

1, static关键字修饰属性:

(String修饰的对象都是在常量池中)

static是表示静态的, 可以用来修饰属性, 方法, 代码块、(初始化块)、内部类

①static修饰属性(类变量): static 修饰的变量属于该类本身, 并不属于任何实例对象, 存放在静态域中, 可以通过对象来调用类变量。

②由类创建的所有的对象, 都共用这一个属性

③当其中一个对象对此属性进行修改, 会导致其他对象对此属性的一个调用都会发生变化。(实例变量非static修饰的属性, 各个对象各自拥有一套属性)

④类变量随着类的加载而加载, 而且只有一份

⑤静态的变量可以直接通过“类.类变量”的形式来调用

⑥类变量的加载是要早于对象, 所以当有对象以后, 可以通过“对象.类变量”使用, 但是“类.实例变量”是不行的

⑦类变量存在于静态域中

(实例变量在未创建这个类的对象的时候, 虚拟机是不给实例变量分配空间, 但是对于一个类变量, 该类在加载的时候, 该属性就已经在静态域了, 而且只有一份; 实例变量是随着对象的加载而被加载的)

2, static修饰方法(类方法):

①随着类的加载而加载, 且在内存中只存在一份;

②可以直接通过“类名.方法进行调用类方法”

③静态方法只能调用静态的属性或者静态的方法, 而不能调用非静态的属性和方法, 反之, 非静态的方法是可以调用静态的方法的; (静态方法内是不可以有this或super关键字的)

注意: 静态结构(static修饰的属性, 方法, 代码块, 内部类)的生命周期要早于非静态的属性和方法, 同时被回收也要晚于非静态的属性和方法。

1.java.lang.Object 类，是所有类的根父类！

2.Object类仅有一个空参的构造器 public Object(){ }

3.关于方法：

① equals(Object obj)

```
public boolean equals(Object obj) {  
    return (this == obj);  
}
```

// ==

// 1.基本数据类型：根据基本数据类型的值判断是否相等。相等返回true，反之返回false

// 注：两端数据类型可以不同，在不同的情况下，也可以返回true。

// 2.引用数据类型：比较引用类型变量的地址值是否相等。

//equals():

>①只能处理引用类型变量②在Object类，发现equals()仍然比较的两个引用变量的地址值是否相等

>像String 包装类 File类 Date类这些重写Object类的equals()方法，比较是两个对象的

//“实体内容”是否完全相同。

>若我们自定义一个类，希望比较两个对象的属性值都相同的情况下返回true的话，就需要重写Object类的equals(Object obj)方法