JVM学习

JAVA的JVM的内存可分为3个区：堆(heap)、栈(stack)和方法区(method)

栈区:

1. 每个线程包含一个栈区,栈中只保存方法中(不包括对象的成员变量)的基础数据类型和自定义对象的引用(不是对象),对象都存放在堆中
2. 每个栈中的数据(原始类型和对象引用)都是私有的,其它栈不能访问
3. 栈分为三个部分:基本类型变量区,执行环境上下文,操作指令区(存放操作指令)

堆区:

1. 存储的全部是对象实例,每个对象都包含一个与之对应的class信息(class信息存放在方法区中)
2. jvm只有一个堆区(heap)被所有线程共享,堆中不存放基本类型和对象引用,只存放对象本身,几乎所有的对象实例和数组都在堆中分配

方法区:

又叫静态区,跟堆一样,被所有线程共享.它用于存储已经被虚拟机加载的类信息,常量,静态变量,即时编译器编译后的代码等数据.

虚拟机类加载机制:

初始化:

1.new 实例化对象,调用一个类的静态方法.

2.reflect反射的时候.

3.初始化类的时候先初始化父类,对于接口则是在使用到父接口的时候才会初始化.

4.启动方法main

对于静态字段,只有直接定义了这个字段的类才会被初始化,static,子类继承并不会.

定义数组并不会初始化类,详情见项目dailyDemo.设置成final常量的类,调用常量时不会初始化,因为编译的时候已经放到常量池中.

JVM启动准备阶段:

在方法区中分配类变量的内存,被static修饰的变量,并赋值为变量类型的零值,如下

public static int value = 123; --------🡪 赋值0 具体值需要在类初始化时执行.

如果类变量被final修饰成为constantValue 则在准备阶段赋值.

类加载器的双亲委派模型:

类加载器的继承关系: 启动类加载器🡪扩展类加载器🡪应用程序类加载器🡪自定义加载器

子类想要加载某一个class文件的时候会将请求委派给父级加载器最后传送到顶层,如果顶层加载器无法加载才会提示子类加载器去加载.这样可以避免应用程序的混乱.