# 1.Java历史

背景：

J2SE(Java基础版 适合做C/S程序) ，AWT、和Swing，画界面 --->淘汰

重点是打好基础。

J2EE(Java企业版 WEB开发，适合做B/S程序)，主流

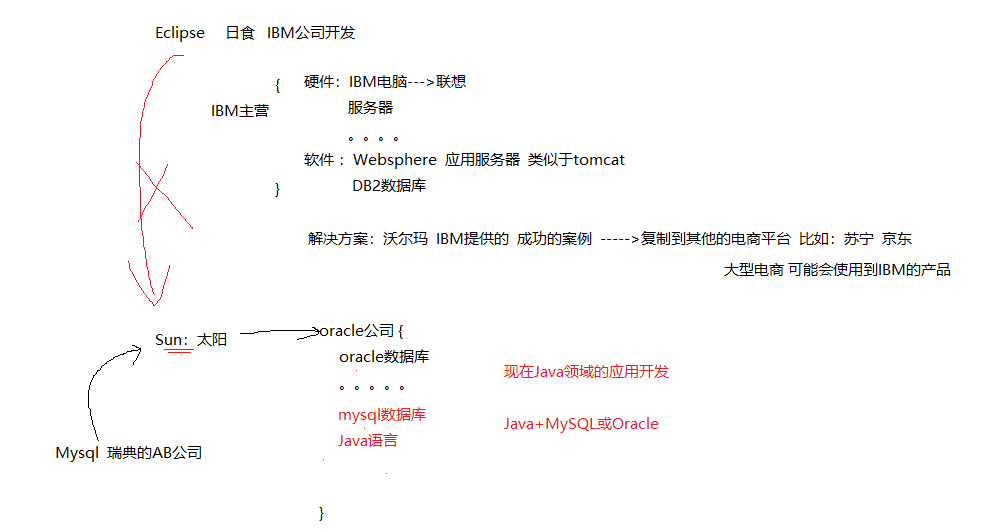
J2ME(移动端)：诺基亚年代 塞班系统大量的使用了J2ME开发

--->安卓(底层Java) IOS(底层是C)

