

day3

时间：2022年4月8日22:22:32

目录

- 局部变量
- bean的作用域

一、局部变量

1.代码

```
public class Exam5 {  
    static int s;  
    int i;  
    int j;  
    {  
        int i = 1;  
        i++;  
        j++;  
        s++;  
    }  
    public void test(int j){  
        j++;  
        i++;  
        s++;  
    }  
    public static void main(String[] args) {  
        Exam5 obj1 = new Exam5();  
        Exam5 obj2 = new Exam5();  
        obj1.test(10);  
        obj1.test(20);  
        obj2.test(30);  
        System.out.println(obj1.i + "," + obj1.j + "," + obj1.s);  
        System.out.println(obj2.i + "," + obj2.j + "," + obj2.s);  
    }  
}
```

运行

让天下

2.考点

- 就近原则
- 变量的分类
 - 成员变量:类变量、实例变量
 - 局部变量
- 非静态代码块的执行:每次创建实例对象都会执行
- 方法的调用规则:调用一次执行一次

3.局部变量 成员变量的区别

a.声明的位置

- 局部变量：方法体{}、形参、代码块{}
 - 类变量：有static修饰
 - 实例变量：没有static修饰

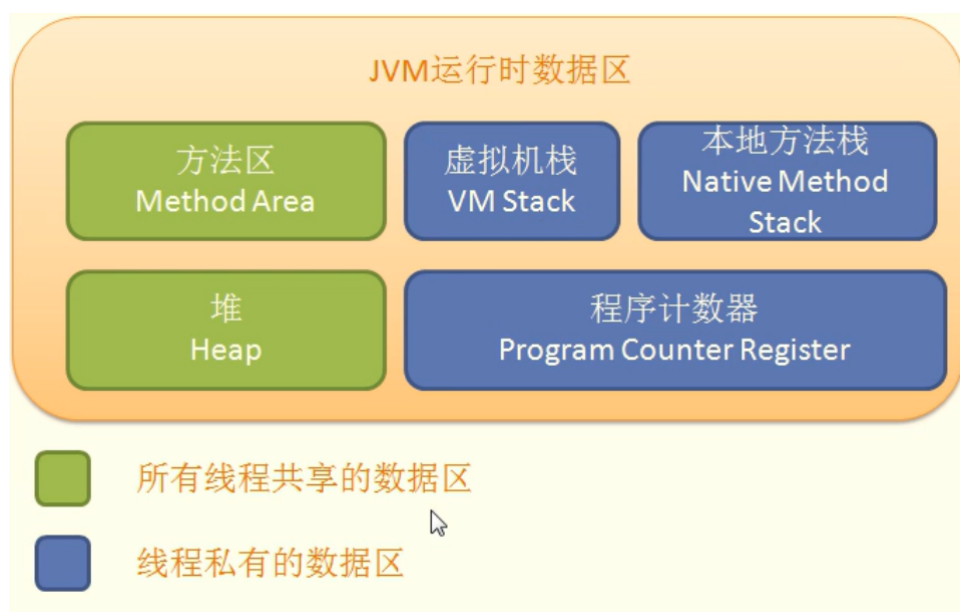
b.修饰符

- 局部变量：final
- 成员变量：public、protected、private、final、static、volatile、transient

c.值存储位置

- 局部变量：栈
- 实例变量：堆
- 类方法：方法区

堆栈



- 1 堆 (Heap)
- 2 此内存区域的唯一目的就是存放对象实例，几乎所有的对象实例都在这里分配内存。这一点在Java虚拟机规范中的描述是：所有的对象实例以及数组都要在，堆上分配。
- 3
- 4 通常所说的栈 (Stack)
- 5 是指虚拟机栈。虚拟机栈用于存储局部变量表等。局部变量表存放了编译期可知长度的各种基本数据类型 (boolean、byte、char、short、int、float、long、double)、对象引用 (reference 类型，它不等同于对象本身，是对象在堆内存的首地址)。方法执行完，自动释放。
- 6
- 7 方法区 (Method Area)
- 8 用于存储已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。

