

「第64期」 - 「JVM-02」 - 「JVM架构剖析」

IparhanGeekIparhanGeekYesterday



IparhanGeek



听着音乐 补充知识

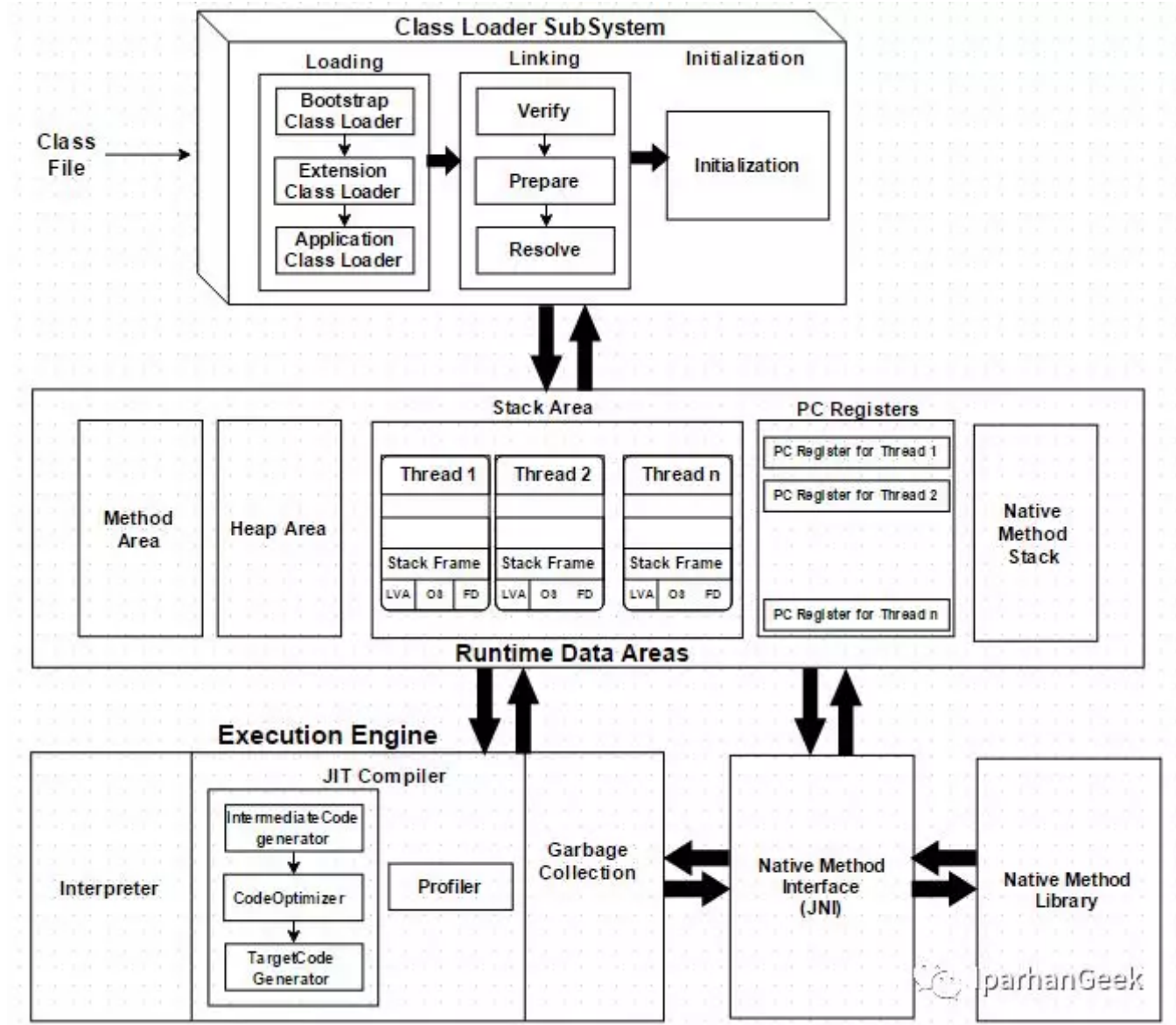


Paris Brune-Mustangcabriolet来自IparhanGeek00:1103:58

JVM架构

JVM架构-JVM架构结构

JVM 架构结构参考图：

