类加载机制原理

动脑学院Android部

hook技术 实现底层java方法转发

art虚拟机加载类 的 流程、方法

java 代码如何在虚拟机内运行？

hook 注入 热修复方案

classloader 加载类

含有main（）方法 -- 可执行类，入口类

Android dex –> PathClassLoader

zygote 安卓Framework 入口

Android App入口类 ActivityThread.main()

### Java基本内存分区

方法区 堆内存 java栈 PC寄存器栈 本地方法栈

成员变量表

类信息等

主动加载

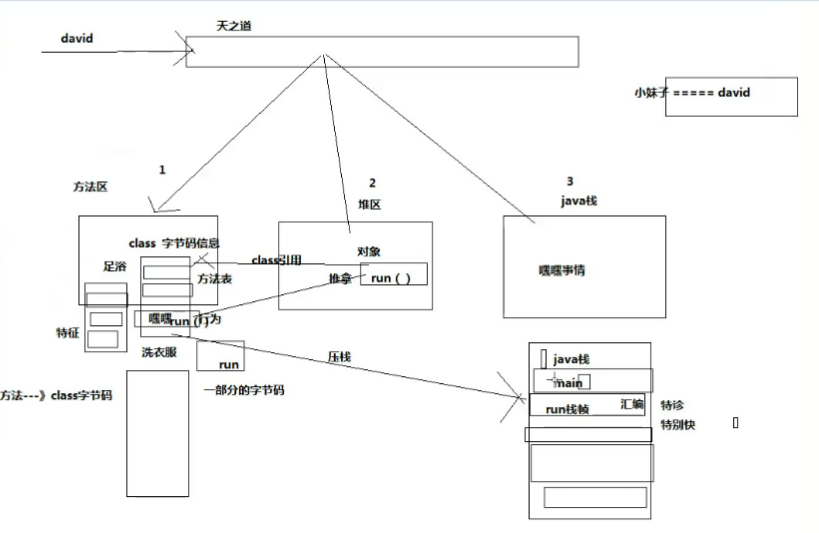
class.forname() ---- 反射

classloader.load();

