[字符串拼接问题 2](#_Toc527727412)

[循环条件中使用++运算符问题 2](#_Toc527727413)

[位运算符(了解) 3](#_Toc527727414)

[按位与 3](#_Toc527727415)

[按位或 4](#_Toc527727416)

[按位异或 4](#_Toc527727417)

[按位取反 4](#_Toc527727418)

[按位右移 5](#_Toc527727419)

[按位左移 5](#_Toc527727420)

[无符号右移 5](#_Toc527727421)

[运算符的优先级和结合性 6](#_Toc527727422)

[参数的分类 7](#_Toc527727423)

[数据类型选择时的原则 7](#_Toc527727424)

[类和对象 8](#_Toc527727425)

[类的概念 9](#_Toc527727426)

[对象的概念 10](#_Toc527727427)

[对象的创建方式 11](#_Toc527727428)

[类和对象的关系 12](#_Toc527727429)

[访问修饰符 12](#_Toc527727430)

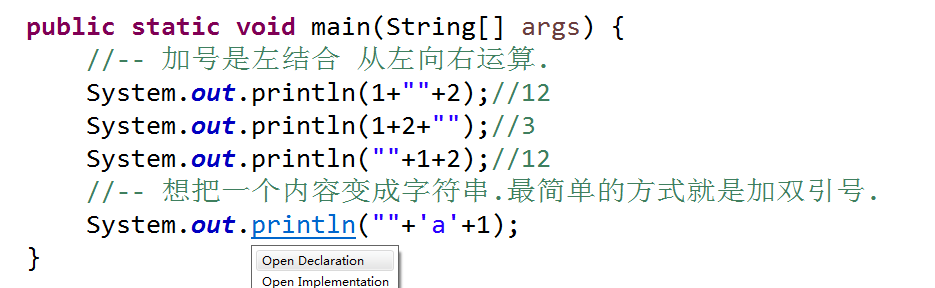
[包(Package) 12](#_Toc527727431)

[变量的不同叫法与对应的英文 13](#_Toc527727432)

[课堂练习 14](#_Toc527727433)

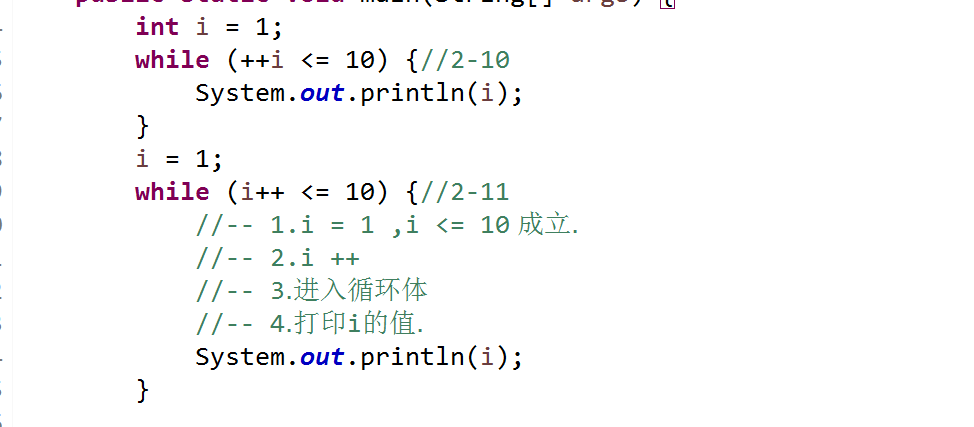
# 字符串拼接问题

任何类型都可以使用+””来转变成字符串类型.字符串遇到加号.这里的加号起到的作用是拼接.所以需要注意位置.



# 循环条件中使用++运算符问题

在循环条件里面使用++也分前后缀的



++如果在前是先自加一,然后在判断条件是否成立.如果++是在后面的,先判断条件是否成立,然后再自加一.