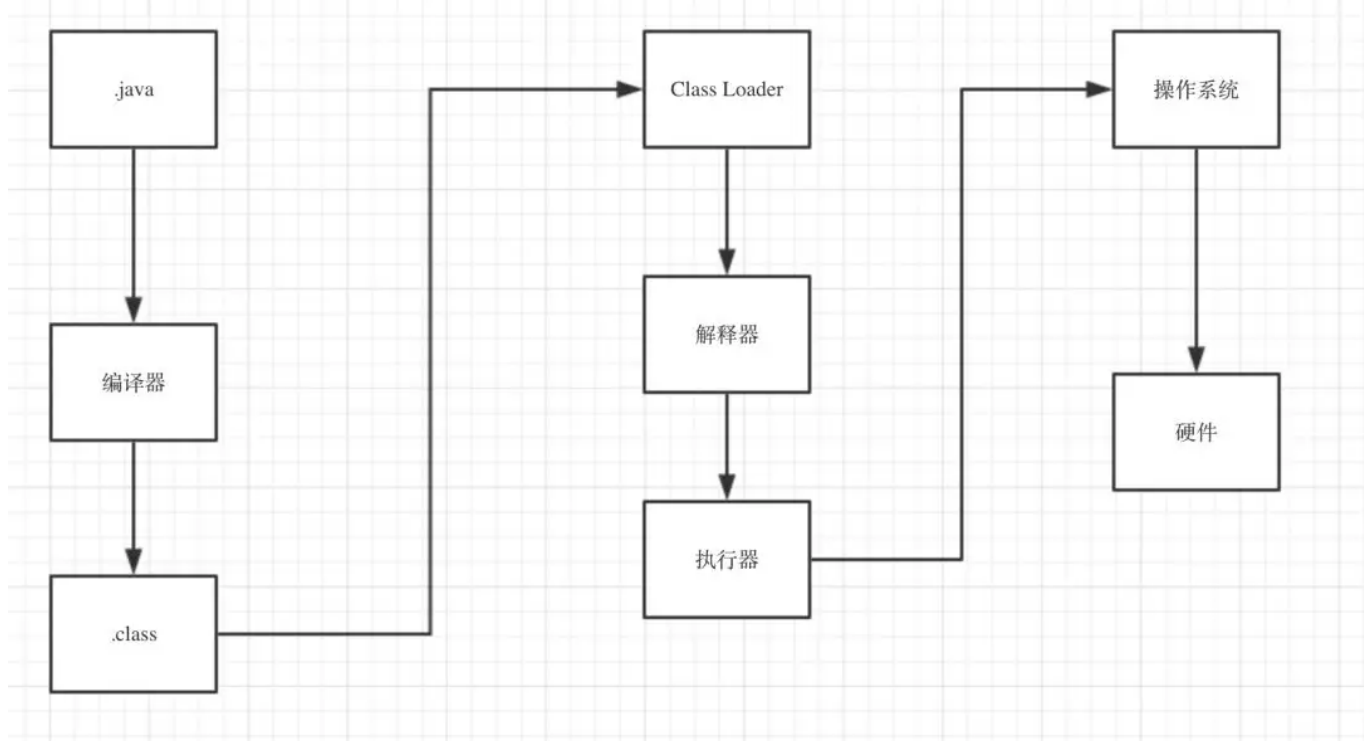


前期的文章我们说到JVM架构的初探这篇文章我们会谈到JVM初步到深入剖析甚至到源码分析和一些问答内容,首先我们先来看看JVM架构的结构分析(JVM 结构可以看上图为参考) 文字介绍:

JVM 整个过程

说到整个过程说简单说复杂因为其里面包含的知识点比较多,我们先看看JVM怎么把.java文件处理的过程:

下面是JVM的处理文件的整个过程(缩略图)

