成员变量和局部变量

变量的作用域是指在程序中可以通过变量名来访问该变量的范围，变量的作用域由变量被声明时所在位置决定的，Java 中根据不同的作用域可以将变量分为成员变量和局部变量。

局部变量：如果一个变量在方法中声明，则该变量是局部变量。

成员变量：如果一个变量在方法外，类中声明，则该变量是成员变量。

```
public class HelloWorld{
  int num2 = 2;
  public int test(){
    int num1 = 1;
  }
}
```

1、成员变量和局部变量的区别在于作用域不同，成员变量的作用域在整个类中，类中的每个方法都可以访问该变量，局部变量的作用域只在定义该变量的方法中，出了方法体就无法访问。

2、成员变量和局部变量的初始值也不同，局部变量不会赋初始值，成员变量会赋初始值，具体的值是由成员变量的数据类型决定的。

封装

封装是指将类的属性隐藏在内部，外部不能直接访问和修改，如何实现？通过修改成员变量的可见性，从公有改为私有。

```
public class Student {
  private int id;
  private String name;
  private int age;
  public void show() {
    System.out.println("学生信息如下：");
    System.out.println("学生编号："+id);
    System.out.println("学生姓名："+name);
    System.out.println("学生年龄："+age);
  }
  public int getId() {
    return id;
  }
  public void setId(int id) {
    this.id = id;
  }
  public String getName() {
```

```
    return name;
  }
  public void setName(String name) {
    this.name = name;
  }
  public int getAge() {
    return age;
  }
  public void setAge(int age) {
    if(age <= 0) {
      System.out.println("输入的数值有误！");
      age = 18;
    }
    this.age = age;
  }
}
```

封装的核心思想就是尽可能把属性都隐藏在内部，对外提供方法来访问，我们可以在这些方法中添加逻辑处理来实现过滤，以屏蔽错误数据的赋值。

封装的步骤：

- 修改属性（成员变量）的访问权限为私有，使得外部不能直接访问。
- 提供外部可以直接调用的方法。
- 在该方法中加入对于属性的逻辑控制，避免出现逻辑上的错误。

什么是访问权限?

访问权限是指该属性可以被直接访问的范围，是在属性定义时设定的，访问权限的可选项一共有 4 种：public、private、默认（不写）、protected，区别在于作用域范围不同。

static

static 表示静态或者全局，可以用来修饰成员变量和成员方法以及代码块。

使用 static 修饰的成员变量和成员方法独立于该类的任何一个实例化对象，访问时不依赖于该类的对象，而是直接通过类去访问，可以理解为被该类的所有实例对象所共用，所以说是全局的。

static 还可以修饰代码块，被 static 修饰的代码块叫做静态代码块。

```
static {
  System.out.println(1);
}
```

静态代码块的特点是只执行一次，什么时候执行？当这个类被加载到内存时执行，不需要开发者手动调用，会自动执行。

被加载到内存中的类叫做运行时类，静态代码块就是在家中类的时候执行的，因为类只加载一次，所以静态代码块也只执行一次。