**成员变量和局部变量的区别**

成员变量：

          1、成员变量定义在类中，方法体之外，在整个类中都可以被访问。

          2、成员变量随着对象的建立而建立，随着对象的消失而消失，存在于对象所在的堆内存中。

          3、成员变量有默认初始化值。

      局部变量：

          1、局部变量只定义在局部范围内，如：函数内，语句内等，只在所属的区域有效。

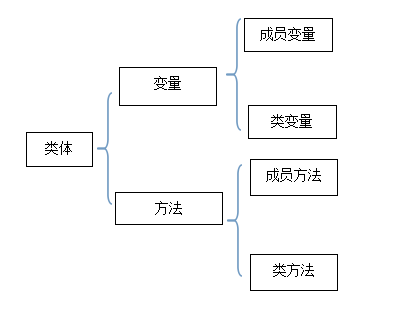
          2、局部变量存在于栈内存中，作用的范围结束，变量空间会自动释放。

          3、局部变量没有默认初始化值

      在使用变量时需要遵循的原则为：就近原则

      首先在局部范围找，有就使用；接着在成员位置找。

**成员变量和类变量的区别**



成员变量：是定义在类中，方法体之外的变量，也称为实例变量。这种变量在创建对象的时候实例化。成员变量可以被类中方法、构造方法和特定类的语句块访问。成员变量只能被对象调用，作用域是在对象中，存在于堆内存中。。

类变量：也声明在类中，方法体之外，但必须声明为static类型。类变量是类创建的时候而存在，随着类的消失而消失的；类变量可以被对象调用，还可以被类名调用，类变量也称静态变量，存在于方法区中。

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。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。在静态方法中访问类的实例变量需首先进行类的实例化

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。