月考简答题范围：每门课程最多两道简答题

《Android电商项目实战》

1. java四种引用类型及区别

**Strong reference     - 强引用**

实际编码中最常见的一种引用类型，也就是new 的形式

**Soft Reference        - 软引用**

软引用所指示的对象什么时候才开始被垃圾回收呢？需要满足如下两个条件：

1.当其指示的对象没有任何强引用对象指向它；

2.当虚拟机内存不足时。

SoftReference变相的延长了其指示对象占据堆内存的时间，直到虚拟机内存不足时垃圾回收器才回收此堆内存空间。

**Weak Reference      - 弱引用**

不改变原有强引用对象的垃圾回收时机

WeakReference来解决Activity中可能存在的Handler内存泄露问题。

**Phantom Reference - 虚引用**

与WeakReference相同，PhantomReference并不会改变其指示对象的垃圾回收时机。

2）接口回调的原理和运用场景

三步 创建接口

实现接口

回调接口

3）BaseActivity、BaseFragment基类抽取的思路

抽取Activity基类.好处呢?第一,可以规范代码...抽基类

4）简述OkHttp对网络请求做了哪些优化；

允许连接到同一个主机地址的所有请求,提高请求效率

共享Socket,减少对服务器的请求次数

通过连接池,减少了请求延迟

缓存响应数据来减少重复的网络请求

减少了对数据流量的消耗

自动处理GZip压缩

5）简述RecyclerView有哪些功能和特性；

默认支持局部刷新。

容易实现添加item、删除item的动画效果。

容易实现拖拽、侧滑删除等功能。

RecyclerView封装了viewholder的回收复用，

提供了一种插拔式的体验，高度的解耦，异常的灵活，针对一个Item的显示RecyclerView专门抽取出了相应的类，来控制Item的显示，使其的扩展性非常强。

但是关于Item的点击和长按事件，需要用户自己去实现。

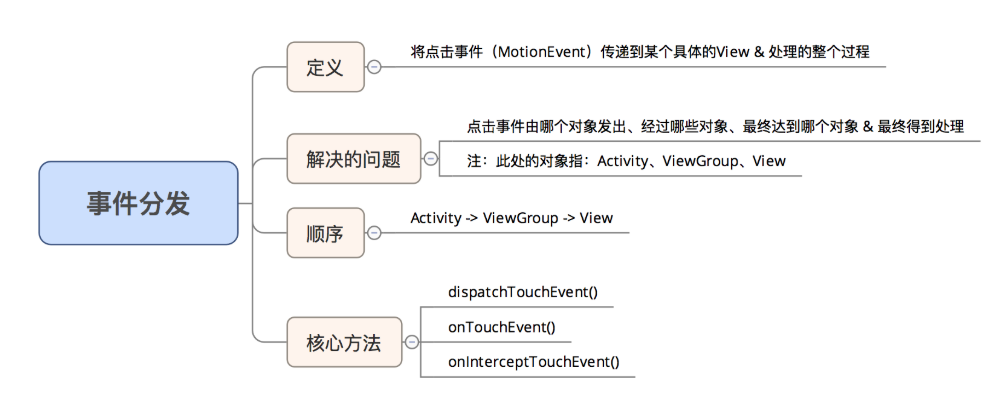
6）详细介绍事件分发流程

事件传递的顺序：Activity -> ViewGroup -> View

**A、dispatchTouchEvent 分发**

**B、onInterceptTouchEvent 拦截**

**C、onTouchEvent 处理时间**



7）简述自定义View的实现思路以及实际的应用场景

1、组合控件

　　组合控件，顾名思义就是将一些小的控件组合起来形成一个新的控件，这些小的控件多是系统自带的控件。比如很多应用中普遍使用的标题栏控件，其实用的就是组合控件

2、自绘控件

　　自绘控件的内容都是自己绘制出来的，在View的onDraw方法中完成绘制。

3、继承控件

　　就是继承已有的控件，创建新控件，保留继承的父控件的特性，并且还可以引入新特性。

8）简述MVP架构的思想以及MVP架构的优化

分离了视图逻辑和业务逻辑，降低了耦合

mvp的全称为Model-View-Presenter，Model提供数据，View负责显示，Controller/Presenter负责逻辑的处理。

在MVP里，Presenter完全把Model和View进行了分离，主要的程序逻辑在Presenter里实现

9）Android中WebView如何与JS进行通信

WebView调用Java方法主要分为三个步骤：

**允许WebView加载JS代码**

**编写JS的接口类，供JS调用**

**给WebView添加JS接口**

//设置兼容JS 这个属性可以实现Android与JS的单向通信  
webSettings.setJavaScriptEnabled(true);

调用HTML的方法,直接加载方法

10）简述Android中的异常捕获机

Android应用不可避免的发生crash,也就是崩溃,除了第三方的收集错误信息外，Android本身也有自己的全局异常捕获。

1、首先需要定义一个类，实现Thread.UncaughtExceptionHandler接口，重写uncaughtException方法，在uncaughtException方法中，

可以把异常存到本地或上传服务端。

2、需要在Application初始化时，初始化CrashHandler