**成员变量和类变量的区别**

由static修饰的变量称为静态变量，其实质上就是一个全局变量。如果某个内容是被所有对象所共享，那么该内容就应该用静态修饰；没有被静态修饰的内容，其实是属于对象的特殊描述。

不同的对象的实例变量将被分配不同的内存空间， 如果类中的成员变量有类变量，那么所有对象的这个类变量都分配给相同的一处内存，改变其中一个对象的这个类变量会影响其他对象的这个类变量，也就是说对象共享类变量。

成员变量和类变量的区别：

   1、两个变量的生命周期不同

      成员变量随着对象的创建而存在，随着对象的回收而释放。

      静态变量随着类的加载而存在，随着类的消失而消失。

   2、调用方式不同

      成员变量只能被对象调用。

      静态变量可以被对象调用，还可以被类名调用。

   3、别名不同

      成员变量也称为实例变量。

      静态变量也称为类变量。

   4、数据存储位置不同

      成员变量存储在堆内存的对象中，所以也叫对象的特有数据。

      静态变量数据存储在方法区（共享数据区）的静态区，所以也叫对象的共享数据。

static 关键字，是一个修饰符，用于修饰成员(成员变量和成员函数)。

   特点：

   1、想要实现对象中的共性数据的对象共享。可以将这个数据进行静态修饰。

   2、被静态修饰的成员，可以直接被类名所调用。也就是说，静态的成员多了一种调用方式。类名.静态方式。

    3、静态随着类的加载而加载。而且优先于对象存在。

弊端：

   1、有些数据是对象特有的数据，是不可以被静态修饰的。因为那样的话，特有数据会变成对象的共享数据。这样对事物的描述就出了问题。所以，在定义静态时，必须要明确，这个数据是否是被对象所共享的。

   2、静态方法只能访问静态成员，不可以访问非静态成员。

      因为静态方法加载时，优先于对象存在，所以没有办法访问对象中的成员。

   3、静态方法中不能使用this，super关键字。

      因为this代表对象，而静态在时，有可能没有对象，所以this无法使用。

什么时候定义静态成员呢？或者说：定义成员时，到底需不需要被静态修饰呢？

成员分两种：

   1、成员变量。（数据共享时静态化）

      该成员变量的数据是否是所有对象都一样：

      如果是，那么该变量需要被静态修饰，因为是共享的数据。

      如果不是，那么就说这是对象的特有数据，要存储到对象中。

   2、成员函数。（方法中没有调用特有数据时就定义成静态）

      如果判断成员函数是否需要被静态修饰呢？

      只要参考，该函数内是否访问了对象中的特有数据：

      如果有访问特有数据，那方法不能被静态修饰。

      如果没有访问过特有数据，那么这个方法需要被静态修饰。

成员变量和静态变量的区别：

   1、成员变量所属于对象。所以也称为实例变量。

      静态变量所属于类。所以也称为类变量。

   2、成员变量存在于堆内存中。

      静态变量存在于方法区中。

   3、成员变量随着对象创建而存在。随着对象被回收而消失。

      静态变量随着类的加载而存在。随着类的消失而消失。

   4、成员变量只能被对象所调用 。

      静态变量可以被对象调用，也可以被类名调用。

   所以，成员变量可以称为对象的特有数据，静态变量称为对象的共享数据。