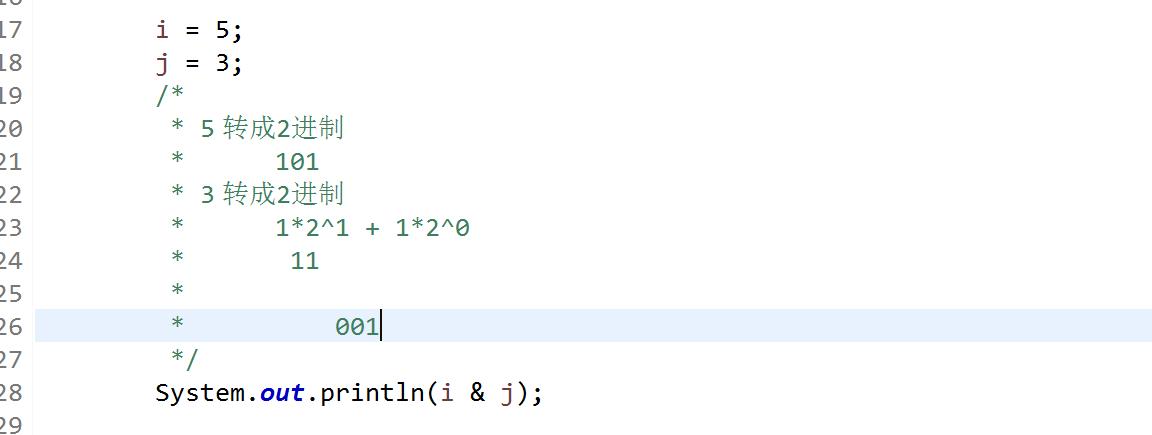
# 位运算符(了解)

位运算符是操作位的.常见的有以下几个

* 按位与
* 按位或
* 按位异或
* 按位非
* 向右移动
* 向左移动
* 无符号向右移动.

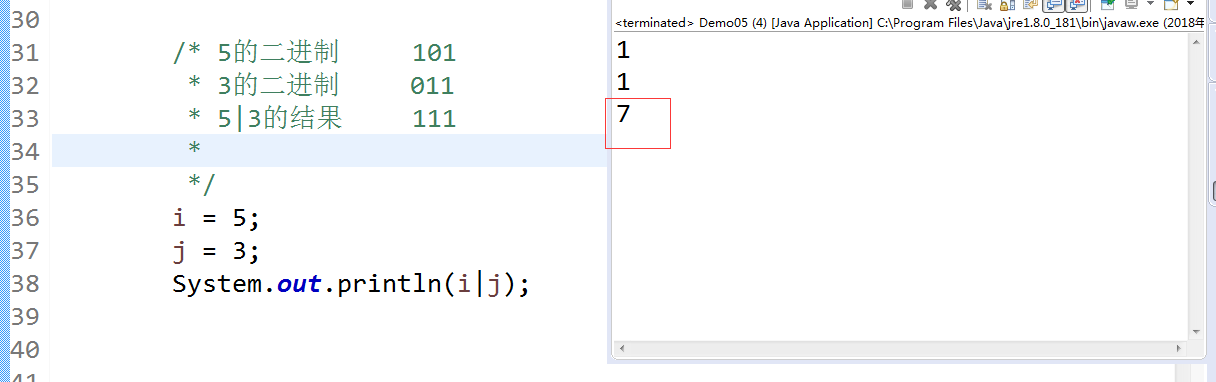
## 按位与

规则:仅当两个操作数都为1的时候,输出结果才是1,否则为0.



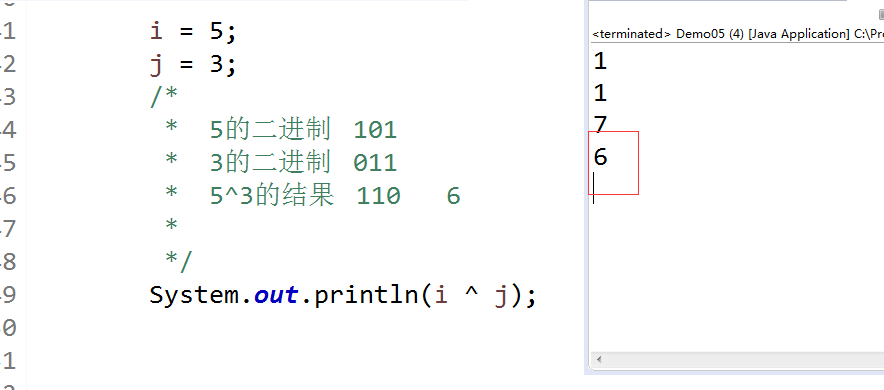
## 按位或

规则:当且仅当两个数都为0时,才为0.



