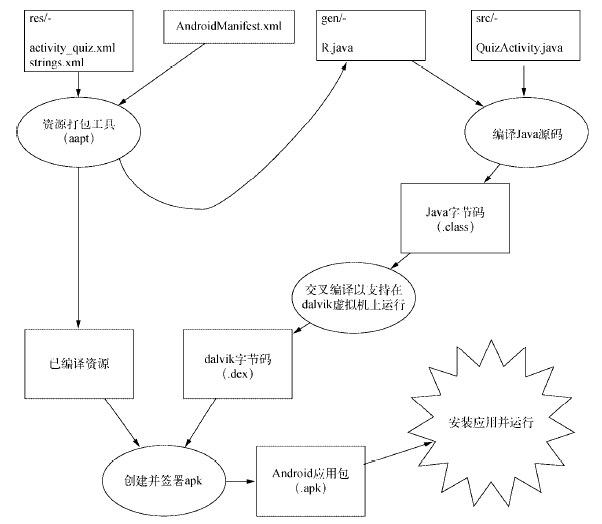
第1章Android应用初体验

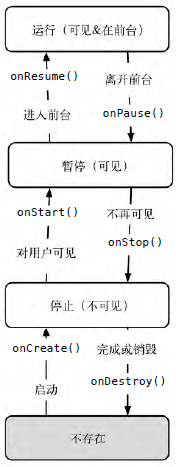
一般通过xml文件定义视图资源，然后使用activity的setContent方法利用资源文件创建视图对象（Builder模式）；



第2章Android与MVC设计模式

aapt会利用资源生成资源id存放在R文件内，通过引用id可以在代码里使用资源，在xml资源文件里也可以通过引用id使用资源，如：android:drawableRight=”@drawable/arrow\_right”

第3章Activity的生命周期



设备配置的变化会导致销毁并重建activity来使用合适的被选资源；

可以通过配置修饰符来区分不同的备选资源；

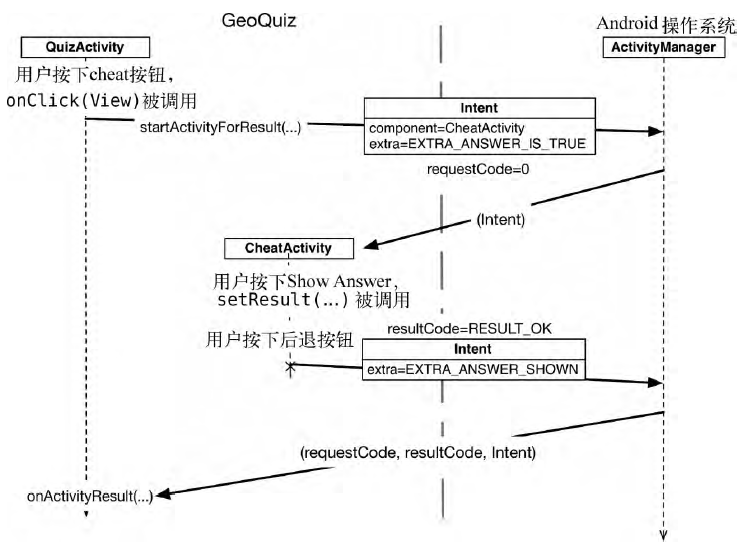
通过使用activity的onSaveInstanceState把状态保存到bundle中，并在onCreate时使用；bundle中保存的只能是基本数据类型和Serializable接口实现；

第4章 Android应用的调试

第5章 第二个Activity

Activity需要在AndroidManifest.xml中配置才能被安卓系统识别和管理；

使用activity的startActivity(Intent)来启动另一个activity；显式Intent(Context, Class<?>)



Activity通过getIntent()来获取传入的Intent；

Activity通过startActivityForResult (Intent, int)来启动activity并使用onActivityResult(int, int, Intent)接收返回；

Activity通过setResult(int, Intent)来返回结果；

ActivityManager维护着一个非特定应用独享的回退栈；

第6章Android SDK版本和兼容性

通过检测SDK版本来安全使用新版本API，并使用@TargetAPI()来关闭lint

第7章 UI fragment与fragment管理器

Fragment把视图从activity中抽离出来，使得视图的根据不同的屏幕有更多的灵活性

Fragment是在11中引入的，以前的版本只能使用支持库来实现



Fragment的生命周期方法又activity来调用

Fragment没有setContent方法，只能在onCreateView中使用inflater来构造view；

Actvity通过FragmentManager来管理Fragment；Fragment生命周期会与activity的生命周期保持同步

第8章使用布局与组件创建用户界面

样式是放在values目录下的xml资源，里面定义了一些样式属性的值；

主题是一些样式资源的集合

dp单位是与像素无关，在每个设备上都一样的大小，一般用于边距，内边距

sp是缩放无关的像素，一般用来设置字体，受用户的字体偏好受影响

第9章 使用ListFragment显示列表



ArrayAdapter是一个iterator模式和享元模式的结合体

RelativeLayout里元素之间通过一些属性来指定元素之间的位置关系；

模型更新后，adapter可通过notify方法触发listview更新

第10章 使用Fragment argument

Intent的putExtra存放的只能是Serializable接口实现；

从一个activity到另一个activity后，新的Fragment一般都是从fragment argument（一个bundle）中获取传进来的参数；activity可通过fragment.setArgument来传入bundle，但这必须在加入Fragment manager之前完成；Fragment可通过getArguments来获得参数

Fragment有startActivityForResult和onActivityResult方法，但没有setResult方法，只能借助activity的setResult来完成返回结果

第11章 使用ViewPaper

在values目录下可以用xml的方式定义一些id供其他视图对象使用，例如当视图对象通过代码生成而又需要拥有id时

视图对象可通过调用构造函数来从代码生成，但没有id