# JVM

## 概念

### 什么是JVM？

JVM是可运行Java代码的假想计算机.它运行在操作系统之上,。

### JVM有什么用

实现了跨平台运行java文件,JVM负责把字节码文件转换成机器码文件

## JVM内存区域



### 线程私有区域

#### 虚拟机栈区

描述java方法执行的内存模型,每个方法在执行的同时都会创建一个栈帧,,栈帧用来存储局部变量,方法出口等与方法有关的信息.(每个方法的完成就代表进栈和出栈的过程)(对于对象变量,是在栈中创建一个地址,然后在堆中实例对象.,)

#### 本地方法区

与虚拟机栈区类似,只不过是为本地方法服务

#### 程序计数器区

当前线程所执行的字节码的行号指示器

### 线程共有区域

#### 堆区

创建的对象本身都保存在此区,,也是垃圾收集器进行垃圾收集的最重要的内存区域

#### 永久代区(方法区)

用于存储被JVM加载的类信息,常量,静态变量等数据其中里面包含了运行时常量池区域(常量池区域包含了编译器生成的各种字面量和符号引用)

### 以GC角度的堆区分区

在GC角度又可以将堆区分为新生代区(包含Eden区,From Suivivor区和To Suivivor区)和老年代区



#### 新生代区

##### Eden区

Java新对象的出生地(如果新对象的内存很大就会被直接分配到老年代区).当Eden区空间不够时,就会触发MinorGC操作,.对Eden区进行内存回收

##### From Survivor区

在上一次GC过程中活下来的幸存者,作为这一次GC操作的被扫描者

##### To Survivor区

保留了一次GC过程中的幸存者

##### MinorGC

对新生代的内存回收使用的是MinorGC,使用复制算法完成内存回收

###### GC过程

###### 1..复制

首先把Eden和From Survivor区存活的对象复制到To Servivor区,同时把复制后的对象年龄+1。如果对象年龄达到老年标准或者To Survivor区的空间不够就把对象复制到老年代区

###### 2.清空

把Eden区和From Survivor区清空

###### 3.互换

把To Survivor区与From Survivor区交换,即当前的To Survivor区是下一次GC操作的From Survivor区

#### 老年代区

主要存放应用程序中生命周期长的内存对象(包含了从新生代转过来的对象).

##### MajorGC

使用标记清除算法对老年代区域进行内存回收

#### GC过程

首先扫描一次所有老年代,标记出所有存活的对象,然后回收所有没有标记的对象

#### 永久代区

指内存的永久保存区域,.主要存放Class和元数据的信息,Class在被加载的时候就被放到了元数据区,GC不会在主程序运行过程中对永久代区进行清理

## JVM问题

### 类加载过程



#### 加载

将class字节码文件加载到内存中,并将这些数据转换成方法区的运行时数据(静态变量,静态代码块,常量池等),在堆中生成一个Class类对象代表这个类,作为方法区数据的访问入口

#### 链接

将Java类的二进制代码合并到JVM的运行状态之中。  
• 验证  
确保加载的类信息符合JVM规范，没有安全方面的问题。  
• 准备  
正式为类变量(static变量)分配内存并设置类变量初始值的阶段，这些内存都将在方法区中进行分配。注意此时的设置初始值为默认值，具体赋值在初始化阶段完成。  
• 解析  
虚拟机常量池内的符号引用替换为直接引用（地址引用）的过程。

#### 初始化

初始化阶段是执行类构造器<clinit>()方法的过程

### 什么是Java虚拟机？为什么Java被称作是“平台无关的编程语言”？

JVM,运行操作系统之上的假想计算机,用来执行java代码,因为JVM的存在所以java代码只需要运行在JVM上,所以与平台无关

### JVM分区

分为线程私有与线程共享

线程私用:虚拟栈区,程序计数器区,本地方法栈区

线程共享:堆区,方法区

### JAVA虚拟机中，哪些可作为ROOT对象？

(ROOT对象就是GC ROOT即活跃的引用)

虚拟机栈中的引用对象

方法区中类静态属性引用的对象

方法区中常量引用的对象

本地方法栈中J引用的对象

### JVM内存模型是什么？

JMM就是之前的分区

### GC中如何判断对象需要被回收？

判断对象是否有用

引用计数算法

可达性分析算法