局部变量：定义在方法之中的变量。
局部变量要先赋值，再进行运算，而实例变量均已经赋初值。这是局部变量和实例变量的一大区别。
实例变量的对象赋值为 **null**。
局部变量不允许范围内定义两个同名变量。实例变量的作用域在本类中完全有效，当被其他的类调用的时候也可能有效。
实例变量和局部变量允许命名冲突。

书写方法的格式：

修饰符	返回值	方法名	调用过程中可能出现的例外	方法体
public	**int/void**	**addNumber**(参数)	**throw Excepion**	**{}**

例：
```
public   int   addNumber(int a,int b){
}
```
注：方法名中的参数 **int a,int b** 为局部变量

类方法中的一类特殊方法：构造方法。
构造方法是当用类生成对象时，系统在生成对象的过程中利用的方法。
注意：构造方法在生成对象的时候会被调用，但并不是构造方法生成了对象。
构造方法没有返回值。格式为：**public** 方法名。
构造方法的方法名与类名相同。
构造方法是在对象生成的过程中自动调用，不可能利用指令去调用。
在一个对象的生成周期中构造方法只用一次，一旦这个对象生成，那么这个构造方法失效。

用类来生成对象的语句：
Student   s=new Student()。
第一个 **Student** 表示这是用 **Student** 类进行定义。“**Student()**”表示调用一个无参数的构造方法。
如果()中有参数，则系统构造对象的过程中调用有参的方法。
此时 **S** 称为一个对象变量。
Student   s 的存储区域存放的是地址：一个对象在硬盘上占有一个连续地址，首地址赋予 **s** 空间。
S 称为对象 **Student** 的引用。
注意：在对象变量中存放的是引用（地址）；在简单变量中存放的是数值。

可以构造多个构造方法，但多个构造方法的参数表一定不同，参数顺序不同即属于不同的构造方法：
```
public student(string name,int a){
}
public student(int a,string name){
}
```
为两个不同的构造方法。
如果我们未给系统提供一个构造方法，那么系统会自动提供一个为空的构造方法。
练习：写一个类，定义一个对象，定义两个构造方法：一个有参，一个无参。
（编写一个程序验证对象的传递的值为地址）
注意下面这种形式：
static void changename(student stu){stu.setName “LUCY”}
注意生成新的对象与旧对象指向无关，生成新对象生命消亡与旧对象无关。

面向对象方法的重载（**overloading**）和覆盖（**overriding**）。
在有些 **JAVA** 书籍中将 **overriding** 称为重载，**overloading** 称为过载。