**1. Standard模式（默认）**

我们平时直接创建的Activity都是这种模式的Activity，这种模式的Activity的特点是：只要你创建了Activity实例，一旦激活该Activity，则会向任务栈中加入新创建的实例，退出Activity则会在任务栈中销毁该实例。

**2. SingleTop模式**

这种模式会考虑当前要激活的Activity实例在任务栈中是否正处于栈顶，如果处于栈顶则无需重新创建新的实例，会重用已存在的实例，否则会在任务栈中创建新的实例。

**3. SingleTask模式**

如果任务栈中存在该模式的Activity实例，则把栈中该实例以上的Activity实例全部移除，调用该实例的newInstance()方法重用该Activity，使该实例处於栈顶位置，否则就重新创建一个新的Activity实例。

**4. SingleInstance模式**

当该模式Activity实例在任务栈中创建后，只要该实例还在任务栈中，即只要激活的是该类型的Activity，都会通过调用实例的newInstance()方法重用该Activity，此时使用的都是同一个Activity实例，它都会处于任务栈的栈顶。此模式一般用于加载较慢的，比较耗性能且不需要每次都重新创建的Activity。

一、启动模式介绍

　　启动模式简单地说就是Activity启动时的策略，在[Android](http://www.linuxidc.com/topicnews.aspx?tid=11)Manifest.xml中的标签的android:launchMode属性设置；

　　启动模式有4种，分别为standard、singleTop、singleTask、singleInstance；

讲解启动模式之前，有必要先讲解一下“任务栈”的概念;

　　任务栈

　　每个应用都有一个任务栈，是用来存放Activity的，功能类似于函数调用的栈，先后顺序代表了Activity的出现顺序；比如Activity1-->Activity2-->Activity3,则任务栈为：

二、启动模式

(1)standard：每次激活Activity时(startActivity)，都创建Activity实例，并放入任务栈；

(2)singleTop：如果某个Activity自己激活自己，即任务栈栈顶就是该Activity，则不需要创建，其余情况都要创建Activity实例；

(3)singleTask：如果要激活的那个Activity在任务栈中存在该实例，则不需要创建，只需要把此Activity放入栈顶，并把该Activity以上的Activity实例都pop；

(4)singleInstance：如果应用1的任务栈中创建了MainActivity实例，如果应用2也要激活MainActivity，则不需要创建，两应用共享该Activity实例；

SingTask的应用：

       可以用来退出整个应用。

       将主Activity设为SingTask模式，然后在要退出的Activity中转到主Activity，然后重写主Activity的onNewIntent函数，并在函数中加上一句finish。