J2ME转换做安卓

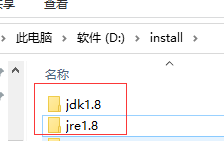
C---->IOS

Java属于Sun公司 ---->Sun被Oracle收购

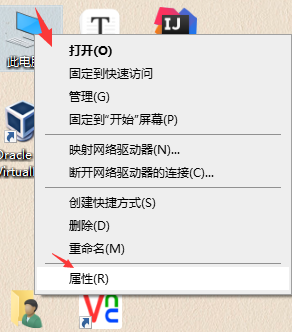


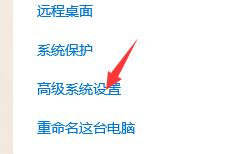
# 2.Java环境搭建

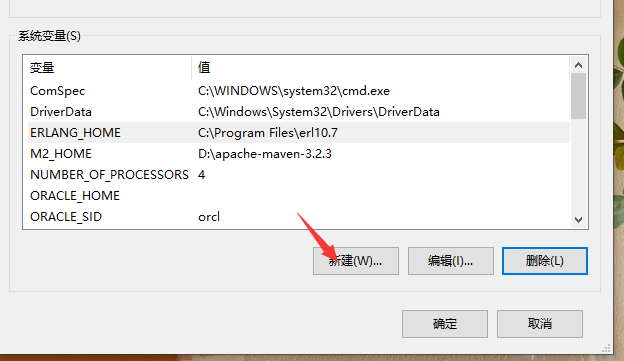
安装Jdk 最低版本1.8

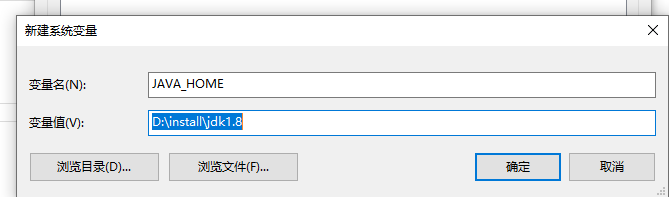


配置环境变量

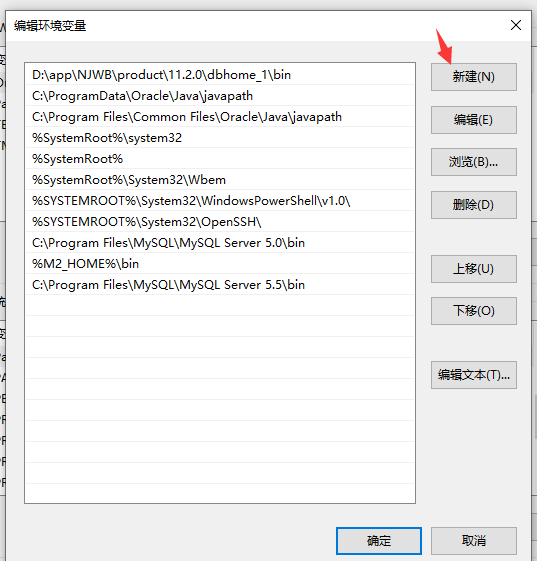


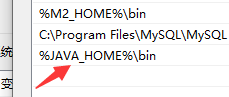






将JAVA\_HOME环境变量加入到Path路径中





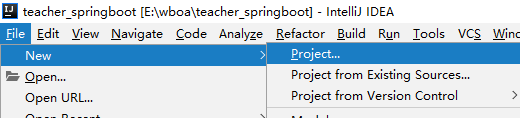
打开dos窗口，验证JDK

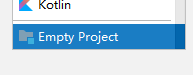
win + R

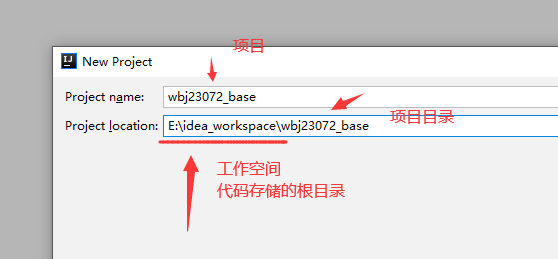


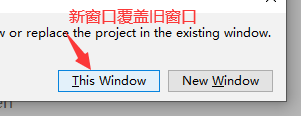
安装idea ---集成开发工具

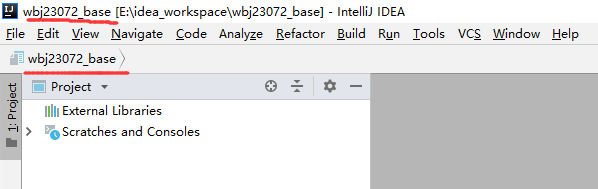
idea的使用



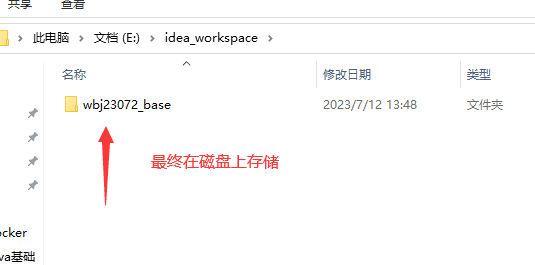




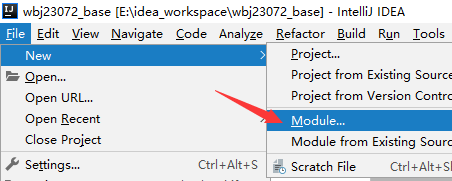


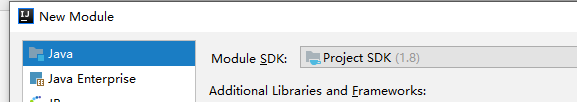


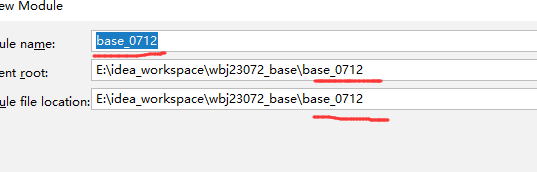
观察磁盘目录

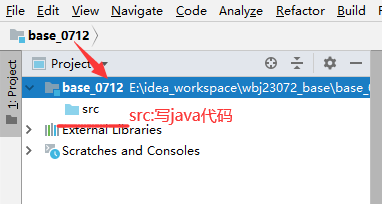


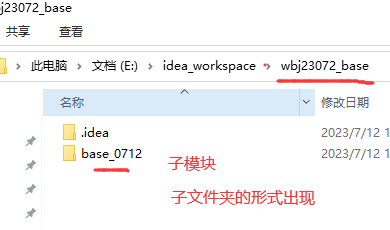
