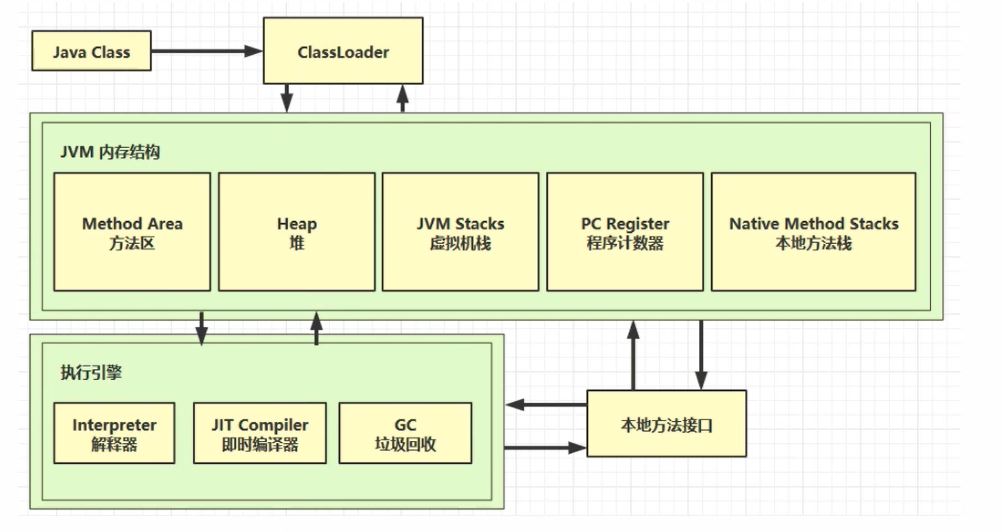
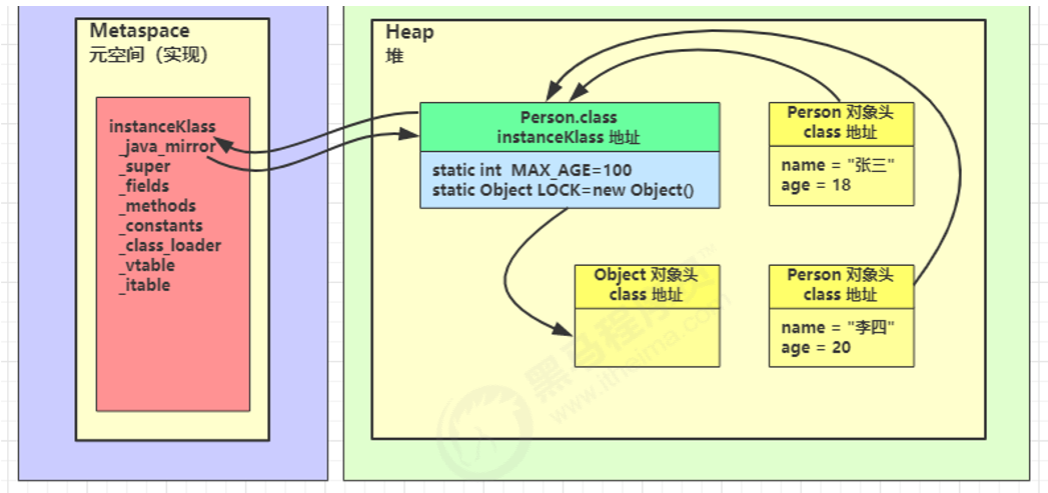
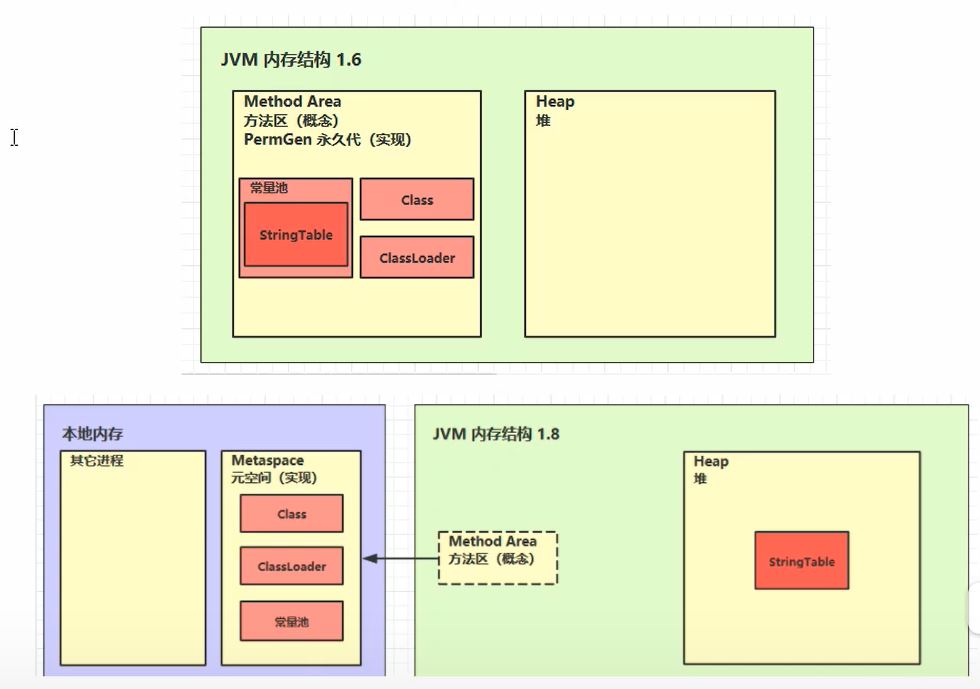
JVM结构





常量池



反编译

javap -v Test.class

类的加载和初始化

1.加载

将类的字节码载入方法区中，内部采用 C++ 的 instanceKlass 描述 java 类

2.链接

2.1校验

验证类是否符合 JVM规范，安全性检查

2.2准备

为 static 变量分配空间，设置默认值

static 变量在 JDK 7 之前存储于 instanceKlass 末尾，从 JDK 7 开始，存储于 \_java\_mirror 末尾 static

变量分配空间和赋值是两个步骤，分配空间在准备阶段完成，赋值在初始化阶段完成

如果 static 变量是 ﬁnal 的基本类型，以及字符串常量，那么编译阶段值就确定了，赋值在准备阶 段完成

如果 static 变量是 ﬁnal 的，但属于引用类型，那么赋值也会在初始化阶段完成

2.3解析

将常量池中的符号引用解析为直接引用

3.初始化

<cinit>()V 方法

初始化即调用 <cinit>()V ，虚拟机会保证这个类的『构造方法』的线程安全 发生的时机

概括得说，类初始化是【懒惰的】

main 方法所在的类，总会被首先初始化

首次访问这个类的静态变量或静态方法时

子类初始化，如果父类还没初始化，会引发

子类访问父类的静态变量，只会触发父类的初始化

Class.forName

new 会导致初始化

不会导致类初始化的情况

访问类的 static ﬁnal 静态常量（基本类型和字符串）不会触发初始化

类对象.class 不会触发初始化

创建该类的数组不会触发初始化