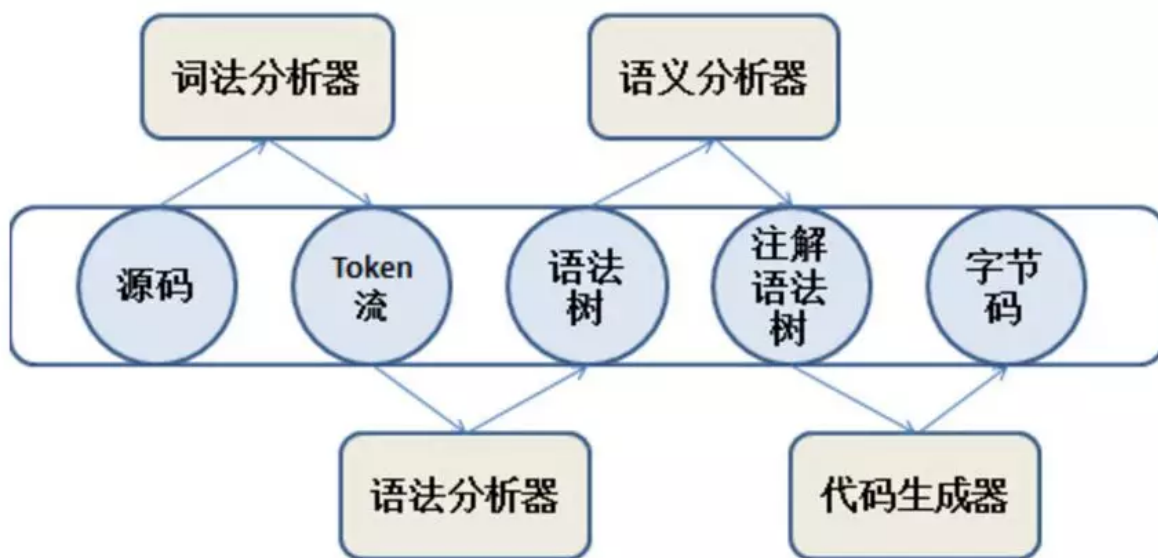


程序员Java文件编写

我们平常所说的Java代码编译整个过程是：

1. Java人员编写.java文件
2. 内存分配
3. 文件编译
4. 类加载
5. 类执行

首先我们就看编译器：



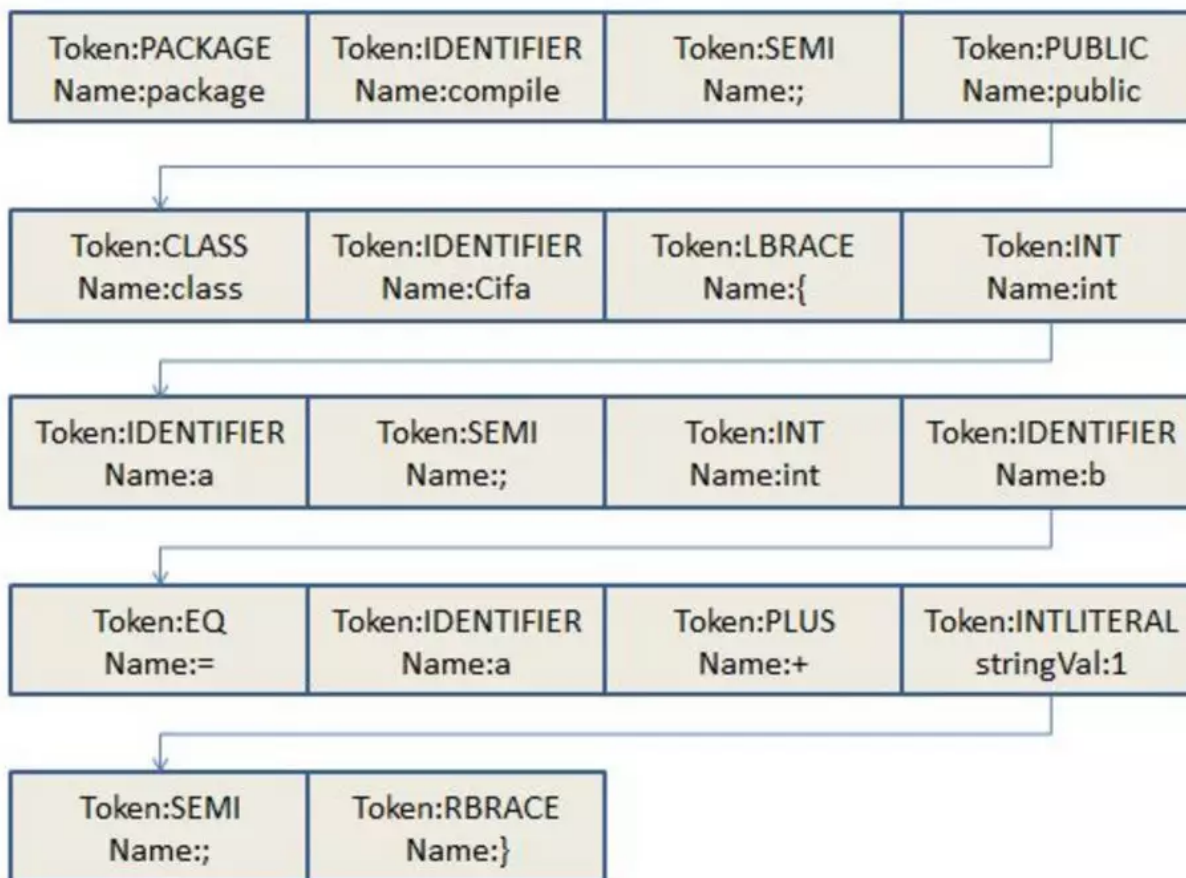
编译器编译过程：

1. 语法分析器
2. 词法分析器