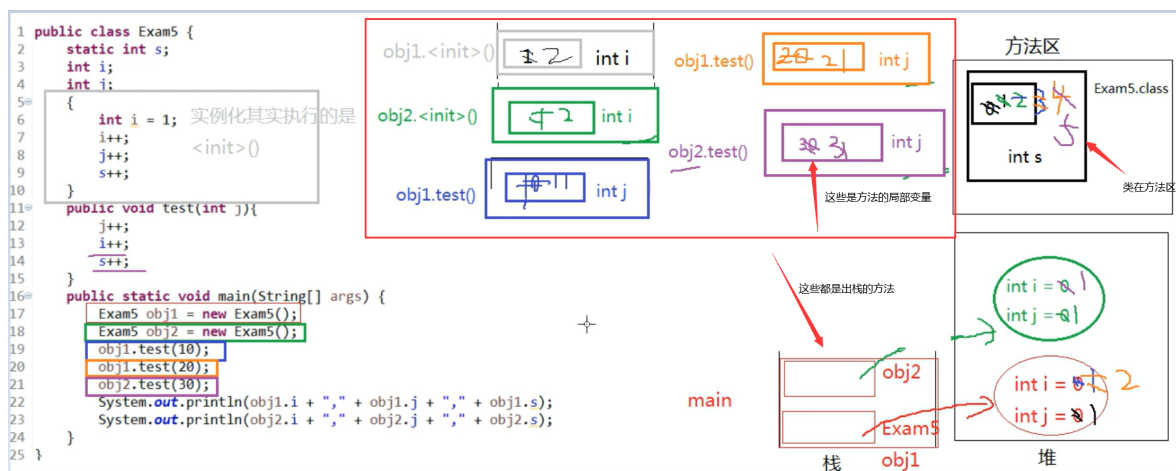
d.作用域

- 局部变量:从声明处开始，到所属的}结束
- 实例变量:在当前类中“this.”(有时this.可以缺省)，在其他类中“对象名.”访问
- 类变量:在当前类中“类名.”(有时类名.可以省略)，在其他类中“类名.”或“对象名.”访问

e.生命周期

- 局部变量:每一个线程，每一次调用执行都是新的生命周期
- 实例变量:随着对象的创建而初始化，随着对象的被回收而消亡，每一个对象的实例变量是独立的
- 类变量:随着类的初始化而初始化，随着类的卸载而消亡，该类的所有对象的类变量是共享的

4.内存模型



- 栈的进出单位是方法
- 类是在方法区中

二、bean的作用域

1.前言

在Spring中，可以在元素的scope属性里设置bean的作用域，以决定这个bean是单实例的还是多实例的。

默认情况下，Spring 只为每个在IOC容器里声明的bean创建唯一的一个实例，整个IOC容器范围内都能共享该实例:所有后续的getBean()调用和bean引用都将返回这个唯一的bean实例。该作用域被称为 singleton,它是所有bean的默认作用域。

类别	说明
singleton	在SpringIOC容器中仅存在一个Bean实例, Bean以单实例的方式存在
prototype	SpringIOC容器不在实例化bean ，只有在每次调用getBean()时都会返回一个新的实例
request	每次HTTP请求都会创建一个新的 Bean，该作用域仅适用于WebApplicationContext环境
session	同一个HTTP Session共享一个Bean，不同的HTTP Session使用不同的Bean。该作用域仅适用于WebApplicationContext环境

2.代码

```
import com.atguigu.spring.beans.Book;

class SpringTest {

    //创建IOC容器对象
    ApplicationContext ioc = new ClassPathXmlApplicationContext("beans.xml");

    @Test
    void testBook() {
        Book book = (Book) ioc.getBean("book");
        Book book2 = (Book) ioc.getBean("book");
        System.out.println(book==book2);
    }
}
```