将来在wbj23072\_base项目中可能会创建多个子项目 ---->子模块

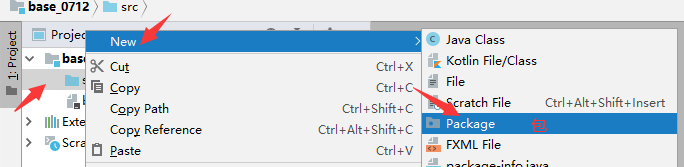


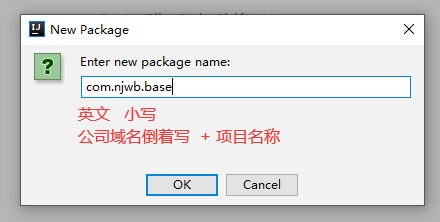


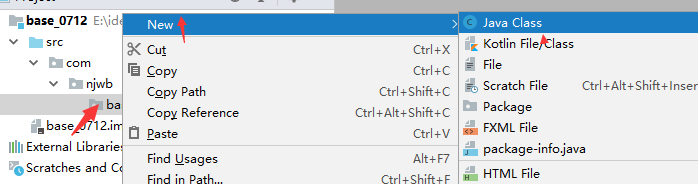


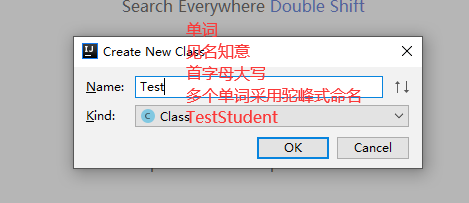


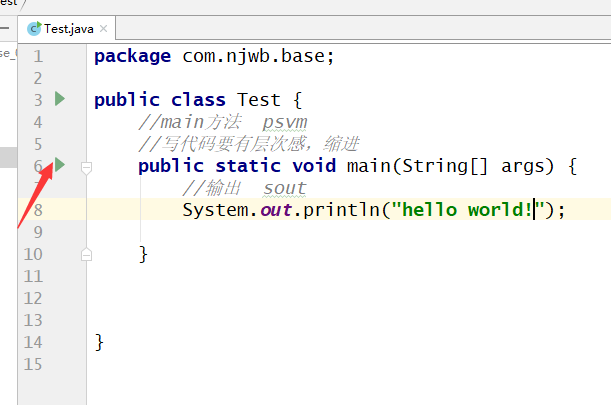


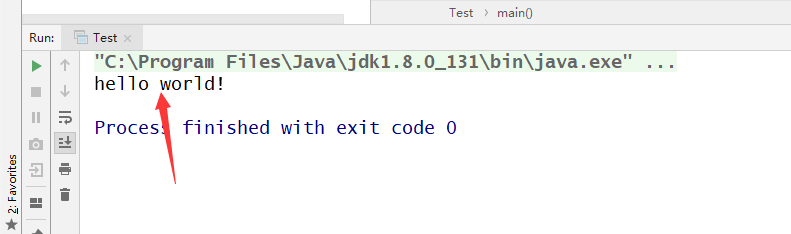


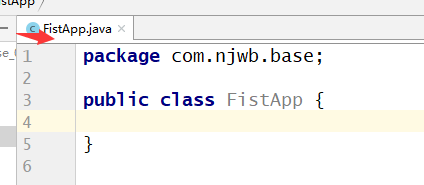










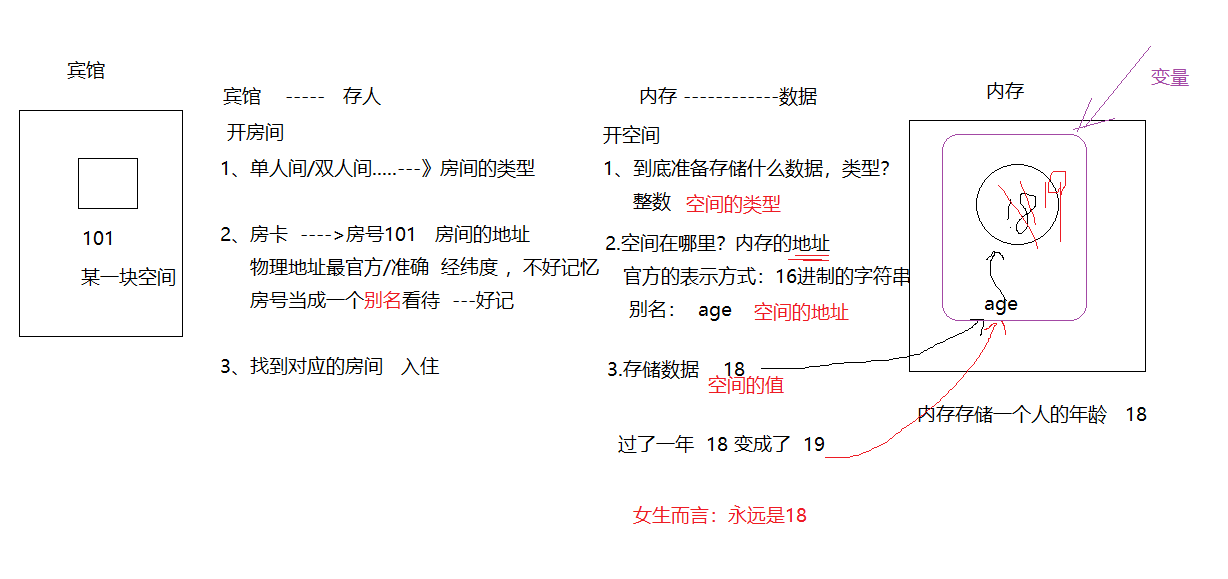


# 3.回顾基础知识

## 3.1 变量

什么是变量？

程序中的数据保存在内存中，内存是如何存储数据？抽象的概念 ---->具体化--->生活中靠



程序中使用专业的名词：变量

变量就是内存中的一块空间表示，包含了三个要素：变量类型、变量名、变量值

不变的量也叫变量，只是变量的一种特殊形态-----常量。

## 3.2 数据类型

Java将数据类型分为两大类

1、基本数据类型

8大基本类型

boolean:布尔 true false 通常和判断语句

整数:

byte(1个字节)、short(2个)、int(4个)、long(8个)

小数\浮点数

float(4个)、double(8个) 可以使用科学计数法

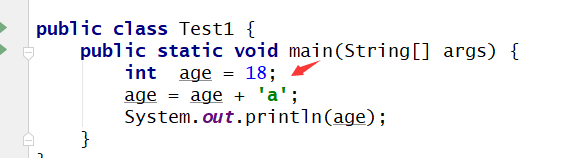
char ：单个字符 ‘A’ ‘a’，在内存中如何存储？ASCII码值

‘0’ ------------ 48

‘A’ ------------- 65

‘a’ ------------- 97

考试：程序片段题 ---->答案是多少



程序题：

给定一个任意长度的字符串,如：’hello1wo2r90lhehe66xixi’

