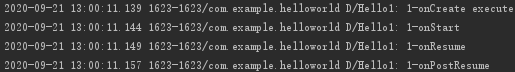
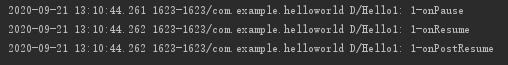
1. **singleTask模式**



Hello1的启动模式是singleTask,且hello1是活动的主界面，运行程序，Hello1创建了一个实例



再点击Hello1按钮，日志结果如下:

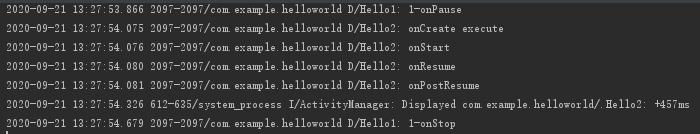


可以看到Hello1先是执行onPause()后又执行onResume()和onPostResume()使自己返回栈的栈顶，并且处于运行状态。可见Hello1活动并未调用onCreate函数，即还是原来的实例。当活动的启动模式指定为singleTask，每次启动该活动时系统首先会在返回栈中检查是否存在该活动的实例，如果发现已经存在则直接使用该实例，并把在这个活动之上的所有活动统统出栈，如果没有发现就会创建一个新的活动实例。

**2.standard模式**

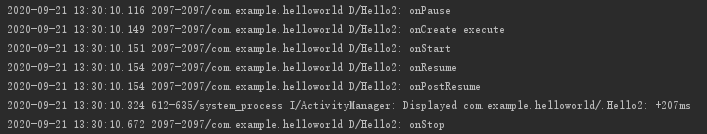


Hello2的启动模式是standard,点击一次Hello2,日志结果如下:



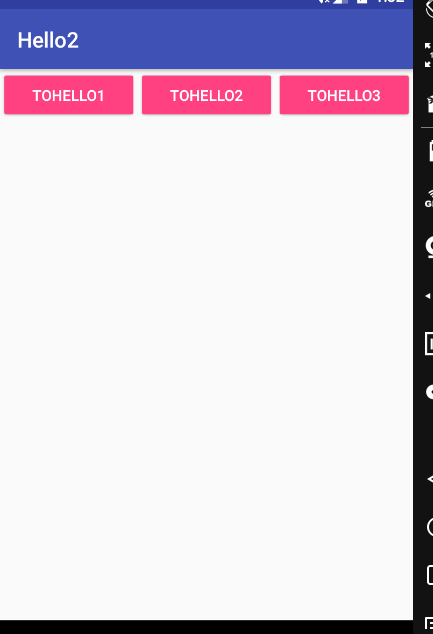
可以看到Hello1进入暂停状态，Hello2执行onCreate()创建了一个实例

再点击一次Hello2，日志结果如下：

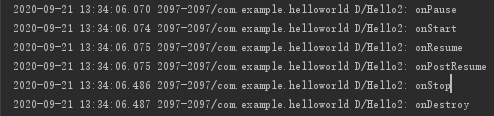


可以看到Hello2执行onPause()后又执行onCreate()创建了一个实例，然后再执行onStart(), onResume(), onPostResume()来显示自己的界面。

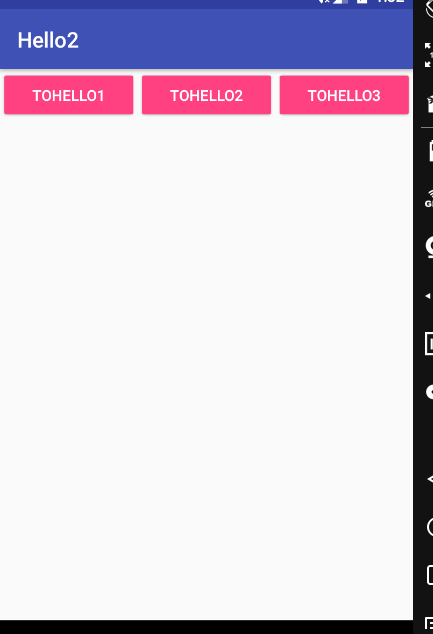
此时界面如下:



