要求：Activity的使用(跳转startActivity，有返回结果的跳转startActivityForResult，生命周期，启动模式等)

1、Activity的基本用法
活动是一种可以包含用户界面的组件，主要用于和用户进行交互。最好每一个活动都能对应一个布局，布局就是用来显示界面内容的。
1.1 创建和加载布局
setContentView(R.layout.first_layout);      // 引用布局文件
1.2 注册活动
<activityandroid:name=".FirstActivity"></activity>  //对活动进行注册
android:label="ThisisFirstActivity">  //活动中标题栏的内容（显示在活动顶部）
配置主活动：
<intent-filter>
<actionandroid:name="android.intent.action.MAIN"/>
<categoryandroid:name="android.intent.category.LAUNCHER"/>
</intent-filter>
1.3 菜单的使用
<item      //用来创建具体的某一菜单项
android:id="@+id/add_item"
android:title="Add"/>

让菜单显示出来：
@Override
publicbooleanonCreateOptionsMenu(Menumenu){
getMenuInflater().inflate(R.menu.main,menu);//得到MenuInflater对象，再调用inflate()方法就可以给当前活动创建菜单了
returntrue;//表示允许创建的菜单显示出来，如果false就无法显示
//returnsuper.onCreateOptionsMenu(menu);
}
菜单响应事件：
@Override
public boolean onOptionsItemSelected(MenuItemitem){
switch(item.getItemId()){//判断我们点击的是哪一个菜单项
case R.id.add_item://点击Add
Toast.makeText(this,"YouClickAdd",Toast.LENGTH_SHORT).show();//弹出提示框
break;
case R.id.remove_item://点击remove
Toast.makeText(this,"YouClickRemove",Toast.LENGTH_SHORT).show();//弹出提示框
break;
default:
}
returntrue;
}
1.4 销毁活动
①按一下Back键就可以销毁当前的活动
②finish（）； //点击一下按钮，当前的活动就被成功销毁了

2、使用Intent
作用：是Android程序中各组件进行交互的一种重要方式（主活动跳转到其他活动），不仅可以指明当前组件想要执行的动作，还可以在不同组件之间传递数据。
未命名图片.png 计算机生成了可选文字:
Add
  
作用：是Android程序中各组件进行交互的一种重要方式（主活动跳转到其他活动），不仅可以指明当前组件想要执行的动作，还可以在不同组件之间传递数据。
（用于启动活动、启动服务以及发送广播）
2.1 显式Intent
Intent intent = new Intent(FirstActivity.this,SecondActivity.class);//第一个参数：启动活动的上下文第二个参数：指定想要启动的目标活动
startActivity(intent);//启动目标活动
2.2 隐式Intent
不明确指出要启动哪一个活动，而是指定了一系列更为抽象的action和category等信息，然后交由系统去分析这个Intent，找出合适的活动去启动
例1：调用系统的浏览器来打开网页
Intent intent = new Intent(Intent.ACTION_VIEW);   //Android系统内置的动作
intent.setData(Uri.parse("http://www.baidu.com"));  //setData（）方法将Uri对象传递进去
startActivity(intent);//启动目标活动
<data>标签配置：
未命名图片.png 囗 android:schemea 用 于 指 定 数 据 的 协 议 部 分 ， 如 上 例 中 的 http 部 分 
囗 android ： hosta 用 于 指 定 数 据 的 主 机 名 部 分 ， 如 上 例 中 的 www.baidu.com 部 分 。 
囗 android ： porta 用 于 指 定 数 据 的 端 口 部 分 ， 一 般 紧 随 在 主 机 名 之 后 。 
囗 android ： patho 用 于 指 定 主 机 名 和 端 口 之 后 的 部 分 ， 如 一 段 网 址 中 跟 在 域 名 之 后 的 内 容 。 
囗 android ： m 文 meType 。 用 于 指 定 可 以 处 理 的 数 据 类 型 ， 允 许 使 用 通 配 符 的 方 式 进 行 指 定 。 
例2：在启动活动的时候传递数据
Stringdata="HelloSecondActivity";
Intentintent=newIntent(FirstActivity.this,SecondActivity.class);
intent.putExtra("extra_data",data);
startActivity(intent);//启动目标活动
     >>>>>>>>
Intentintent=getIntent();//获取到用于启动SecondActivity的Intent
Stringdata=intent.getStringExtra("extra_data");//传入键值，得到传递的数据
Log.d("SecondActivity",data);
2.3 startActivityForResult()方法
用于启动活动，同时在活动销毁的时候能够返回一个结果给上一个活动（第一个参数是Intent，第二个参数是请求码，用于在之后的回调中判断数据的来源）

