**购物车面试：购物车是一个获取数据，展示商品信息的过程，这里面涉及到网络访问、数据解析、数据展示，ListView优化等等。首先用户浏览商品，你看在这个商品详情页添加产品到购物车时候，访问服务器获取该商品信息显示在listVIew列表，同时原空购物车的图片设为隐藏，购物车数据保存到本地数据库，同时同步到服务器，（如果同步服务器时候网络有问题，就把发送的数据保存到临时文件，然后定时判断网络是否正常，正常就把临时文件上传到服务器，上传成功后，再删除临时文件数据），商品信息的展示主要是在ListView列表上，少不了对ListView进行优化，我用到了**1、重用 ConvertView，对ConvertView判空，当ConvertView不为空时直接重用ConvertView避免创建不必要的View 2、 给 ConvertView 绑定静态类ViewHolder，将convertView的tag设置为ViewHolder，不为空时重新使用缓存3、 分页加载数据，如果有很多条数据的话，客户端不可能傻瓜式的一次性用 ListView 把这些数据全部加载进来，因此我们就用分页加载数据，每次加载10 页**。（分页加载迁移：）**

**我在开发的过程中遇到过这些问题：**

**1、一个就是购物车的某个列表商品点击+的时候，发现是其他的商品数量增加了，也就是要搞清楚点击的按钮是在哪一个列表项的，我给点击按钮设置了一个Tag标记（setTag（i）），然后在点击监听事件中根据tag的position，对相应列表的数据也就是数量、该商品的总价信息进行更改，记得刷新一下；还有就是购物车的数据比较多的时候滑动列表发现图片会错乱，我在网上找到一种解决方式，就是Picasso开源框架，仔细了解后发现Picasso能很好解决在Adapter中加载图片出现的错乱问题，使用很方便，一行代码就实现了图片的异步加载，而且Picasso还支持对图片的压缩转换处理，这可以减少内存消耗，我觉得这在一定程度上避免了OOM异常，毕竟OOM异常出现的多数情况都是因为图片问题，还有Picasso有一个内存和磁盘的二级缓存功能，这个我没有去深入了解（或者讲解ImageLoader框架）；**

**2、还有一个问题就是这个焦点冲突的问题，这个列表项不能点击，不过左边的图片可以点击进去查看商品详情，当时产品经理说这个不要紧，我也就没有管它了，毕竟开发进度有要求，后来我百度了一下，有的说在布局的最外层父容器中加一个属性叫什么Focusability，应该是关于焦点的属性，这样就能解决；**

**3、还有就是全选功能，也就是Checkbox控件，刚开始的时候我简单的遍历一下购物车这个集合，把checkbox都选中，但是上下滑动后发现选中状态被取消了，后来我就**创建一个map集合，这是一个键值对，在adapter初始化的时候put所有item的false，在onItemClick监听的时候记录checkbox的选择状态，然后在getView方法中获取集合的选择的状态，如果点击全选，就把状态全部置为true，否则就是false，大致流程就时这样。（<http://blog.csdn.net/zhangfei_jiayou/article/details/6971986>）

**模块的网络请求我用的是OkHttp网络框架（这个框架现在用的也很多，用法比AsyncHttpClient要麻烦点，但是功能要更强大，像get请求的时候首先构建一个Request对象，传入url强求的地址，不过它还可以通过.build设置其他参数，有了请求对象就可以构造获得Call对象，然后执行它的enqueue方法加入调度队列进行异步网络访问了，在回调接口CallBack里面得到访问的结果，然后就可以进行相关操作了，post请求就是参数里面有请求体，可以添加键值对，其他的和get就差不多了，还有这个框架还支持文件的上传和下载，功能蛮多）。然后访问服务器获取的数据是Json格式，这个就需要进行解析了，我用的是谷歌官方推荐的Gson框架，这个很好用，不像以前用的jsonObjectjsonArray比较麻烦，Gson解析Json数据后转换成一个数据对象，通过这个对象然后就可以进行数据操作或显示了。 然后这个调整商品数量下面的这些总金额啊数字跟随变化我用了观察者模式，观察者观察到数量变化就会计算金额进行UI更新，适配器它自己有一个观察者模式，这个观察类是DataSetObserver，不过这是一个抽象类，需要创建子类，还有就是要记得在生命周期方法中注册数据变化的观察者，DataSetObserver的紫烈中重写onChanged方法，这里具体执行金额变化的操作，适配器记得刷新一下，整个过程还比较简单。对于已经添加大购物车的商品，就说添加前遍历购物车商品集合的id，进行id对比，如果已经添加过，再加添的话就把数量加1就OK。**

