IDEA/eclipse
这是两个IDE，主要用于java开发，其中eclipse是个开源免费的框架，有各种大神和公司为其开发了不同功能大插件，例如CDT插件，可以写c/c++代码；svn，用于版本控制。IDEA是JetBrain公司的java开发工具，有付费和社区（免费）版。
eclipse通过adt插件可以开发Android程序，adt即google为eclipse开发的插件；开发人员可以通过eclipse安装adt插件调用Android SDK，实现Android应用的创建、开发和打包；
IDEA也可以开发Android应用，同样需要调用Android SDK，用IDEA开发Android并非google官方（也可能是官方，没有调研过），而是JetBrain公司根据google提供的Android SDK开发的
如上两款IDE并非自己本省可以开发Android应用，而是必须通过Android SDK；同时，这两款IDE开发java应用时也需要对应的jdk，这也是为什么要自己单独安装jdk的原因；

Android Studio
Android Studio是google在IDEA基础上结合gradle生成的新的Android开发工具；IDEA如上所述是个IDE，gradle是自动化构建工具；
能够自动化构建就能够手动构建，我们可以通过Android SDK 和NDK提供的命令来编译和打包Android应用，但是很不方便，所有有了eclipse 的ADT插件，IDEA的Android开发功能等；gradle更加强大，可以实现打不同包名，不用渠道等包，省去人工修改的部分；
注：自动构建工具相当于脚本工具，可以实现对文件的修改、添加和删除，并且调用sdk提供的工具来打包生成应用。例如我们想要对一个同一个应用打包两个不同渠道的包，分别上传给应用汇和豌豆荚，区分这两个渠道的信息在AndroidManifest.xml中，名为channel的meta data；如果手动打包，我们需要修改channel值为yyh，然后使用eclipse、idea，Android studio，甚至直接使用Android SDK的打包命令生成正式安装包，然后将channel的是修改为wdj并进行同样的操作生成另外一个包。使用gradle我们就可以省去这写繁琐且容易出错的手工操作，直接使用gradle脚本一键打包。

PyCharm
JetBrain公司的python IDE；让python开发想java一样有提醒功能，但也需要配置python 环境才能运行和开发；

Android SDK/NDK
Android SDK是Android开发的根本，有google提供；
SDK全称Software Development Kit即软件开发工具包，是个工具包必然就提供了开发过程中不同层面上的工具；一个安卓应用从源代码到成品apk需要经过如下流程：

