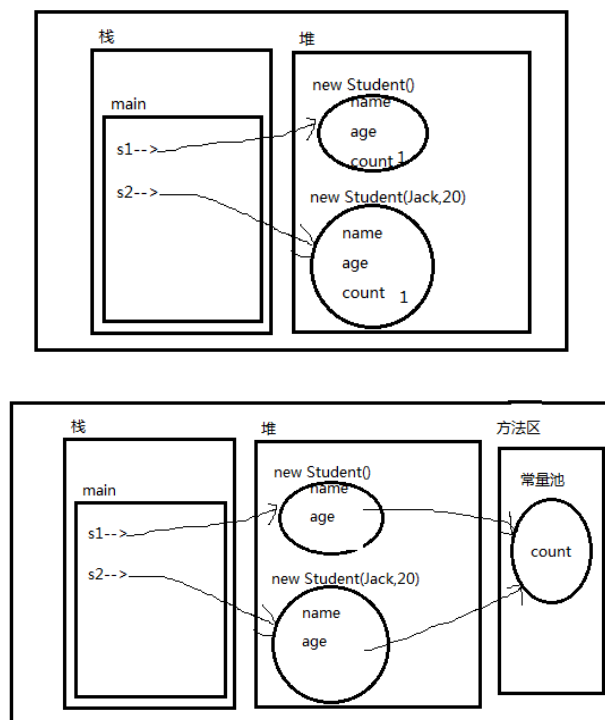


# Java Day 08

## static

统计类实例对象创建的次数？



### static 关键字的特点

- 1): 随着类的加载而加载
- 2): 优先于对象存在
- 3): 被类的所有对象共享

那么它就应该定义为静态的

举例：

饮水机(用静态修饰)

水杯(不能用静态修饰)

共用静态,特性用非静态

4):可以通过类名调用

其实它本身也可以通过对象名调用

推荐使用类名调用

静态修饰的内容一般我们称其为:与类相关的,类成员

### *static* 的注意事项

1):在静态方法中不能使用 `this` 关键字

静态是随着类的加载而加载, `this` 是随着对象的创建而存在,  
静态比对象先存在。

2):静态方法只能访问静态的成员变量和静态的成员方法

静态方法:

成员变量:只能访问静态变量

成员方法:只能访问静态成员方法

非静态方法:

成员变量:可以是静态的,也可以是非静态的

成员方法:可以是静态的成员方法,也可以是非静态的成

员方法。

### 静态变量和成员变量的区别

#### 1): 所属不同

静态变量属于类，所以也称为类变量

成员变量属于对象，所以也称为实例变量(对象变量)

#### 2): 内存中位置不同

静态变量存储于方法区的静态区

成员变量存储于堆内存

#### 3): 内存出现时间不同

静态变量随着类的加载而加载，随着类的消失而消失

成员变量随着对象的创建而存在，随着对象的消失而消失

#### 4): 调用不同

静态变量可以通过类名调用，也可以通过对象调用

成员变量只能通过对象名调用

---

代码块

---

### 代码块概述

在 Java 中，使用{}括起来的代码被称为代码块

### 代码块分类

根据其位置和声明的不同，可以分为局部代码块、构造代码块、静态代码块、同步代码块(多线程讲解)

### 常见代码块的应用

#### 1):局部代码块

在方法中出现；限定变量生命周期，及早释放，提高内存利用率

#### 2):构造代码块（初始化块）

在类中方法外出现；多个构造方法中相同的代码存放到一起，每次调用构造方法都执行，并且在构造方法前执行

#### 3):静态代码块

在类中方法外出现，并加上 `static` 修饰；用于给类进行初始化，在加载的时候就执行，并且只执行一次。一般用于加载驱动

---

静态导入：将类中的静态属性、静态方法导入本类，

直接访问，简化编程

---

---

## final

---

*final* 关键字修饰类,方法以及变量的特点-->FinalTest.java

修饰类, 类不能被继承

修饰变量, 变量就变成了常量, 只能被赋值一次; 一般与 public static 同用

修饰方法, 方法不能被重写

*final* 关键字修饰局部变量

基本类型, 是值不能被改变

引用类型, 是地址值不能被改变, 对象中的属性可以改变

*final* 修饰成员变量的初始化时机

显示初始化

在对象构造完毕前即可 (构造器中)

---

## abstract

---

抽象类: 不具体的类, 不能实例化, 用来继承用的  
-->AbstractTest.java

- 1) abstract 修饰类、方法
- 2) 抽象类不能实例化

- 3) 抽象类中不一定有抽象方法，有抽象方法一定是抽象类
- 4) 抽象类的子类要么是抽象的，要么实现父类中所有抽象方法
- 5) `abstract` 不能修饰变量
- 6) `abstract` 不能与 `static`、`final`、`private` 同用
- 7) 抽象类有构造器

---

## interface

---

接口：一种特殊的抽象类，用来定义规范，让子类实现  
-->InterfaceTest.java

- 1) `interface` 进行定义
- 2) 不能实例化
- 3) 子类要实现接口中所有抽象方法
- 4) 变量必须是常量 (`public static final`)
- 5) 成员方法都是抽象的 (`public abstract`)
- 6) 接口没有构造器

---

## 类、接口继承关系-->InterfaceTest2.java

---

- 1) 类与类：`extends` 继承关系，只能单继承，可以多层次继承

2) 类与接口: `implements` 实现关系, 可以单实现, 也可以多实现; 可以继承一个类, 同时实现多个接口

3) 接口与接口: `extends` 继承关系, 可以单继承, 也可以多继承