3. 语义分析器

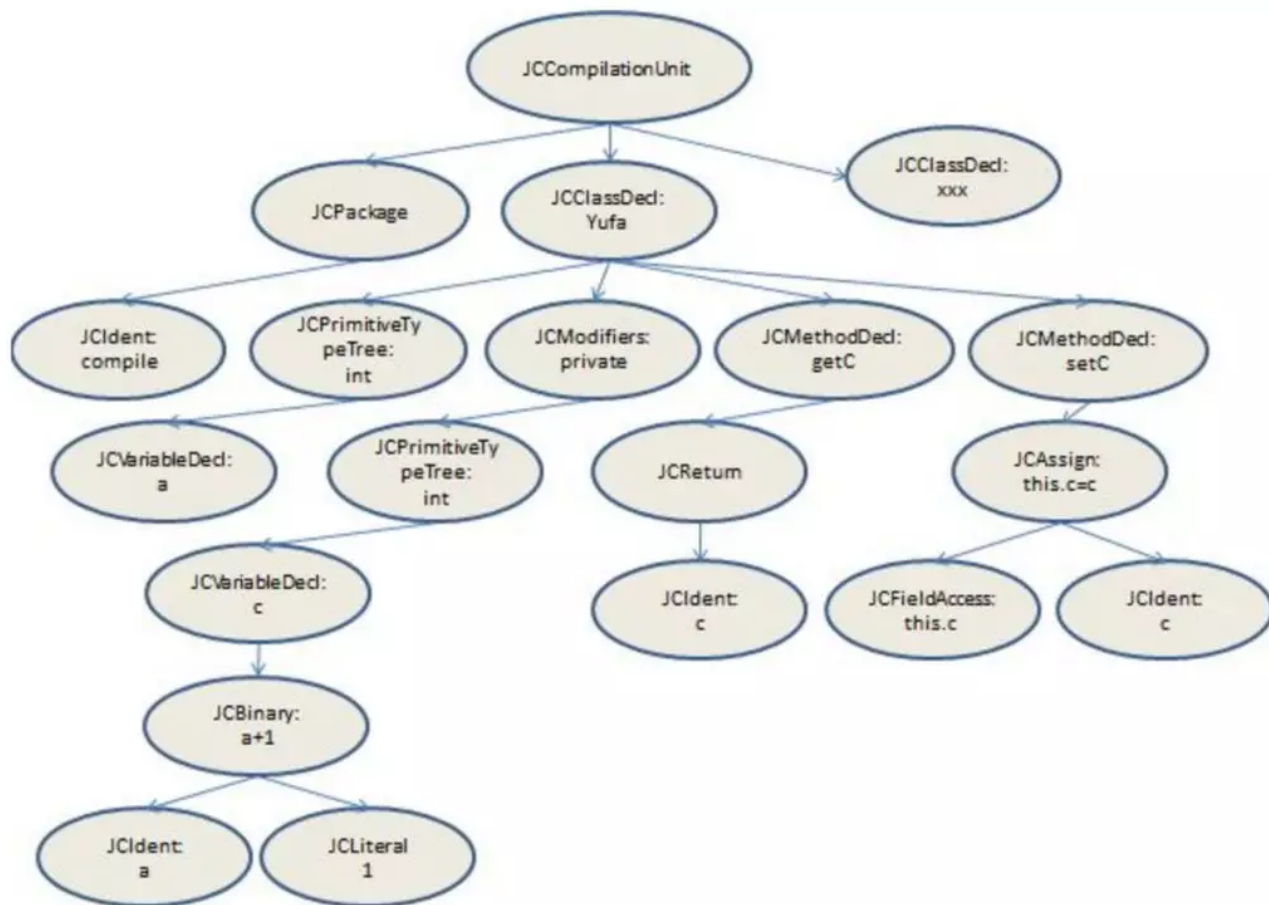
4.代码生成器

javac编译流程：

- **词法分析器**：将源码转换为Token流
 - 将源代码划分成一个个Token(找出java语言中的if, else, for等关键字)



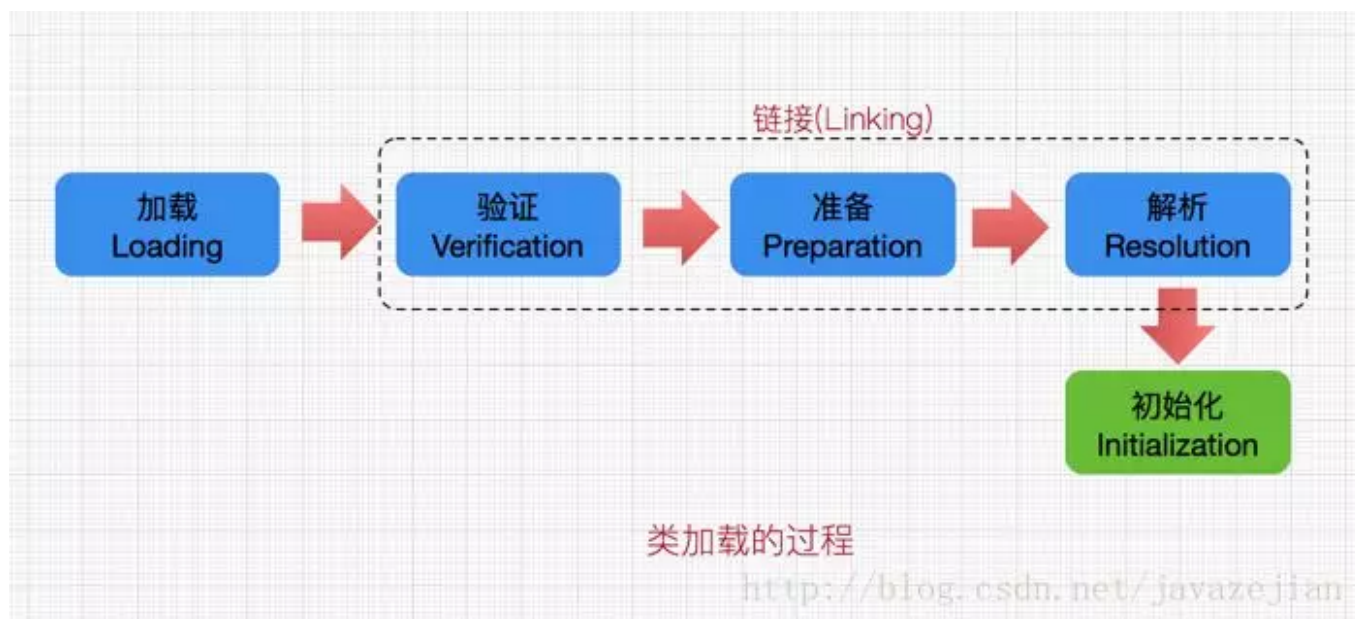
- **语法分析器**：将Token流转化为语法树
 - 将上述的一个个Token组成一句句话（或者说成一句句代码块），检查这一句话是不是符合Java语言规范(如if后面跟的是不是布尔判断表达式)



- **语义分析器**：将语法树转化为注解语法树
 - 将复杂的语法转化成简单的语法（eg.注解、foreach转化为for循环、去掉永不会用到的代码块）并做一些检查，添加一些代码(默认构造器)
- **代码生成器**：将注解语法树转化为字节码(即将一个数据结构转化成另一个数据结构)