3.活动的生命周期
Android是使用任务（Task）来管理活动的，一个任务就是一组存放在栈里的活动的集合，这个栈也被称为返回栈（Back Stack）——后进先出的数据结构【系统总是会显示处于栈顶的活动给用户】
3.1 活动状态
运行状态
每个活动在其生命周期中最多可能会有4种状态
暂停状态
当一个活动不再处于栈顶位置，但仍然可见时（如对话框形式的活动），这时活动就进入了暂停状态。
停止状态
当一个活动不再处于栈顶位置，并且完全不可见的时候，就进入了停止状态。
销毁状态
        当一个活动位于返回栈的栈顶时，这时活动就处于运行状态。
  
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。
onCreate（），在活动第一次被创建的时候调用（完成活动的初始化操作，如加载布局、绑定事件等）
onStart（），在活动由不可见变为可见的时候调用
onResume（），在活动准备好和用户进行交互的时候调用，此时活动一定位于返回栈的栈顶，并且处于运行状态
onPause（），在系统准备去启动或者恢复另一个活动的时候调用
onStop（），在活动完全不可见的时候调用（如果启动的新活动时一个对话框式的活动，那么执行onPause（）方法，而onStop（）不会执行
onDestroy（），在活动被销毁之前调用，之后活动的状态将变为销毁状态
onRestart（），在活动由停止状态变为运行状态之前调用，也就是活动被重新启动了

完整生存期：onCreate（）：完成各种初始化操作
                    onDestroy（）：完成释放内存的操作
可见生存期：onStart（）：对资源进行加载
                    onStop（）：对资源进行释放       活动对于用户总是可见的！！                  
前台生存期：onResume（）
                    onPause（）                             活动总是处于运行状态！！                        
未命名图片.png 计算机生成了可选文字:
onRestart()
返回上一个活动
返回上一个祜动
onCreate()
onStart()
onResu0
另一个活动来前台
返回上一个活动
另一个优先级更高
的程序需要内存
OnPause()
活动不再可见
on-Stop()
活动被销毁
onDestroy()
图2。23
活动的生命周期
        
3.2活动的生存期
4.活动的启动模式 
  
4.活动的启动模式 
standard（活动默认的启动模式）
每当启动一个新的活动，它就会在返回栈中入栈，并处于栈顶的位置（每次启动都会创建该活动的实例）
AndroidMainfest.xml中通过给<activity>标签指定android：launchMode属性来选择启动模式
未命名图片.png FirstActiVity 
启 动 新 活 动 
Fl.rstActiVlty 
启 动 新 活 动 
FirstActiVity 
返 回 栈 
返 回 
返 回 
singleTop
未命名图片.png 查 栈 训 
甄 是 否 需 要 
启 动 新 活 动 
启 动 新 活 动 
FirstActivity 
返 回 
返 回 
FtrstActwitY 
返 回 栈 
在启动活动时若发现返回栈顶已经是该活动，则认为可以直接使用它，不再创建新的活动实例
singleTask
未命名图片.png 直 接 出 栈 来 重 新 启 动 Fir ctivity 
SecondActwity 
启 动 SecondAcuvity 
FirstActivity 
返 回 栈 
每次启动该活动时首先会在返回栈中检查是否存在该活动的实例，若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；若不存在则会创建一个新的活动实例。
singleInstance
启动一个新的返回栈（在这种模式下会有一个单独的返回栈来管理这个活动，不管是哪个应用程序来访问这个活动，都共用同一个返回栈，也就解决了共享活动实例的问题）
        
  
未命名图片.png ThirdActivity 
返 回 
巧 山 v 呵 
返 回 栈 A 
启 动 新 活 动 
启 动 新 活 动 
返 回 
SecondActmty 
返 回 栈 B 
图 2 ． 41 妯 g n 咖 n “ 模 式 示 意 图 
启动一个新的返回栈（在这种模式下会有一个单独的返回栈来管理这个活动，不管是哪个应用程序来访问这个活动，都共用同一个返回栈，也就解决了共享活动实例的问题）