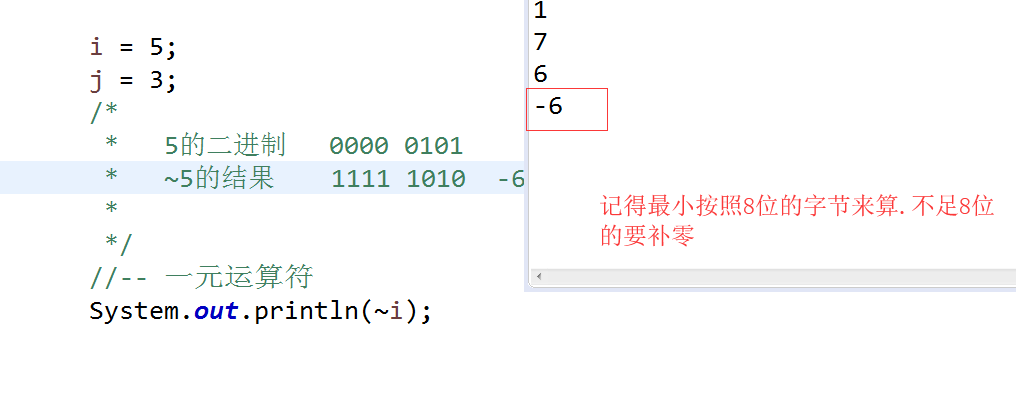
## 按位异或

规则:两个操作数不相同的时候输出1.否则为0.



