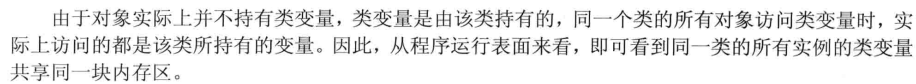
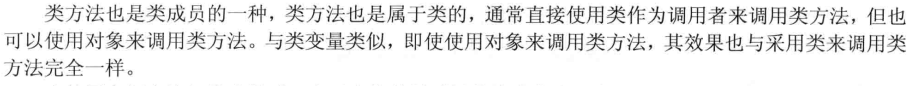
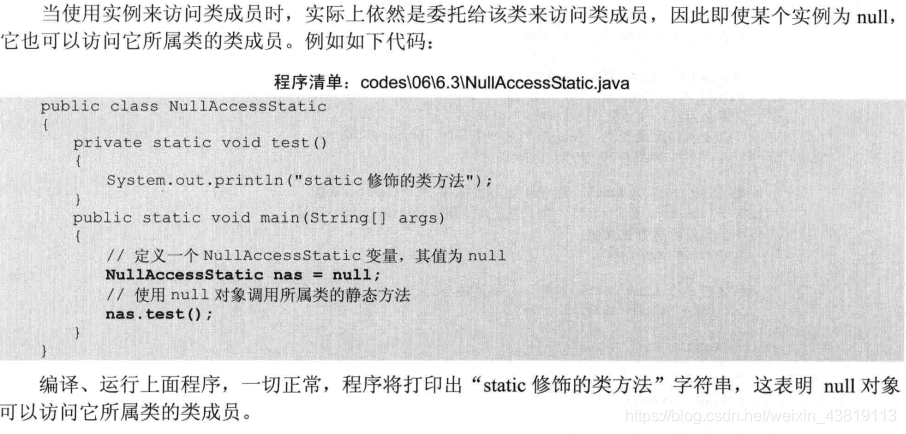
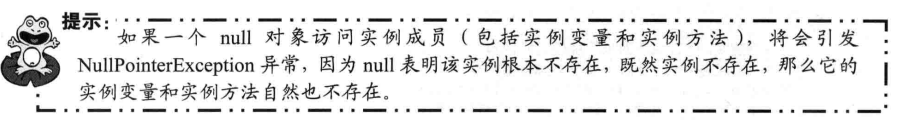
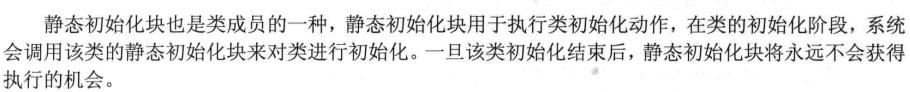
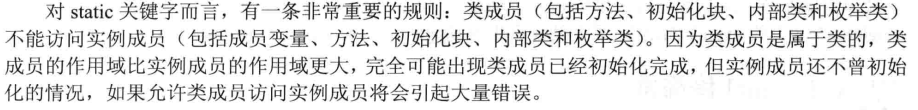
**static修饰的成员就是类成员**，前面介绍的类成员有类变量、类方法、静态初始化块三种。  
static 关键字不能修饰构造器。  
static 修饰的类成员属于整个类，不属于单个实例。

本文内容：

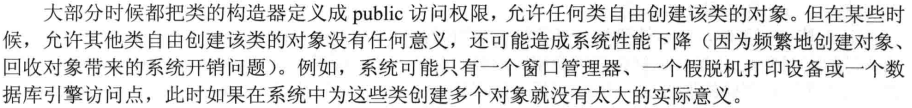
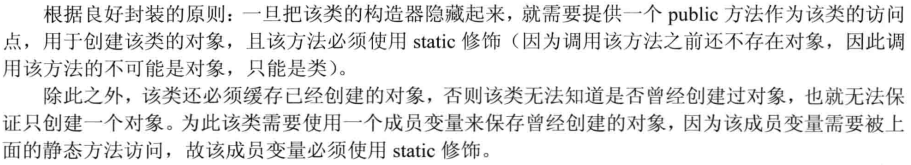
1. 理解类成员
2. 单例（Singleton）类

**1. 理解类成员**

Java类包含5种成员：成员变量、方法、构造器、初始化块、内部类（包括接口、枚举）5种成员。  
static可以修饰除了构造器之外的其它四种。

类变量属于整个类，当系统第一次准备使用该类时，系统会为该类变量分配内存空间，类变量开始生效，直到该类被卸载，该类的类变量所占有的内存才被系统的垃圾回收机制回收。  
类变量的生存范围几乎等于该类的生存范围。当类初始化完成后，类变量也被初始化完成。  
类变量可以通过类来访问，也可以通过类的对象来访问。  
但通过类的对象来访问时，实际上并不是访问该对象拥有的变量，因为系统创建类的对象时，系统不会为类变量分配内存，也不会再次对类变量进行初始化，即对象不拥有对应类的类变量。通过对象访问类变量只是一种假象，通过对象访问的仍然是该类的类变量。可以理解为：当通过对象来访问类变量时，系统会在底层转换为通过该类来访问类变量。  
类变量的调用：  
  
类方法的调用：  
  
实例访问类成员实际上是委托给该类来访问类成员，因此即使是 null 对象也能访问类的类成员。  
一般来说， null对象访问实例成员会出现NullPointerException异常  
静态初始化块：  
  
static不能访问非static的：  


**2.单例（Singleton）类**

有时候，将类的构造器定义成 public 访问权限，允许其他类自由创建该类的对象没有意义，会造成性能下降。  
  
**如果一个类始终只能创建一个实例，那么这个类称为单例类。**  
为了避免其他类自由创建该类的实例，可以把类的构造器使用private修饰，从而隐藏构造器。  
  
下面程序创建了单例类：  