**购物车的这个小红点数量提示用到的是一个自定义的TextView控件，主要是重写onDraw方法，进行各种样式设置，还有就是购物车没有商品的时候把这个textview隐藏掉。（小红点：http://blog.csdn.net/swust\_chenpeng/article/details/37505719）。或者说这个小红点提示我用到了开源的第三方控件BadgeView，导入包就可以了，把要显示的控件绑定上BadgeView对象上，进行设置再show出来，这个三方控件还支持设置显示的位置、颜色、背景一些功能，还有动画，用起来很方便（http://www.tuicool.com/articles/am63ue）。**

面试：

一番自我介绍：你好，我是XX，前段时间才从杭州回来，之前一直在搞Android开发，14年过去的，到现在差不多做了两年了，在杭州那边的公司是做外包，我经手的项目还很有几个，市面上主流的app我差不多都做过，像新闻资讯、电商购物的，之前很火的P2P啊，还有与体育教育有关也接手过。我最近做的一个app是电商项目（拿出手机，向面试官展示），你看，叫海悦汇，这个实际上做的是跨境电商项目，现在跨境电商也算是一个电子商务发展趋势，市场比较看好。和我开发的还有一个同事，我主要负责的是框架搭建，首页，购物车这些模块，不过有时候开发进度不一，有些模块里面的内容是同事做的。

项目讲解开始：这个项目在框架整体上是由Fragment+LinearLayout搭建的，整个主界面它是这样的，你看，上面的这部分的根布局是一个FrameLayout，和下面的这个导航栏，你看着下面的5个导航栏是用线性布局做的，使用了weight权重5等分，这个屏幕适配的效果还可以，绑定点击事件就可以点击切换了，这个切换的几个界面就是Fragment了，各自对应不同的界面内容，（延伸：这个切换Fragment是用hide和show实现的，没有用replace。不然每次切换都重新实例化重新加载数据，比较消耗性能和流量），还有这个切换要把Fragment绑定到Activity上，拿到Fragment管理器后开启事务，切换到哪个页面的时候add一下到容器，最后记得提交就可以实现这个切换了，大致流程是这样。其他的就是颜色背景啊一些设置了。其他的还有就是创建了便捷工具类（像dp转换，获取颜色、字符串等），还有统一了接口请求的常量类的配置，对Activity栈也定义了一个类来管理，做了各种全局性的东西。首页：这个首页模块东西还比较多，你看首页模块首先是一个轮播图，这个轮播图是我自己以前做的一个demo直接拿来用的，当然网上啊还有很多刷新的框架，像Pulltorefresh,SwipeListView,，做一个广告推广，然后这个刷新是自己很早的时候做的一个demo，拿来用了，下面是一个GridView的商品分类，点进去就是这一类的商品展示，上面这个商品搜索是同事做的我把它嵌套进来了你看，功能还是没问题的，还有就是这些商品像淘宝一样可以根据销量价格来进行排列，其实就是调不同接口，刷新下就行了，还有一个抽屉效果的侧边栏的筛选，用SliddingMenu做的，当然还有很多这种类似的框架。然后这个商品进入就是商品详情了，这个Viewpager展示商品的几张图，这个布局整体上用了ScrollView，然后可以滑动，下面有四个button，也就是4个功能，不过我觉得加入购物车会麻烦一点，选择加入后你看最上面这个购物车图片会有个小红点显示数量，这个是三方控件**BadgeView做的，刚开始我是自定义一个textview来实现，比较麻烦点，效果也不太好就没有用。再回到首页，下面的新品推荐也是一个viewpgaer搞的，然后接下来这些和上面差不多，最下面是个listView，首页大概就这些了，还有就是这个购物车模块了………**