## 按位取反

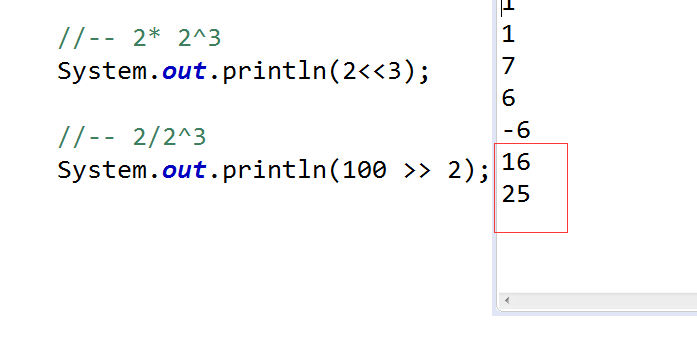
规则:就是0变成1,1变成0

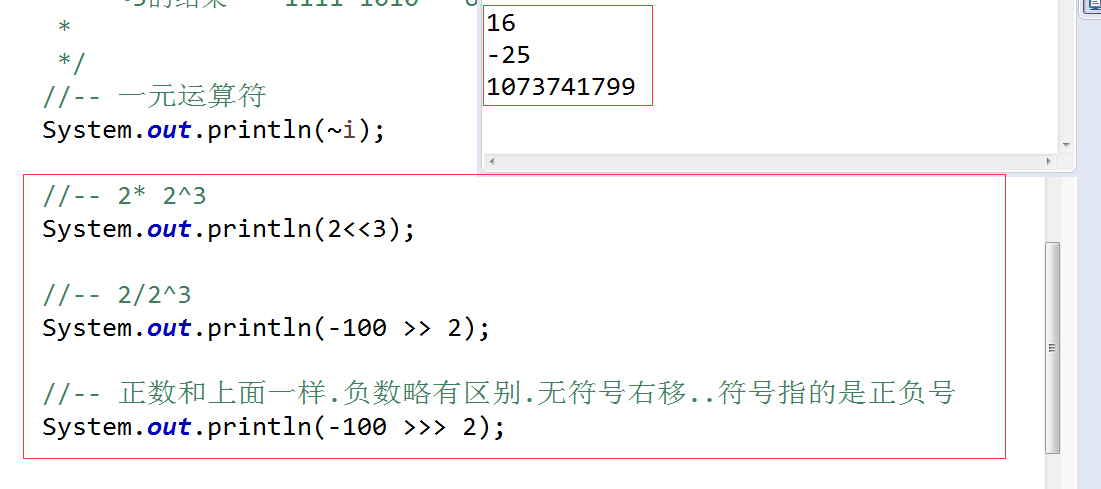


## 按位右移

## 按位左移

## 无符号右移





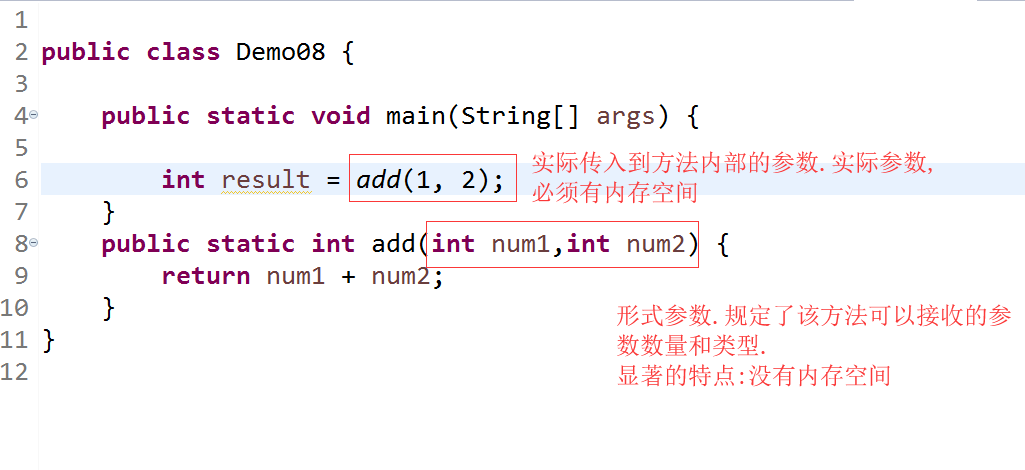
# 运算符的优先级和结合性



# 参数的分类

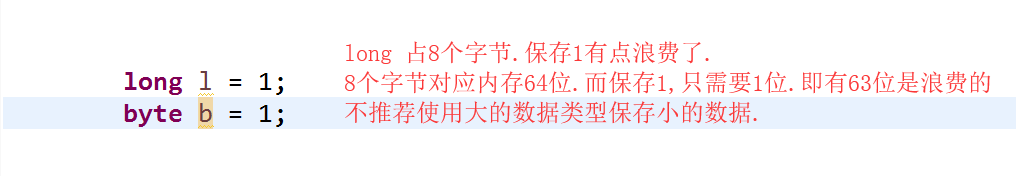
参数按照位置可以分为两种

* 形参:形式参数
  + 在方法声明时叫做形式参数
* 实参:实际参数
  + 在方法调用时叫做实际参数



# 数据类型选择时的原则

在选择数据类型时,我们参考最小原则即够用就好.没有必须要使用较大的类型来保存小的数据.比如



# 类和对象

面向对象:Object Oriented简称OO.一般有一下几个分类

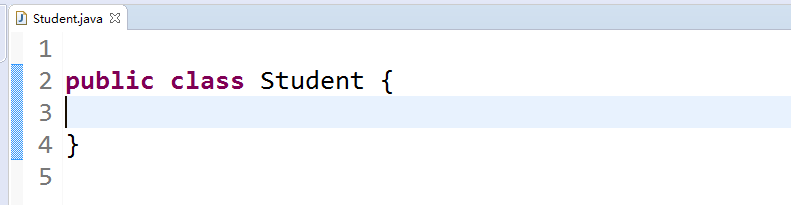
* OOP:面向对象编程
* OOL:面向对象语言
* OOD:面向对象设计
* OOA:面向对象分析

面向对象是一种对现实世界进行理解和抽象的方法.我们所编写的程序是为了解决现实生活中的问题.我们需要通过代码来构建一种对现实生活的映射.在这个映射关系中,每一个现实生活中实际存在的个体,就是一个对象.

我们在描述现实生活时会发现一个问题.有很多事物或者内容或者动作,我们用原生数据类型是无法进行描述的.

比如说用原生数据类型来描述:人..我们用原生数据类型可以描述人的静态特征.比如身高体重性别.但是当遇到行为的时候,就没有任何办法.并且每个人都会有上面所说的静态特征.

针对这个问题.Java采用了C++中的一个关键字:class.用它来设计概念.即创建新的数据类型.这个数据类型就是它说描述的一类对象(实际存在的事物)的概括.



如上所示.我们创建了一个新的类型:Student类型.它代表所有Student实体(Student类型的对象)的共性.可以理解为像制造业的设计图一样.设计图中涵盖了所设计事物的所有特征.但是它是不可以实际使用的.比如建筑图是不可以住人的.汽车设计图是不可以再马路上行驶的.只有把这些图所代表的事物实际的制作出来才可以被实际使用.