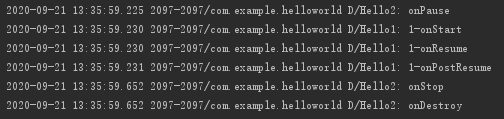
点击back键，日志结果如下：



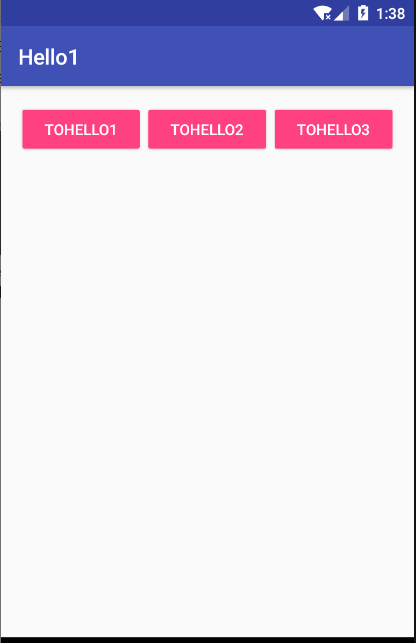
可以看到Hello2执行onPause()后又执行onStart(), onResume(), onPostResume()来显示自己的界面，最后又执行onStop()和onDestroy()销毁实例。此时界面如下:



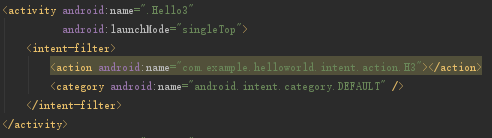
再点击back键，日志结果如下:



可以看到Hello2执行onPause()后，Hello1开始活动，Hello2活动执行onStop()和onDestroy()后被销毁。此时界面如下：

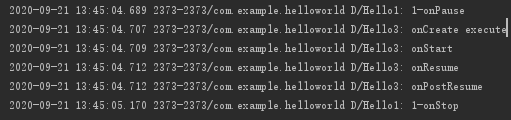


由此可以得出活动的启动模式为standard时，系统不会在乎这个活动是否已经在返回栈中存在，每次启动都会创建该活动的一个新的实例。



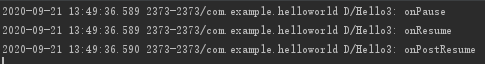
**3.singleTop模式**

Hello3的启动模式是singleTop.点击Hello3，日志结果如下:



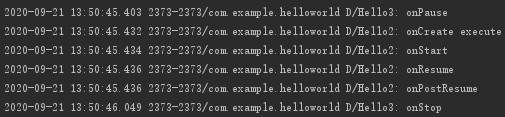
可以看到Hello1进入暂停状态，Hello3执行onCreate()创建实例，后又执行onStart(), onResume(), onPostResume()来显示自己的界面。

点击一次Hello3,日志结果如下:



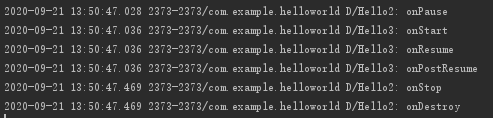
可以看到Hello3先是执行onPause()后又执行onResume()和onPostResume()使自己返回栈的栈顶，并且处于运行状态。可见Hello3活动并未调用onCreate函数，即还是原来的实例。

点击Hello2，日志结果如下:



Hello3进入了停止状态，Hello2执行onCreate()创建实例后又执行onStart(), onResume(), onPostResume()来显示自己的界面。

点击返回键，日志结果如下:

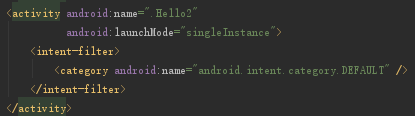


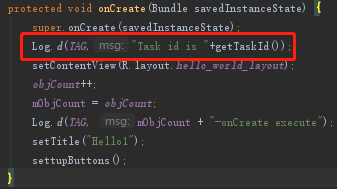
Hello2执行onPause()后，Hello3执行onStart()将活动由不可见变为可见，Hello2活动执行onStop()和onDestroy()后被销毁

由此可知：Hello3的启动方式singleTop处于栈顶时不创建实例对象，但不处于栈顶时便会创建新的实例对象。

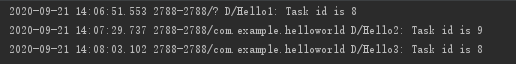
**4.singleInstance模式**

将Hello2的启动模式改为singleInstance,同时修改每个活动中onCreate()方法的代码,打印当前返回栈的id:





重新运行程序，在Hello1界面点击按钮进入到Hello2，然后在Hello2界面点击按钮进入到Hello3,查看logcat中的打印信息



然后按下Back键进行返回，Hello3直接返回到Hello1,再按下Back键又会返回到Hello2,再按下Back键才会退出程序。

由此可知Hello2的启动模式singleInstance，它会返回一个新的返回栈，而Hello3和Hello1共用一个站，Hello3和Hello1依次从返回栈中出栈，此时当前的返回栈已经空了，于是就显示了另一个返回栈的栈顶活动，即Hello2。最后再次按下Back键，这时所有返回栈都已经空了，也就自然退出了程序。