统计非数字的字符个数或者统计数字的个数。

思路：利用ASCII码值

2、引用数据类型

String 字符串

## 3.3 变量名

规范

1、必须是英文单词

2、必须见名知意 --- 配合字典，切忌使用拼音

3、首字母必须小写，多个单词采用驼峰式命名

name age studentName teacherName

4、不要使用关键字

## 3.4 变量的使用



## 3.5 动态赋值

主要谈的就是从控制台(键盘)接收一个数据，使用变量来保存

利用Scanner



## 3.6 注释

注释很重要，提高代码的可读性

一般来说，注释占比 20%-30%

分类三类

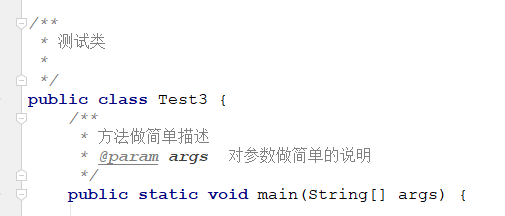
(1)类注释，可以使用在类上，属性上，方法上

/\*\*

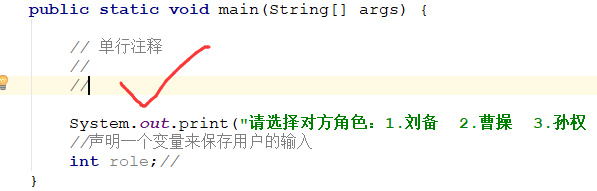
\*

\*

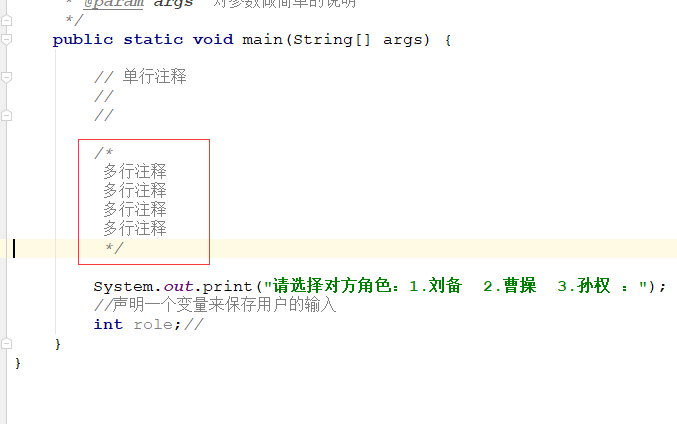
\*/



(2)单行注释 ，主要写在方法的内部



(3)多行注释 ，用在方法内部

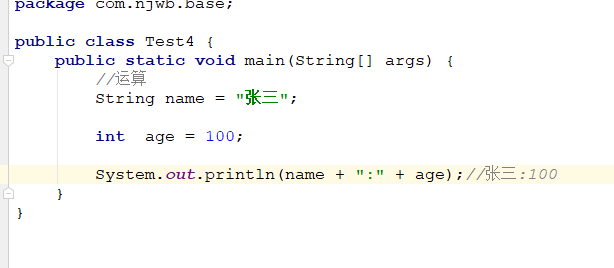


## 3.7 运算符

(1)赋值运算

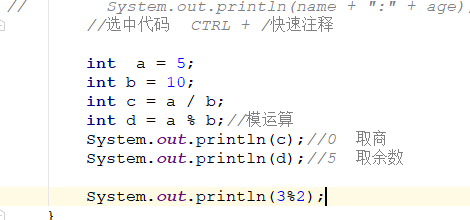
变量 = 值;