## 类的概念

每一个class,都是一个新的数据类型.是现实生活中一类实体的所有共性.

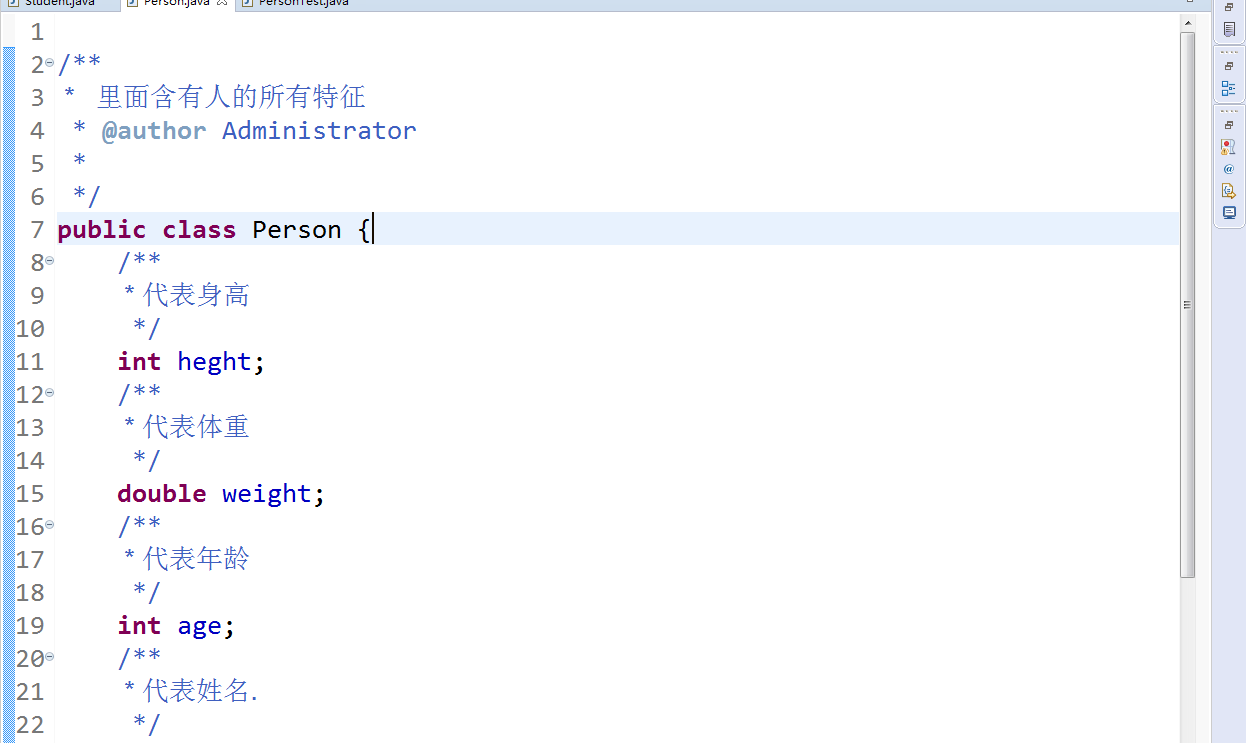
共性可以分为两方面(以人为例)

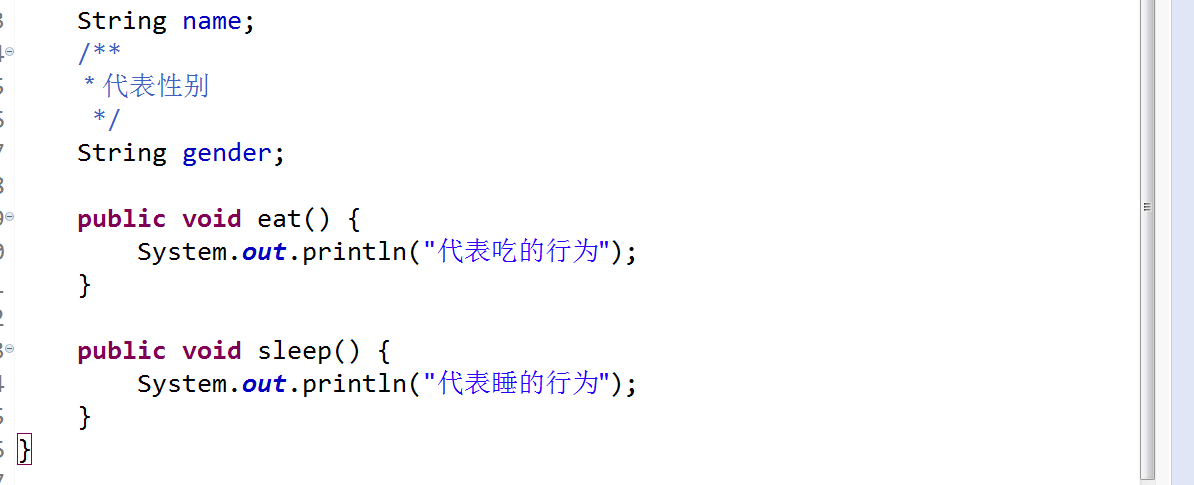
* 静态特征
  + 身高
  + 体重
  + 年龄
* 动态行为
  + 吃
  + 睡
  + 玩