static 不会引入类信息，

声明 也不会

在**调用动态对象实例时**加载类



只能用C方法替换java对象

【java中的安全机制不允许】降维打击

理想很丰满，现实很骨感

热修复技术

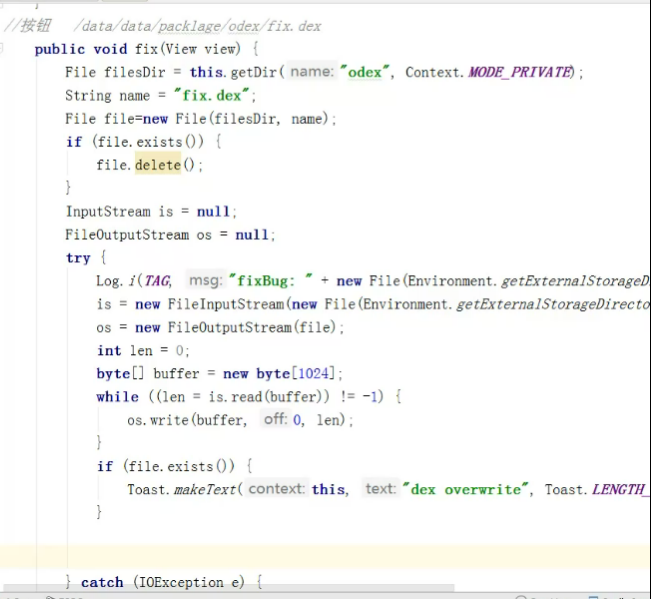
偷偷修复 【hook】

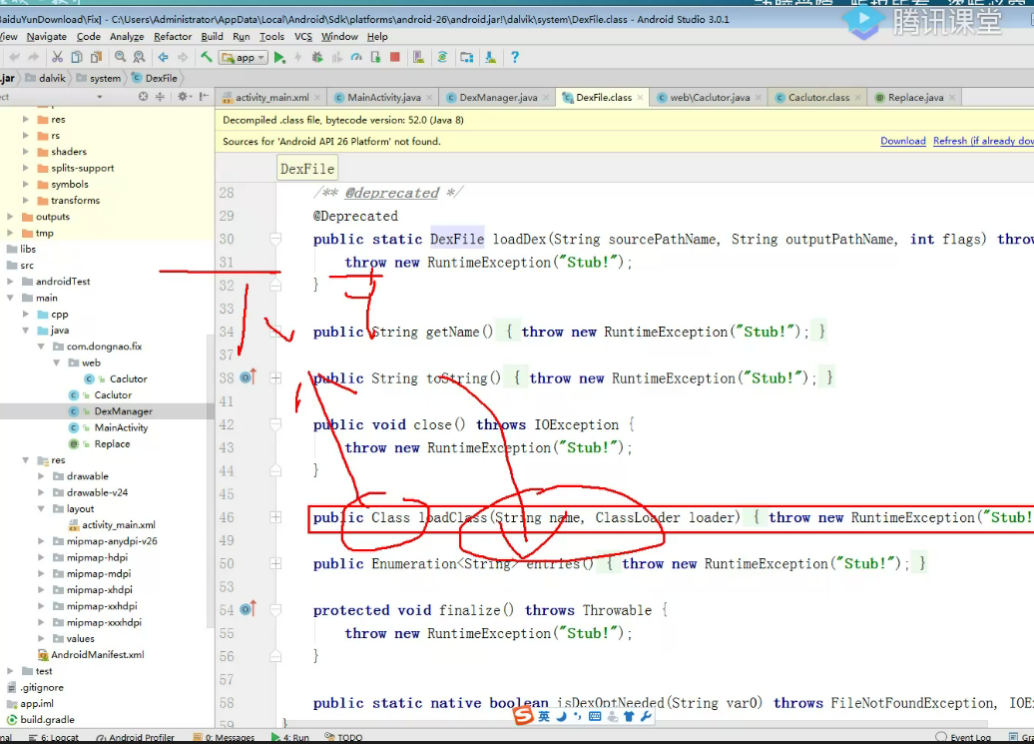
本来会出bug的代码【抛异常的】

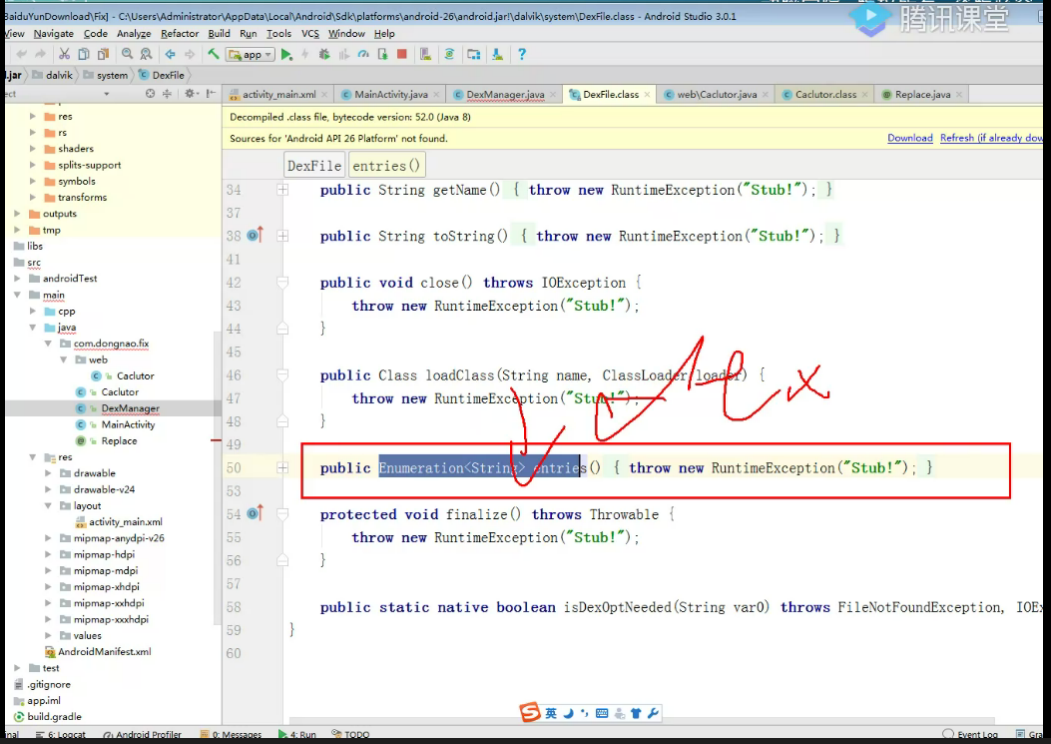
### 打包dex

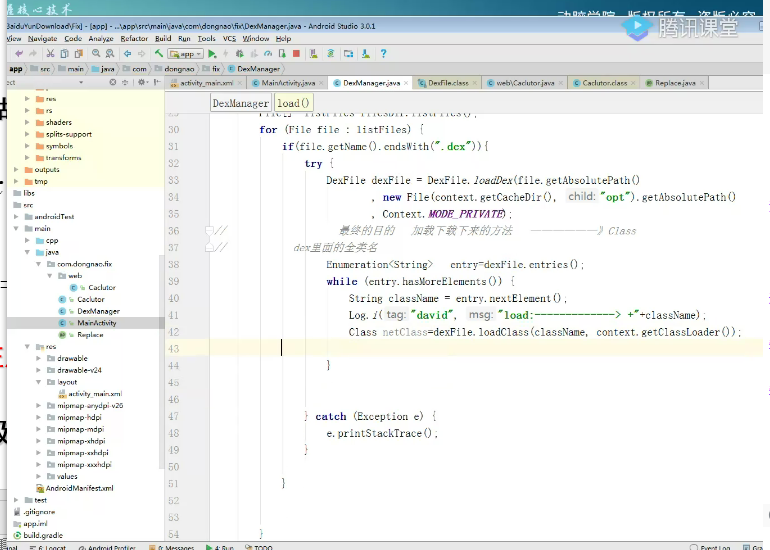
使用build-tools

dx –-dex –output=<out\_dir> <.class>目录









DexFile 类 读取dex的方法。

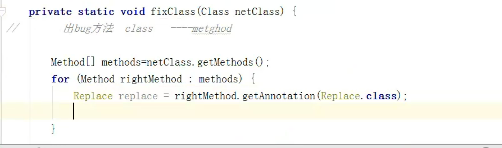
可以获取entries();