购物车讲解完后就是结算界面，结算界面可以结合布局说一说Android屏幕适配和布局的优化（这个布局优化适配也可以在讲首页的时候说，首页的布局比较多，结算界面布局相对简单）（适配：屏幕适配方式比较多。 Android 系统本身提供了很多适配方法， 比如存放图片资源的 drawable 目录根据不同分辨率的手机提供了 drawable-hdpi、 drawable-ldpi、 drawable-mdpi、 drawable-xhdpi 等多中目录。我们只需把适应不同分辨率的多套图片分别放到对应的目录中即可。 Android 的 layout、 values 目录也提供了类似 drawable 的适配功能。 但是在开发中，不可能针对不同的手机分辨率提供多种图片资源，这太耗费资源了。我们一般在写控件宽高的时候都会用 dp 单位取代 pix 单位。 因为 dp（像素密度）是一个相对单位， pix 是绝对单位，使用 dp 替 代 pix 也可以解决很多适配问题。 dp 跟 pix （像素）之间可以通过公式进行转换。还有就是代码上的适配，通过getDefaultDisplay()获取屏幕宽高后，对相应的控件进行设置；运用线性布局的时候可以多使用weight权重，注意下根据情况设置宽度或高度为0dp就可以了，以及由于代码中的单位是像素，要用工具类进行转化。 优化：然后优化的话，我写布局的过程中尽量不要把布局层级嵌套太深，有一个<merge>标签可以用，有时候还会用SDK中的一个工具来进行视图分析；布局开发中经常会有重复的代码，我喜欢抽取出来再通过<include>标签复用；对了还有一个就是用标签<ViewStub>去实现View的延迟加载，它不占用位置，比较节省内存，默认情况下是不显示的，特殊情况下下显示，像刷新失败，进度条这些可以用到。

），然后就是提交订单结算了，这个时候的重点就是支付了，我的项目中有三种支付，支付宝、微信、银联，支付（以支付宝支付为例：支付宝这个集成主要根据官方的文档来，里面很详细，并且它还有个demo，大致流程有这样，把开发包导入到工程里面，然后要到manifest里面添加组件声明和权限声明，还有记得代码混淆，之后在代码中生成订单数据和一个支付接口的调用{具体怎么做可以参考demo，我大概记得先要创建个订单，再做个RSA签名，还有URL编码啊接口调用啊这些}，最后就可以获取支付结果进行处理了，这个支付状态码是9000的时候就支付成功了，如果是8000就是支付结果还在确认中，其他值就是失败了，按照文档折腾哈基本上就可以搞定了，还可以参考哈demo。除了支付宝，其他支付根据文档都比较好实现，方式大同小异）讲完后，自己负责的这个项目的基本讲完了，接下来可以问一下公司目前主要做的什么产品，上线过什么产品，给公司产品和公司吹下牛逼（比如：这个产品比较符合当前发展趋势，运用的技术跟上了当前Android技术的发展趋势，比如说点新特性啥的，这样也让面试官感觉自己有不断学习新技术的精神，再说说这款产品的用户应该很多，产品的质量在这里，设计都很好，最好看到公司的app在手机上展示一下….），再问问公司的发展方向，也就是给公司吹牛，公司很有远景，公司未来发展肯定不错，表达出我很希望进入这个公司的意愿，给面试官一种积极向上的态度…..

微信支付： http://dkylin.com/archives/2016/android-wechat-pay-tutorial.html

前提：微信开放平台申请支付

微信支付AppID

微信支付依赖包

一个activity类——**WXPayEntryActivity.java**

1. 导入jar包；
2. 配置manifest；组件和权限声明；
3. 在子线程中创建统一订单，然后获取参数prepay，之后就可以在pay方法发起支付了，最后处理支付结果。