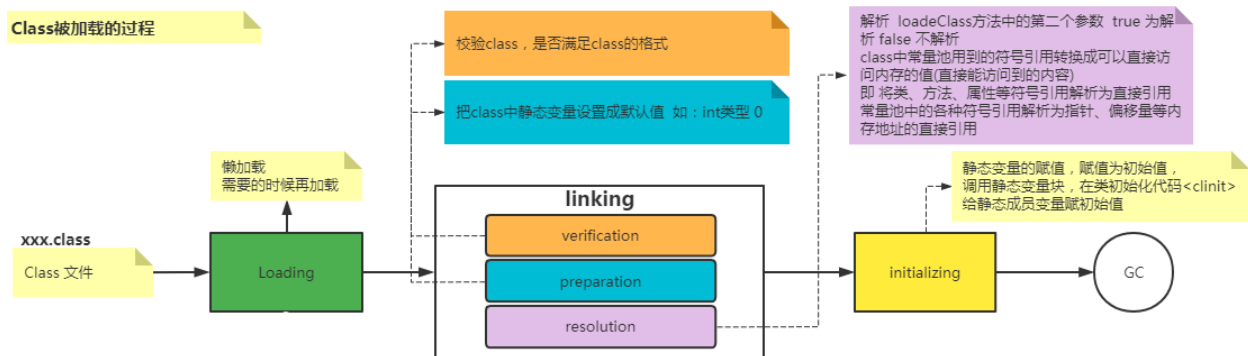


# Class Loading Linking Initializing

## Class 文件是怎样被放在内存的



## 硬盘中的Class被加载的过程

### 1、loading

把一个class 加载到内存

懒加载, 需要的时候再加载

Loading (双亲委派)

ClassLoader原理和自定义类加载器

Class加密/解密 (了解)

Lazyloading懒加载(了解)

### 2、linking

#### (1)、verification

校验class, 是否满足class的格式

#### (2)、preparation

把class中静态变量设置成默认值 int类型 0

### (3)、resolution

解析 loadClass方法中的第二个参数 true 为解析 false 不解析

class中常量池用到的符号引用转换成可以直接访问内存的值（直接能访问到的内容）

即将类、方法、属性等符号引用解析为直接引用

常量池中的各种符号引用解析为指针、偏移量等内存地址的直接引用

### 3、initializing

静态变量的赋值，赋值为初始值，调用静态变量块

在类初始化代码<clinit>,给静态成员变量赋初始值

Class: load 默认值 初始化

new: 申请内存 默认值 初始化

### Java语言执行模式

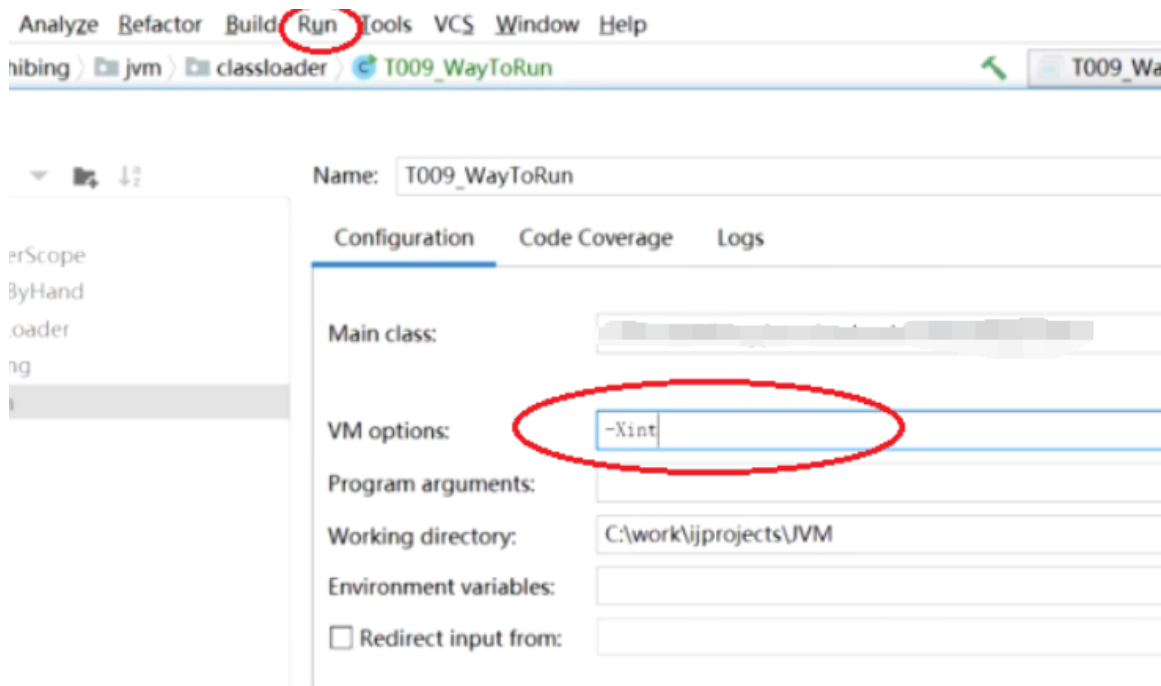
JIT: 有些代码需要编译成本地代码

热点编译: jvm在执行的时候发现有一块代码经常被解释本地代码 提升效率

# 混合模式

- 解释器
    - bytecode interpreter
  - JIT
    - Just In-Time compiler
  - 混合模式
    - 混合使用解释器 + 热点代码编译
    - 起始阶段采用解释执行
    - 热点代码检测
- 96011 多次被调用的方法（方法计数器：监测方法执行频率）
- 多次被调用的循环（循环计数器：检测循环执行频率）
  - 进行编译
- `com.mashibing.jvm.classloader.T009`

## 指定运行模式



默认是混合模式

不同模式下执行时间不同