

Java---程序执行过程中的内存分析

原创

风中飞舞的火

2019-01-03 10:11:47

👁

版权

925 ★ 收藏 5

分类专栏: JAVA知识体系

文章标签: Java

内存分析

堆

栈

必须知道

在了解内存的内部运作方式之前，我们必须要知道3个概念；栈、堆、方法区；

官方给出的定义为：

- 1.栈：存放-局部变量
- 2.堆：存放-new出来的新对象
- 3.方法区：存放-类的信息（代码）、static变量、常量池（字符串常量）等；

首先声明一个类（以Student为例）和执行入口

```
public class Student {

    //静态的数据
    String name;

    int age;

    int id;
    |
    int gender;

    int weight;

    //动态的行为
    public void study(){
        System.out.println(name+"在学习");
    }

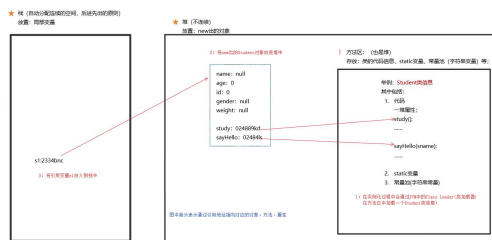
    public void sayHello(String sname){
        System.out.println(sname + "说：你好");
    }
}
https://blog.csdn.net/Eric\_smxu
```

```
public static void main(String[] args) {
    Student s1 = new Student();
    s1.name = "小明";
    s1.study();
    s1.sayHello("小高");

    Student s2 = new Student();
    s2.name = "小红";
    s2.age = 18;
}
https://blog.csdn.net/Eric\_smxu
```

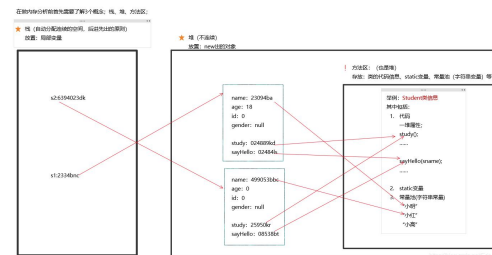
1.在执行入口中对对象实例化：Student s1 = new Student();
实例化，声明一个引用 类型变量s1（除基本数据类型外，Java
中其他所有的变量声明皆为引用类型：）

在做这一步操作时，内存中主要做了三个操作；如下图：



这里需要注意的是：在方法区加载Student类信息时会先去方法区中找该类是否已经实例化，如果已经被实例化；则不再加载；

2.这时我们开始为Student类的变量赋值，并调用类中方法，同时声明一个s2进行对比，结果如下图：



首先，我们发现在方法区的常量池里面多出了3个常量，这三个常量的来自于对象实例化后的赋值操作；

常量池具有共享机制，在声明常量时会首先到常量池中遍历是否有相同常量，如果有相同的，则不再声明

同时，将堆中对应对象的属性“name”通过引用地址指向常量池中对应的常量；而像age、id这种基本类型的数据，直接赋值即可；

多类加载内存分析

上图我们是把Student类和执行类放在一起的（项目中一般会分开）；这时我们再添加一个Computer类；如图：

```
public class Computer {
    String brand;
    int cpuSpeed;
}
```

在Student类中添加Computer属性：

```
public class Student {

    //静态的数据
    String name;

    int age;

    int id;

    int gender;

    int weight;

    Computer computer;
    //动态的行为
    public void study(){
        System.out.println(name+"在学习");
    }

    public void sayHello(String sname){
        System.out.println(sname + "说：你好");
    }
}
```

添加执行类：

```

public class TestComputer {
    public static void main(String[] args) {
        Student s1 = new Student(); //实例化, 声明-

        s1.name = "小明";
        s1.age = 18;

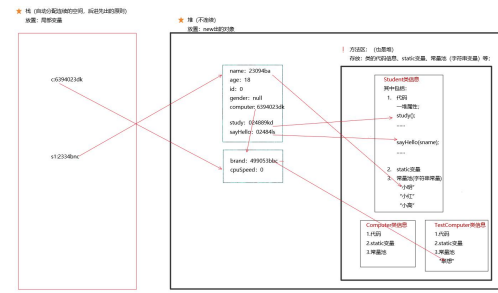
        Computer c = new Computer();
        c.brand = "联想";

        s1.computer = c;

        System.out.println(s1.computer.brand);
    }
}

```

这样，在程序启动时一共有三个类被加载到方法区中，如下图：



根据代码显示可知：

c.brand = “联想”

因此在TestComputer类里面的常量池会加载联想这个常量

和图二一样，声明完成后赋值，加载方法区，new一个新对象，将栈中的变量名称指向堆里面new的对象的地址；然后我们再看s1指向堆中的对象，其中关联了computer对象（表示某个学生有一台电脑），通过s1.computer.brand，我们将c的地址指向到s1中的computer，这个时候c指向的地址实际上和computer指向的地址一致，因此，当改变c的值（即改变c的指向地址，那么s1.computer的指向地址也一样会被改变）

Java学习1：图解Java内存分... Liu_Arvin的芝士小栈 1万+

Java学习1：图解Java内存分析详解（实例）

linux下分析Java程序内存汇总 zizlei的专栏 2万+

使用pmap查看进程内存 运行命令使用pmap可以查看某一个进程...

Java 程序内存分析，深入Java核... hemingwey的专栏 437

Java 程序内存分析
Java内存分配与管理是Java的核...

java程序执行过程中内存分析(转)_GoodLiang的博客 2-6

内存分析 Java程序运行在JVM上,可以把JVM理解成Java程序和...

Java程序执行过程的内存分析_yangzheng'Blog_ja... 1-20

程序执行过程的内存分析 示例一: Student.java package cn.bjst...

java程序调用内存变化过程分析(详细) 于华 5313

java程序调用内存分析详解(内存分析太重要了，这个类比于武...

java程序运行时内存变化图解 XMprogram的博客 605

程序执行的内存分析过程 为了让初学者顺利的分析内存，更加容...

java程序执行过程中内存分析详解_tudanbaotian的... 2-13

java程序执行过程中内存分析详解 //编写Person类publicclassPer...

JAVA程序执行过程的内存分析_lixiaoxiong的专栏 1-29

JAVA程序执行过程的内存分析 内存分析先知 1、每个线程私有,...

Java程序内存分析Java Visu... weixin_33728268的博客 222

VisualVM 是一款免费的集成了多个JDK 命令行工具的可视化工...

java方法执行内存分析 qq_42895153的博客 61

注意:在!