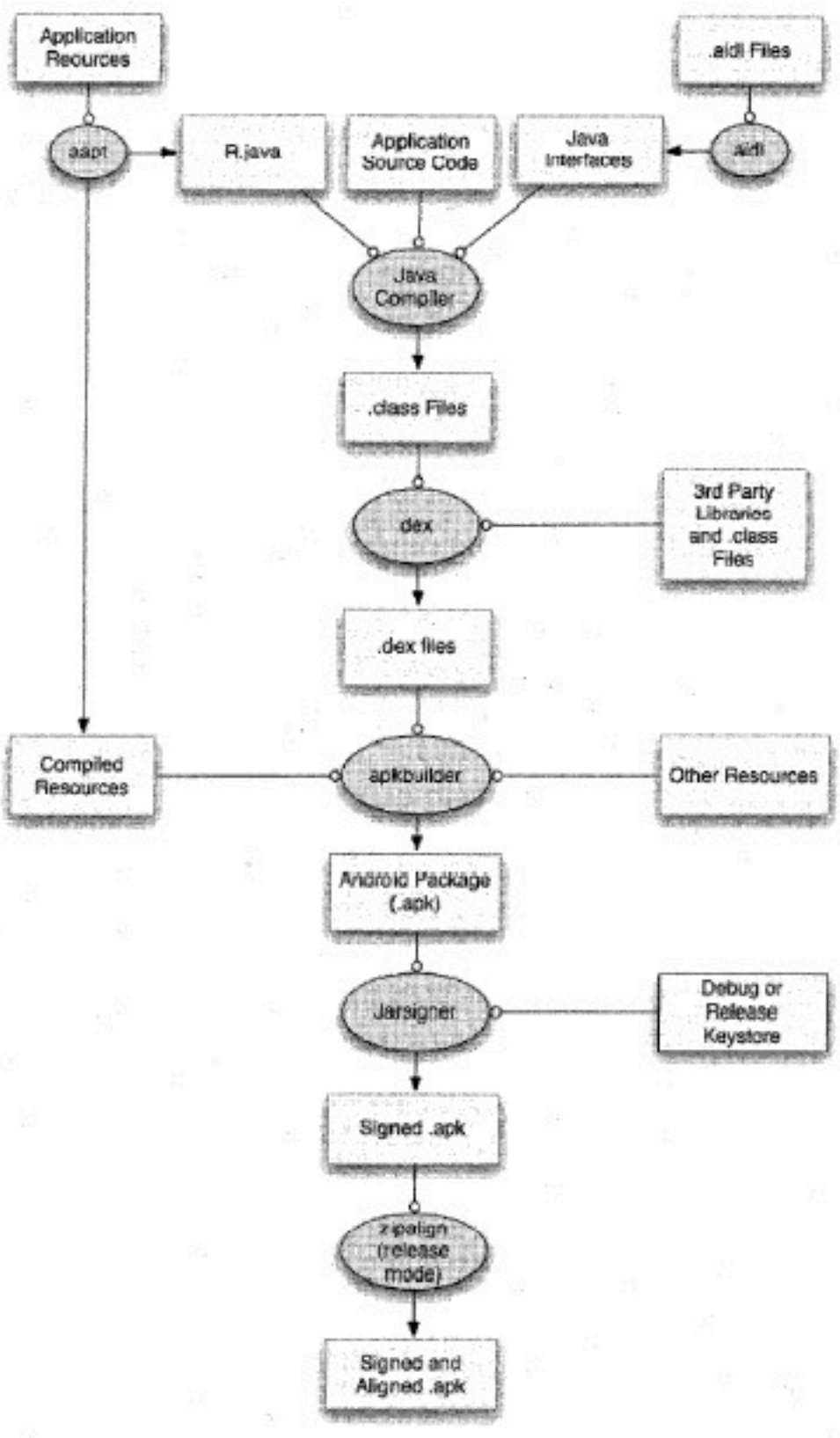

图4-1 APK的打包过程

详细参考非虫大大《Android软件安全与逆向分析》，这里重点介绍一些有用的工具；

-adb，全称Android Debug Bridge（安卓调试桥），起到调试作用，使用adb我们能够将pc和手机连接起来，在pc端对手机进行操作，比如安装卸载应用，导入文件，提出文件等；
adb最主要的功能是桥的作用，转发我们在pc端的命令给手机，然后在等待手机端的反馈；例如：adb shell ls，我们在pc端执输入该命令，adb将我们的ls命令转发到手机端去实行，得到返回结果并在pc的命令行中显示；

-zipalign，用于对签名后的apk进行对齐操作，据说能够使应用更流畅运行（没有研究过）
-更多工具可以看非虫大大的书；

Android NDK全称Native Development Kit（Android 原生开发工具包），用来开发native代码，这里的native是指c/c++开发；我们知道java是跨平台的，一次编译到处运行，因此同一个jar包可以在windows，Linux，mac os平台正常运行，这是因为sun公司为我们在这些平台上开发了一个叫JVM的东西，JVM有自己的指令集，jar包作为可执行文件，其指令是符合JVM指令集的，因此可以实现在所有安装了jvm的系统上运行；c/c++是针对本地环境的，在windows上编译出来的程序只能在windows平台运行，因此称为Native；
Android系统是在Linux的基础上搭建了一个Dalvik虚拟机，执行smali指令集，其可执行文件为dex文件即我们使用java代码开发出来的最终被编译生成的文件；一方面JVM本身支持在java代码中调用native平台的库文件进行执行，另一方面一些处理对执行效率要求比较高，确实需要Dalvik能够支持native库，这就导致Android NDK的产生；
Android NDK的主要功能只有一个交叉编译；所谓交叉编译就是在pc环境上生成手机环境中的运行库；例如pc平台可以使windows平台，在windows平台上需要编译出Android支持的native库，这些库有需要根据手机不同的cpu来具体分类（arm，x86等）；
同时一个开放工具包除了编译，还包括动态调试工具，这就是gdb，Android NDK为我们提供了针对Android平台的动态调试工具gdb，具体位置详查NDK目录吧（gdb工具的gdb server针对不同cpu有不同的版本）；
对于反编译，Android NDK中有arm-xxx-readelf，arm-xxx-objdump等分别针对不同cpu型号的对应工具。