Dexfile.loadClass();

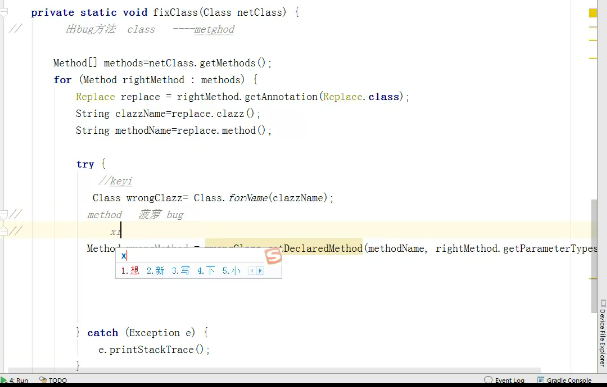
### 拿到java方法表中方法的地址

注解@



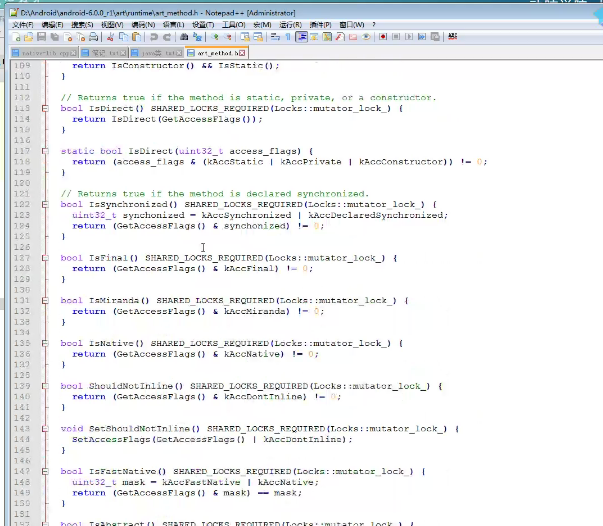


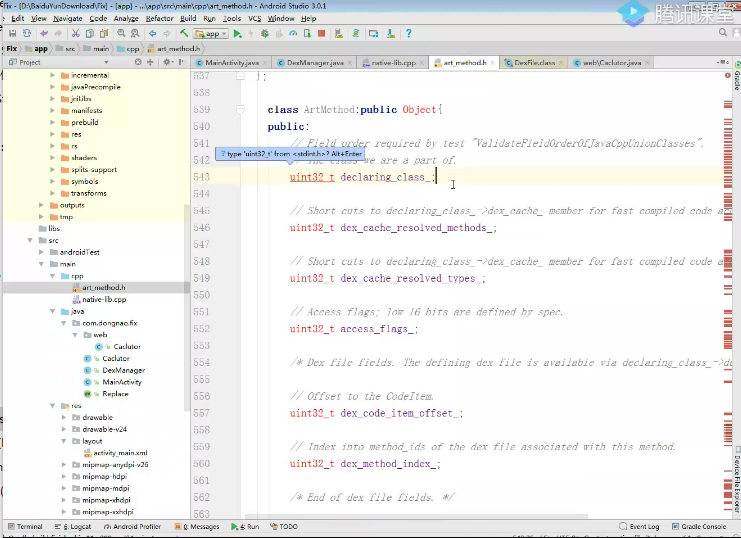
得到方法签名



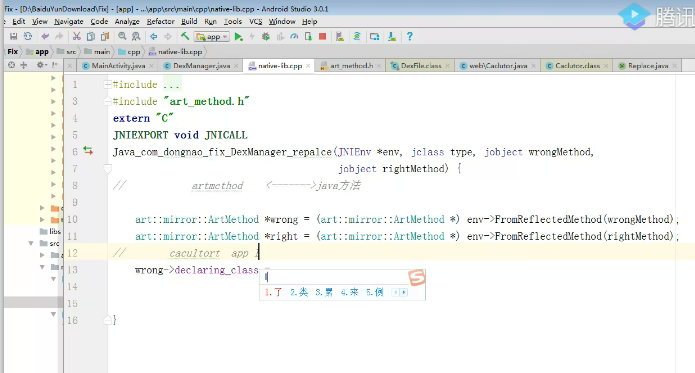
替换：在native层进行替换

1. 找到artMethod









实际替换的：类信息 方法信息

不能替换的：类加载器

不是实际地址替换或者对象替换



