https://www.cnblogs.com/chanshuyi/p/jvm\_serial\_07\_jvm\_class\_loader\_mechanism.html

**类的加载机制：**

加载、验证、准备、解析、初始化、使用、卸载。

加载：加载代码到内存中（加载class字节码文件）。

验证：验证文件是否符合jvm规范，验证代码是否有语法等错误。

准备：为static修饰的对象/方法分配内存，但不初始化。static final修饰的分配内存并初始化。

解析：JVM 针对类或接口、字段、类方法、接口方法、方法类型、方法句柄和调用点限定符 7 类引用进行解析，将其在常量池中的符号引用替换成直接其在内存中的直接引用。

初始化：初始化类：按顺序初始化static修饰的。

初始化(new)对象：若类没有进行初始化，则先初始化类，再初始化对象；不初始化static修饰的。

初始化类时先初始化父类。

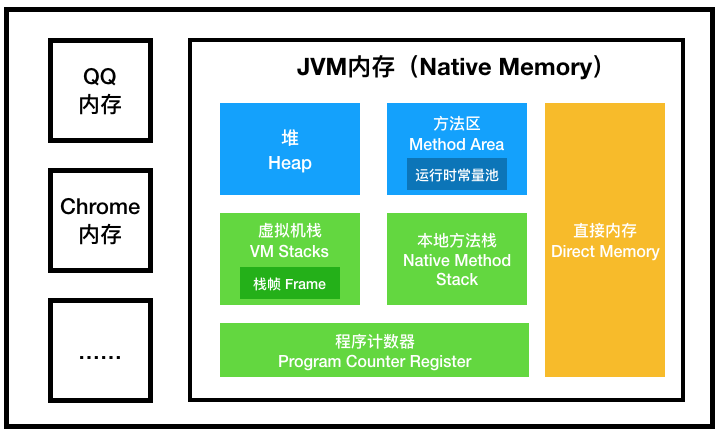
获取父类的静态变量时，只初始化父类。

使用：执行代码。

卸载：销毁class文件。

Spring项目启动时，jvm只加载并初始化在容器内的类及对象，不在容器内的类在用到时才会加载。

**JVM内存模型：**

****

线程共享部分：java堆，方法区，常量池。

Java堆：用于java实例对象的内存分配。

方法区&常量池：存储 Java 类字节码数据的一块区域，它存储了每一个类的结构信息，如运行时常量池、类的成员变量、类的方法等。Java8前称为永久代(Permanent Space)，

-XX:PermSize=配置；java8后称作元空间(MetaSpace)，-XX:MetaSpaceSize=配置。

线程私有部分：pc寄存器，java虚拟机栈，native方法栈。