(2)字符串连接运算



(3)四则运算

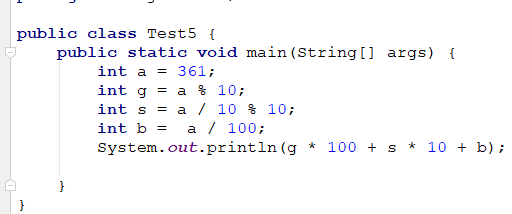
+ - \* % /



如果想得到一个精确值？将被除数提升到double类型



给定任意一个三位数 如361 ,输出 163



(4)自增自减

i++ ----> i + 1

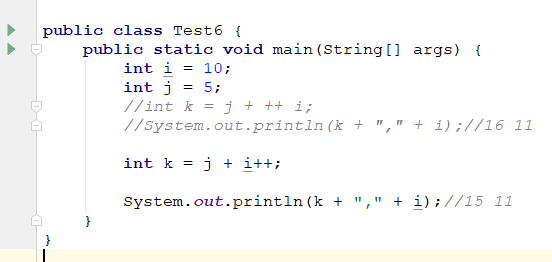
++i ---> i + 1

i-- ----> i - 1

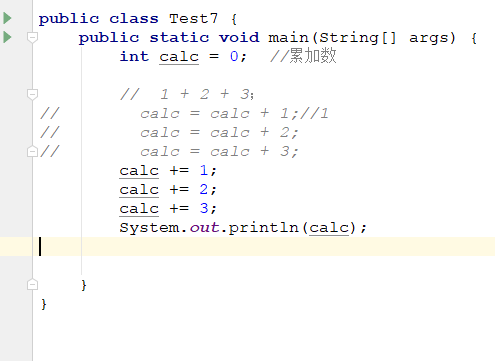
--i ----> i - 1

++ 或 -- 在前 ： 先自增或自减 然后再运算

在后：先参与其他运算再自增或自减



(5)复合运算



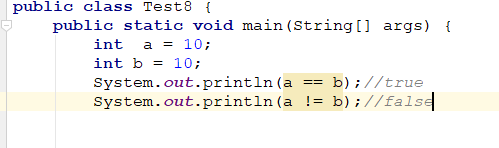
cal += -= \*= /= %=

(6)关系运算

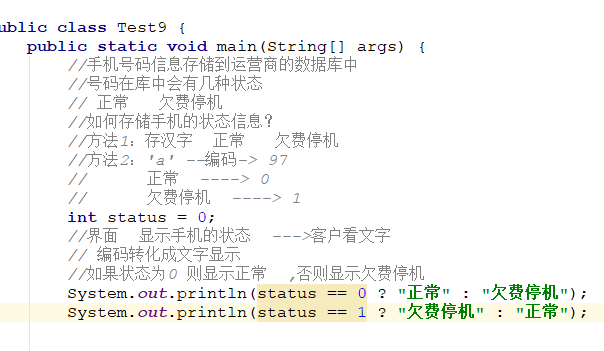
> >= < <= == !=

x > y 要么成立 要么不成立 ---->boolean类型

两个值参与计算(两个操作数) ---->双目运算 --->结果是一个布尔类型



(7)条件表达式



条件 ? 值1 : 值2

条件成立了，则取值1，否则取值2

(8)逻辑运算

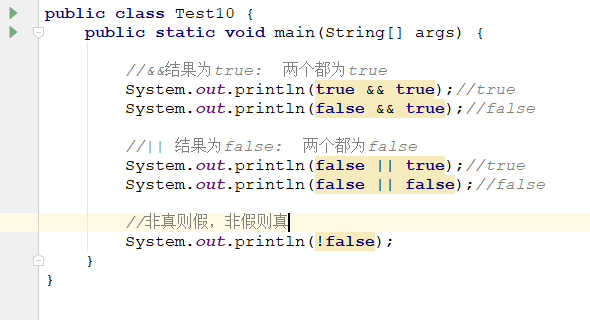
与 && (并且，交集) --->双目

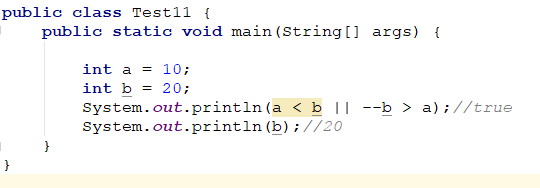
或 || (并集) --->双目

非 !(取反) ---->单目

操作数是boolean类型

逻辑运算结果 --->boolean类型



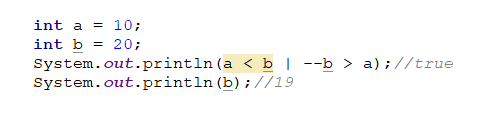


分析：

(1) a < b 成立 true

(2)考虑到是或运算 || 只要有一个成立，则整个表达式为true ,

|| 短路或，右侧的计算忽略

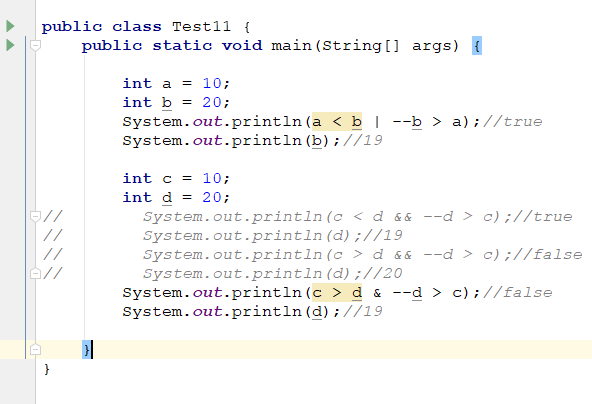


分析

(1) a < b 成立 true

