# static关键字

## 特点

1.随着类的加载而加载，优先于对象存在

2.被类的所有对象共享，实现对象之间重复属性的数据共享

3.static存在的意义：表明一个成员变量/方法在对象（实例）不存在时便可以访问

## 修饰及调用

主要用于修饰类的成员（成员变量，成员方法）以及代码块

调用：1.类名直接调用

2.对象访问

## 注意事项

### 共性和特性

1.共性用静态,特性用非静态

\* 举例：咱们班级的学生应该共用同一个班级编号。

\* 其实这个特点也是在告诉我们什么时候使用静态?

\* 如果某个成员变量是被所有对象共享的，那么它就应该定义为静态的。

\* 举例：

\* 饮水机(用静态修饰)，共同喝一个

\* 水杯(不能用静态修饰)，各喝各的

### 无this，super

2.在静态方法中是没有this，super关键字的

静态是随着类的加载而加载，this是随着对象的创建而存在，super随着父类存在而存在

静态比对象先存在。

### 静态只能访问静态

3.静态方法只能访问静态的成员变量和静态的成员方法



# 静态成员

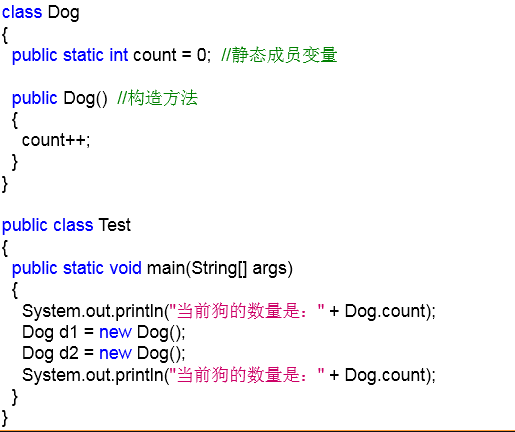
## 静态成员变量

### 概述

在成员变量前加static关键字，可以将其声明为静态成员变量；

如果类中成员变量被定义为静态，那么不论有多少个对象，静态成员变量只有一份内存拷贝，即所有对象共享该成员变量；

静态成员变量的作用域只在类内部，但其生命周期却贯穿整个程序。



### 注意事项

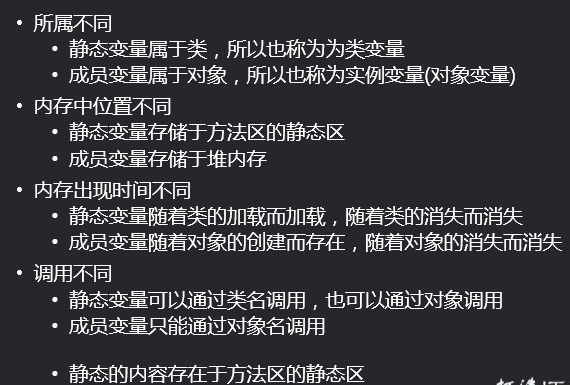
和程序同生命周期；

在没有实例化对象时，可以通过类名访问静态成员变量；

也可以通过对象访问静态成员变量，但不论使用的是哪个对象，访问到的都是同一个变量；

静态成员变量在声明时最好初始化，如果不进行初始化，系统会默认初始化为0。

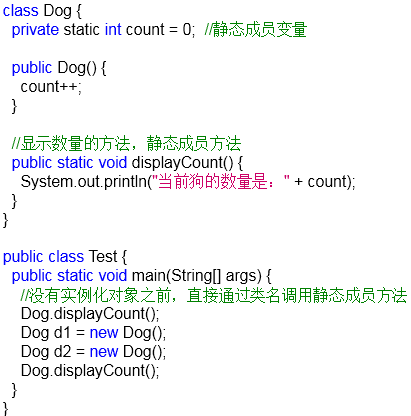
### 静态成员变量和成员变量的区别



## 静态成员方法

### 概述

* 在成员方法前加static关键字，可以将其声明为静态成员方法；
* 静态成员方法只能对类的静态成员变量进行操作；
* 静态成员方法没有this引用；
* 在没有实例化对象时，可以通过类名访问静态成员方法。



## 小结

静态成员包括静态数据成员和静态成员方法；

静态成员属于整个类而不是属于某个对象，它被该类的所有对象共享；访问静态成员时一般通过类名访问，也可以通过对象访问；

静态成员也受访问权限的约束；

静态数据成员在使用之前应尽量初始化；

静态成员方法不能操作非静态成员；

静态成员方法没有this引用。