类加载器 -- ClassLoader

编译器把.java文件编译成.class文件后再用加载器改成机器语言。



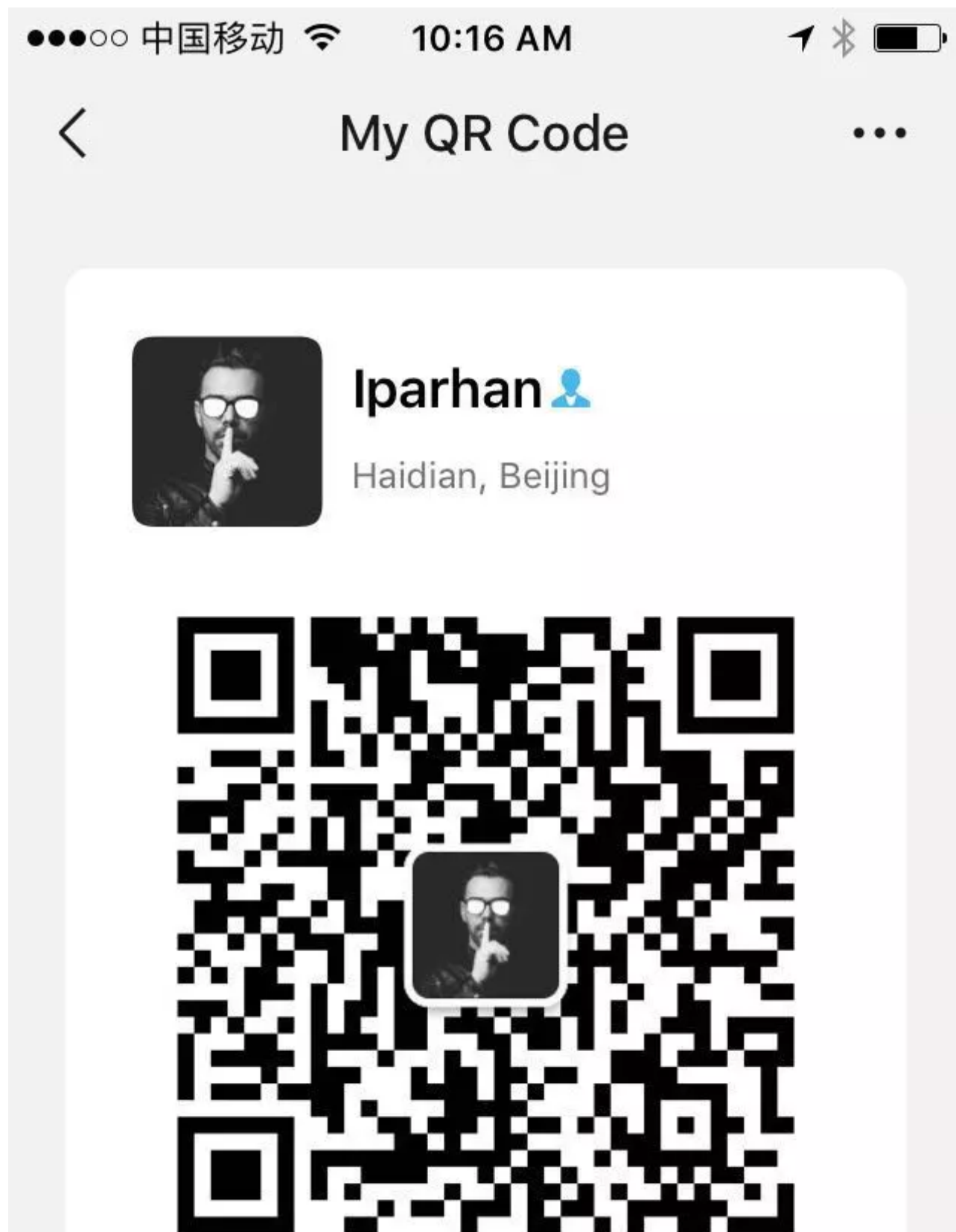
加载器分类：

- 1) Bootstrap类加载器 – JRE/lib/rt.jar
- 2) Extension类加载器 – JRE/lib/ext或者java.ext.dirs指向的目录
- 3) Application类加载器 – CLASSPATH环境变量, 由-classpath或-cp选项定义,或者是JAR中的Manifest的classpath属性定义.

END

IparhanGeek

海赛尔



Scan the QR code to add me on WeChat



Scan QR Code via
WeChat
to follow Official
Account