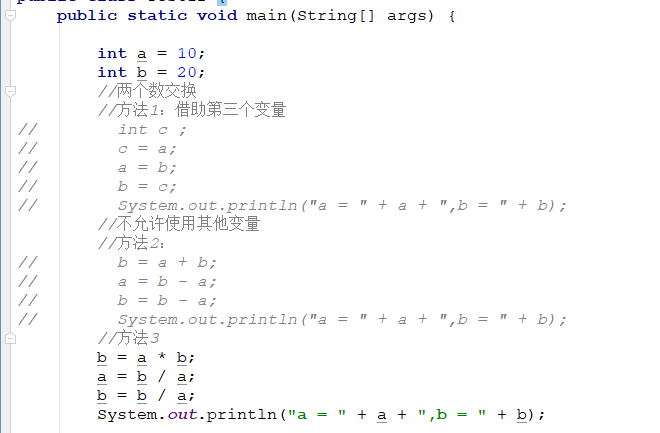
(2) 考虑到是或运算 | 只要有一个成立，则整个表达式为true

(3)但是 | 是一个非短路或，所有右侧仍然要计算

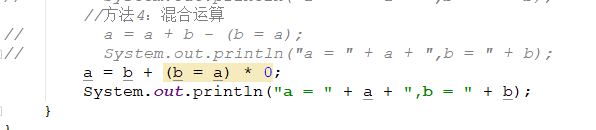


随堂练习：

给定两个整数，int a = 10 int b = 20 a和b做交换



提示：混合运算有优先级 ，其中赋值运算的优先级最低。



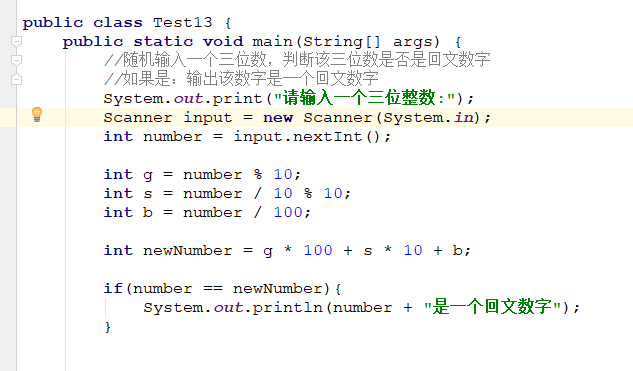
## 3.8 分支结构

if(条件){

//逻辑代码

}

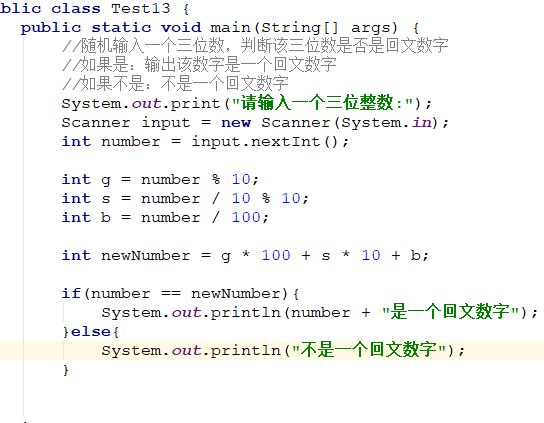
不管逻辑代码有多少，{}不要省略



if(条件){

}else{//条件不满足

}



多重if

if(){

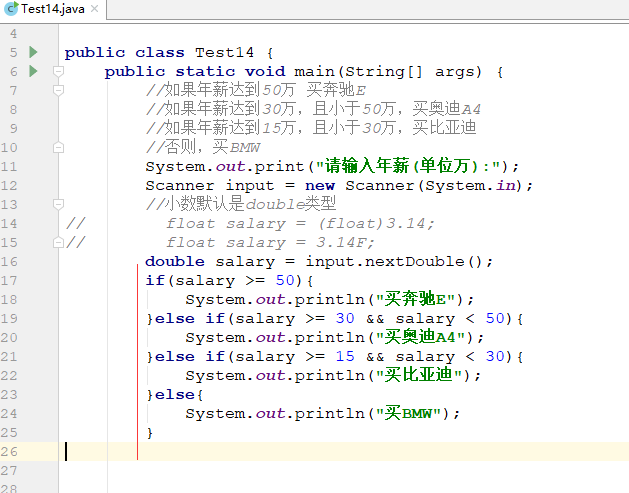
}else if(){

}else if(){

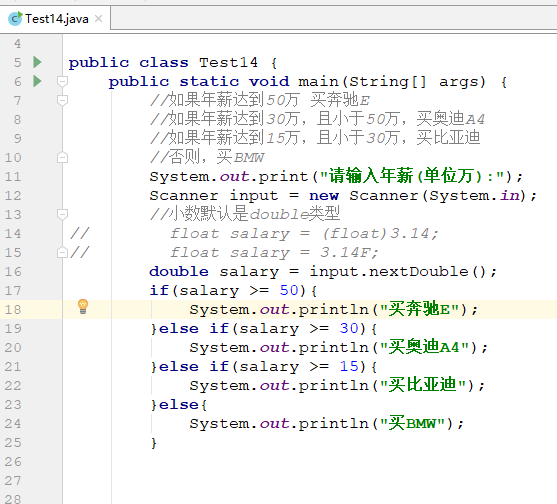
}…..

{

}



稍微改造：



多重if，记住：条件的写法，要么从大到小写，要么从小到大写

需求：从控制台接收一个整数模拟当天是周几？范围是1-7

如果今天是周一、周二，学linux

如果今天是周三、四、五，学Java

如果今天是周六，复习

如果今天是周末，放松休息。

使用多重if完成

int day = x

if(day == 1 || day == 2){

}else if(day == 3 || day == 4 || day == 5 ){

}else if(day == 6){

}else{

}

代码省略

对比上述两个多重if的案例：

连续区间 ： 使用多重if比较合适

离散区间(等值判断) ： 最好使用switch语法

switch(值){

case 值1:

//业务代码

break;

case 值2:

//业务代码

break;

…………………………

case 值N:

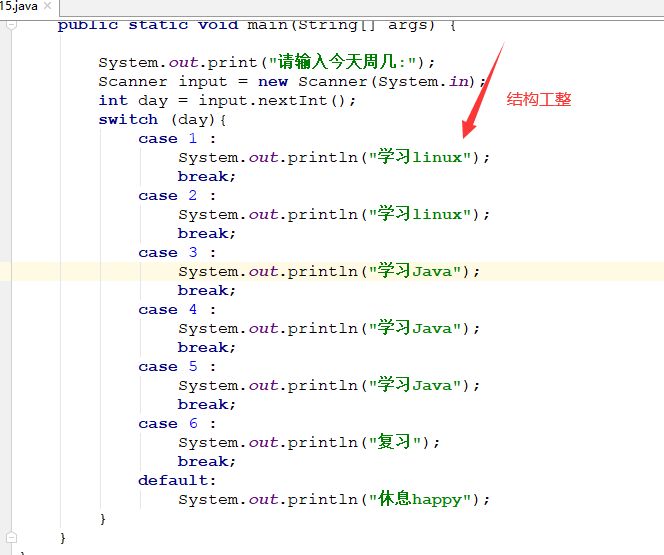
//业务代码

break;

default:

//…..