在代码中我们通过两个内容来反应行为和特征

* 字段(一般使用英文Field(域).非局部变量)
  + 反应的特征.
* 方法
  + 反应的是行为.

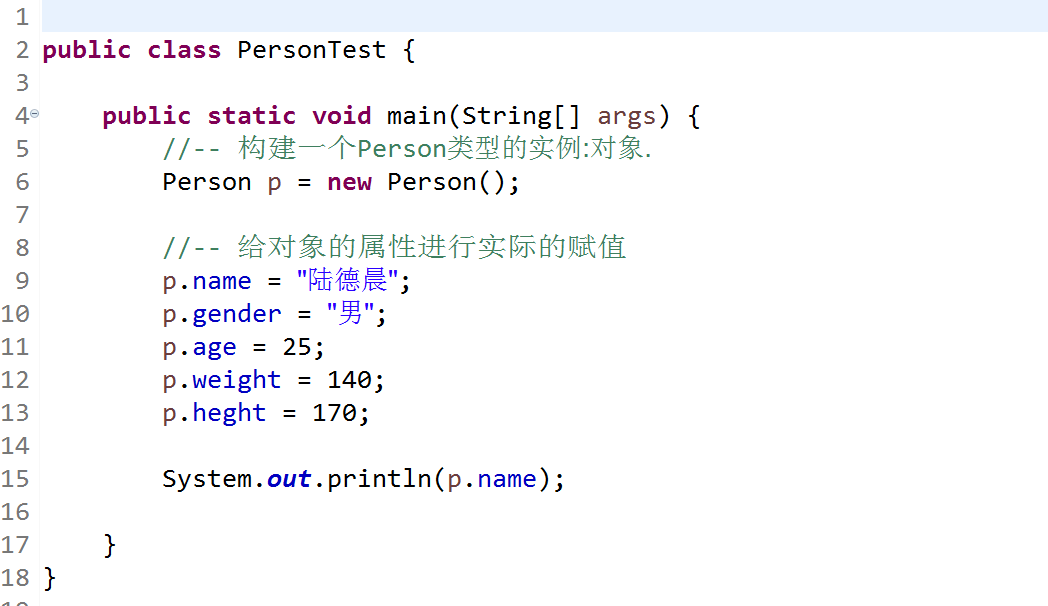
设计一个Person类型.代码如下





## 对象的概念

类是一个抽象的概念,是不可以被使用的.可以使用的是类的实例.对象.



### 对象的创建方式

在Java中创建对象就是创建一个实例,目的是为了映射现实生活中具体存在的一个个体.

因为类是概念组合是一个抽象的事物.不可以被使用的,我们能够使用的只有对象.

Java中创建对象的方式有以下几种

* 使用new关键字来创建
* 使用克隆来创建
* 使用反序列化来创建
* 使用反射来创建.

这里面使用频率最高的是第一种,也是必须掌握的一种.

new 一个对象的过程也被我们叫做是创建实例的过程或者是实例化的过程.

### 类和对象的关系

类是一类具有共性行 为和特征的对象的抽象.

对象是类的实例化.

# 访问修饰符

Java中访问修饰符有四种

|  |  |
| --- | --- |
| 访问修饰符 | 名称 |
| Public | 公共的(所有人都可以访问) |
| Privated | 私有的(只有自己可以访问) |
| Protected | 受保护的 |
|  | 默认(什么也不写)等同于包权限.同包可以访问 |

## 包(Package)

用来分类管理我们所建立的class文件.包的命名一般使用域名反转.

