  
 **super**.onRestart();  
 Log.*i*(*TAG*, **"(6) onRestart()"**);  
 }  
  
 @Override *//在活动生命周期结束时被调用，用来保存持久的数据或释放占用的资源* **public void** onPause() {  
 **super**.onPause();  
 Log.*i*(*TAG*, **"(7) onPause()"**);  
 }  
  
 @Override *//在可视生命周期结束时被调用，一般用来保存持久的数据或释放占用的资源* **public void** onStop() {  
 **super**.onStop();  
 Log.*i*(*TAG*, **"(8) onStop()"**);  
 }  
  
 @Override *//在完全生命周期结束时被调用，释放资源，包括线程、数据连接等* **public void** onDestroy() {  
 **super**.onDestroy();  
 Log.*i*(*TAG*, **"(9) onDestroy()"**);  
 }  
}

activity\_life\_cycle.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".ActivityLifeCycle"**>  
  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Hello World!"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"** />  
  
</**android.support.constraint.ConstraintLayout**>

**五、实验思考题**

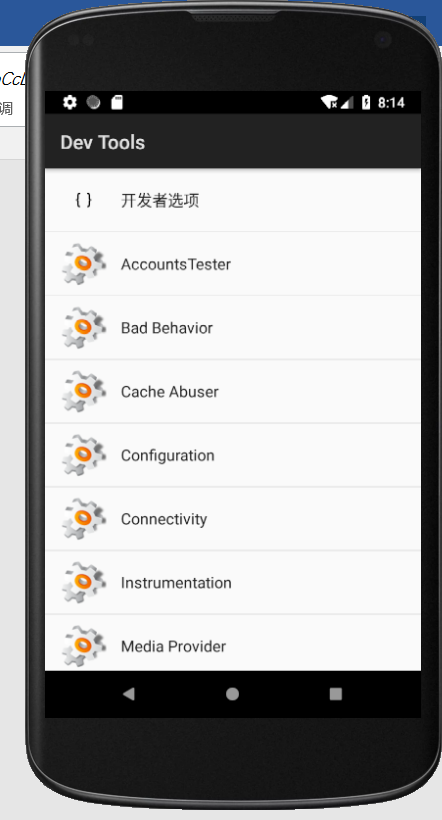
1. LogCat工具如何使用？



如上图所示，在Log类中有各种不同优先级的静态日志打印方法可以调用，从v、d、i、w、e、a级别依次增高。当程序代码执行到这一行时，就会在LogCat调试工具中自动输出相应级别的调试信息。

2．DevTools工具如何使用？

如下图所示，打开模拟器中的Dev Tools应用程序，即可以选择对应的程序功能用于开发者调试。



**六、实验感想**

使用LogCat调试的主要好处就是可以使用过滤器和等级，和直接println在run中输出相比更方便于观察和整理。