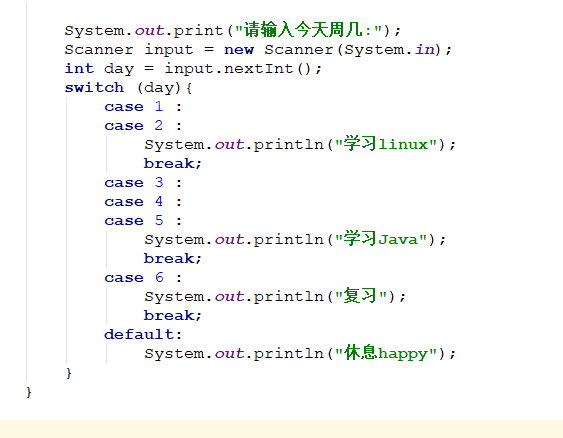
}



说明：

(1)理论上每个case中都要有break,用于退出switch结构

(2)当多个case中含有相同业务逻辑块时，可以省略break,进行合并，去除冗余代码



(3)default的顺序有没有要求？ 没有要求，可以写到前面也可以写到最后

default中没有break行不行？{

如果default语句在最后，可以省略break

如果放在首行，必须加上break

}

(4)switch中的条件变量可以是哪些数据类型？

String

char

byte

short

int

## 3.9 循环

### 3.9.1 循环的定义

现实生活：绕操场跑5圈，活到老学到老

循环：有条件的重复做某件事

条件 -----> 循环条件，不是无止境 ，有退出的那刻。

重复做的事情 ----> 循环体

### 3.9.2 while循环

语法：

while(循环的条件){

循环体

}

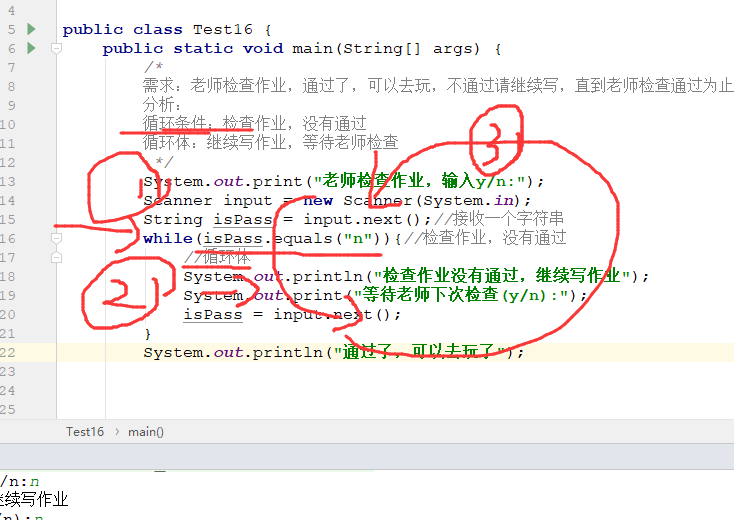
需求：老师检查作业，通过了，可以去玩，不通过请继续写，直到老师检查通过为止。

分析：

循环条件：检查作业，没有通过

循环体：继续写作业，等待老师检查

利用语法翻译代码



while循环的过程：

(1)先判断循环条件是否成立，如果成立执行(2)

(2)执行循环体

(3)再判断循环条件

…….

