# 怎么阅读

我是先**看gradle文件，用到了哪些库，如何打包**，有个大概的了解，然后运行起来。跑起来以后，去代码中看**launch activity**，这算是整个应用的入口，**从入口顺藤摸瓜，找到每个自己感兴趣的界面的代码**。中间可能会穿插了解一些**基类**，**service**等，但是整体思路是**先找到自己想看的代码**。

大概了解了这些以后，就可以深入点，看看项目的**具体架构**。后面的事情就是开始上手**改bug，填坑，挖坑**了。

楼主做 Android，看公司的 Activity 基类，网络框架，图片加载框架，以及各种 UI 基类、组件。这些是你上手干活需要掌握的。

用业务逻辑一条一条串起来，从页面到后台接口   
然后再看看公共封装的方法   
架构上的设计   
这是我一般看公司代码的思路，希望对你有所帮助

我习惯从activity入手，联系界面，一个view一个view看，不知道别人怎么做的

了解ANDROID基础架构、配合API和注释看。   
运行+断点调试+LOG日志  
某些命名奇葩的代码只能靠经验和苦读了

额，有的童鞋肯定会说，**直接写一个BaseActivity**，让所有的Activity都继承自这个BaseActivity，再在这个BaseActivity里面打印子类的名字呀，你去，你去写吧，几百个，你慢慢写，不送哈。

写到这里，不得不说，我亲爱的开发同仁们，你们写的时候，拜托你们都写一个BaseActivity吧，以表你们的专业，我代表万千接手别人代码的苦逼程序猿谢过。

先看懂整体结构，然后画类图整理代码逻辑，多调试

先了解项目用到的框架 ssh 还是 ssm 或者是公司自己开发的框架，先把框架大概弄懂。如果是 web 项目，先测试环境跑通，了解业务，然后就根据项目去调试了解代码。

然后，在工程文件当中考它的manifest.xml了解有哪些Activity View Service，

再看每一个activity对应的.java 就可以简单看懂了

同新人今年刚毕业，来公司之后先培训了一下业务流程。现在组长给了一个早期版本的程序让我们先试着跑一跑调试一下，同时让我们自己分析比较 springmvc 和 struts2，hibernate 和 mybatis 等，大概一周交一次周报。现在在学习 ssm

# 方法二

<http://www.cnblogs.com/liushilin/p/5737126.html> adb打印

# 方法3

1、使用这个应用或者游戏，直到熟悉它的使用流程和功能  
        2、根据自己的体验分析代码实现（整体）  
        3、如果有条件和原作者请教项目思路  
        4、从最先启动的Activity开始，弄懂每个Activity功能如何实现以及它的布局实现  
        5、遇到不懂的代码，首先找搜索引擎（一般都能解决）。还不行就可以请教他人了  
        6、对新学到的知识点，一定要总结下来，并且定期的去回顾  
        7、最重要的一点：我认为要自己动手写，那怕是重新写一遍这个项目，或者自己设计一个类似的，再或者只写这个项目中的一部分。  
        为什么要自己动手呢？大家都知道coder这个东西，那怕你理论学得再好，不动手，那也是毫无意义的！所以，动手写一下，你会遇到问题，你就会解决问题，这样，你才能学到更多。这就是经验·······

# 优点

# 缺点