因为在一个项目中可能会遇到同名文件.虽然名字一样但实际的功能可能是完全不一样的.为了区分我们就会把这两个文件分别放在不同的包中.如同Window下.建立不同的文件夹里面都放一个新建文本文档.txt一样.

一般包名都是唯一的.一般域名都是唯一的.所以我们在命名包的时候会使用域名反转.(公司有自己的一套命名规则.)

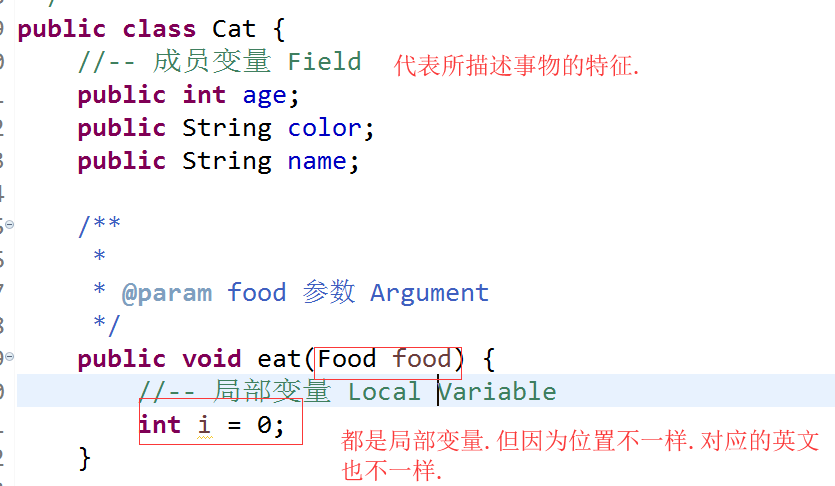
包的路径一般写三级:com.XXX.YYY.具体的class文件.

这里的com和XXX和YYY都会在本地创建实际的文件夹

# 变量的不同叫法与对应的英文

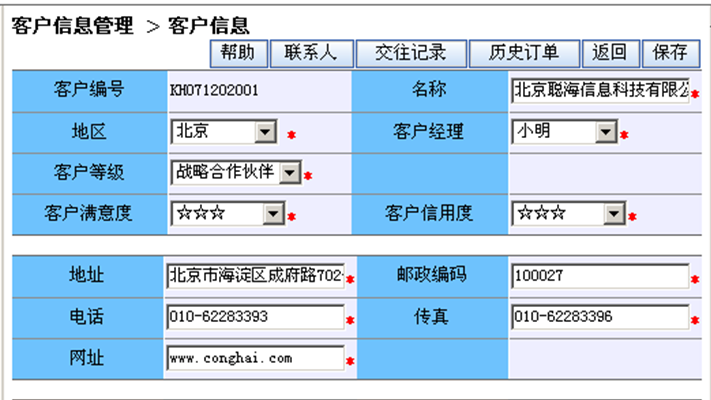
根据位置不同的.变量有不同的叫法

* 常量
  + 标准就是看有无final
* 成员变量
  + 写在方法外面的变量.作用域是整个class
* 局部变量
  + 写在某个方法中的变量.作用域就是当前方法.
* 方法参数
  + 写在方法名字后面的小括号中的.



# 课堂练习

1. 建立客户类
   1. 编号的取值:KH + 当前日期的月和日(不足两位数的补零) + 四位随机数.



1. 提供一个数组,数组中的元素类型是客户类型..
   1. 向数组中插入10条客户信息
   2. 遍历整个数组中所有客户的信息.

# 家庭作业

1. 接课堂练习.提供如下几个功能(6道题)
   1. 插入一个客户
      1. 顺序插入
      2. 指定位置插入
   2. 删除一个客户
      1. 指定位置删除
   3. 按照客户等级降序排列
   4. 按照客户信用度降序排列
   5. 汇总不同区域客户数量

# 预习

1. 封装
   1. 封装的含义
   2. 封装的写法
   3. 封装的好处
2. 继承
   1. 继承的概念
   2. 继承的写法
   3. 继承的优缺点
3. 下载一款软件:starUML.学习它的使用方式