### 3.9.3 do while循环

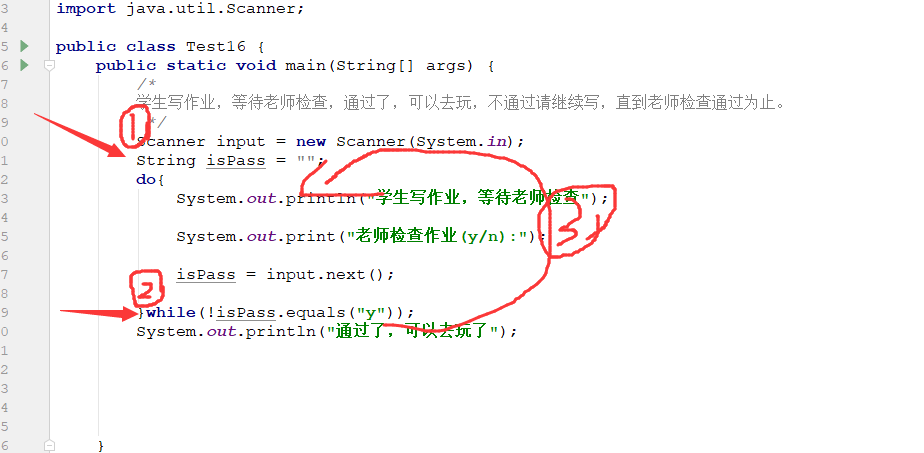
语法：

do{

循环体

}while(循环条件)**;**

需求：学生写作业，等待老师检查，通过了，可以去玩，不通过请继续写，直到老师检查通过为止。



do while循环的过程：

(1)先执行循环体，然后执行循环条件

(2)循环条件成立后，再执行循环体

比较：

while和do while的区别：

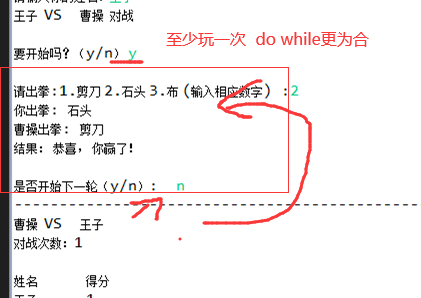
共性：处理循环

不同点：

(1)语法不一样

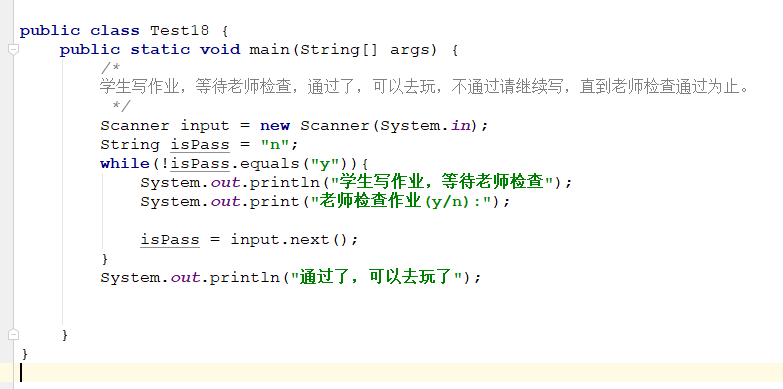
(2)执行的过程 while是先判断再执行 do while是先执行再判断

(3)while的循环体执行0次或者多次 do while循环体至少执行1次



当然：while和do while是可以相互转换的

使用while实现和do while同样的效果



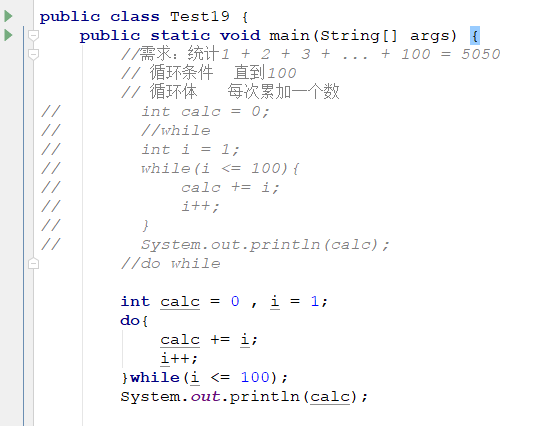
### 3.9.4 for循环

语法结构

for( 初始化变量 ; 循环条件 ; 循环变量的变化 ){

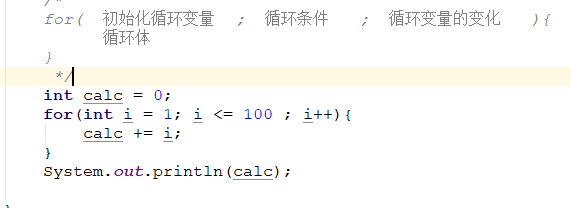
循环体

}





当循环次数固定时，可以使用for循环



说明

(1)for循环的执行过程

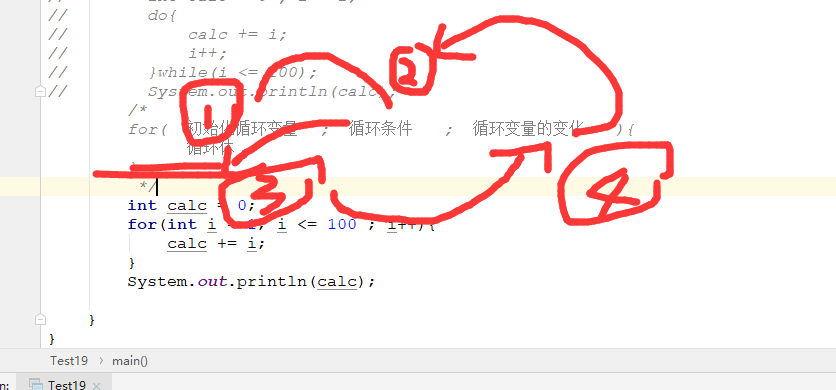
a、执行初始化部分

b、判断循环条件

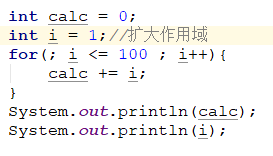
c、循环体

d、循环变量的变化

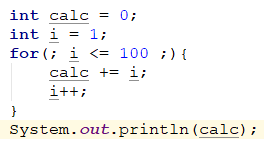
e、判断循环条件



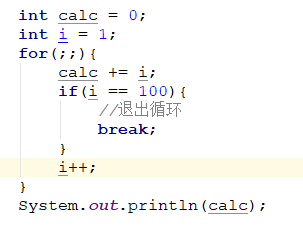
(2)初始化部分可以省略，目的就是为了扩大作用域



(3)循环变量的变化部分可以省略



(4)循环条件可以省略



作业：

a、复习

b、可以扩展的学习

c、输出100-1000之内的所有水仙花数

从控制台输入一个整数，表示第n 项斐波拉契数列 ，求第N项的值是多少？

d、重写猜拳

### 3.9.5 break和continue

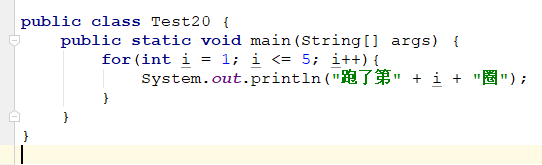
break:可以用在switch或循环中，退出结构

continue:

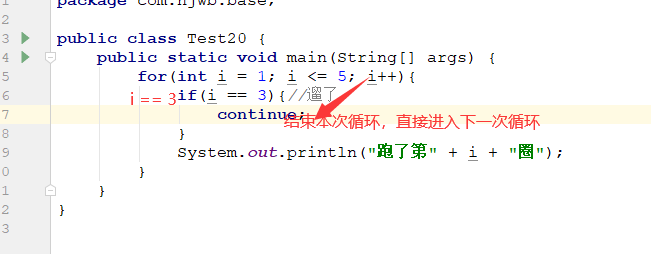
例子：上体育课，先跑5圈

三类人：

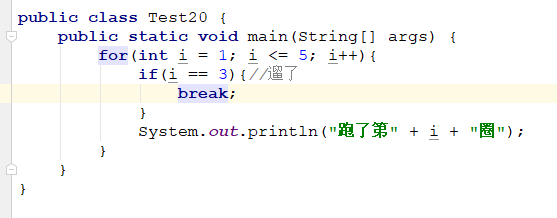
a、老老实实跑完



b、跑了一些，中途遛了，比如第三圈溜了，从四圈继续跑



c、跑了一些，中途遛了，后面没人了。就跑完两圈

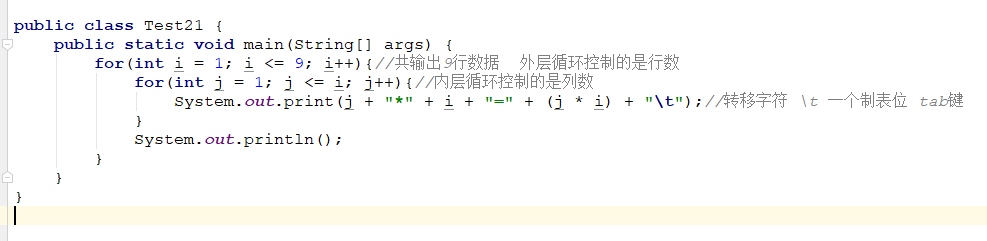


### 3.9.6 二重循环

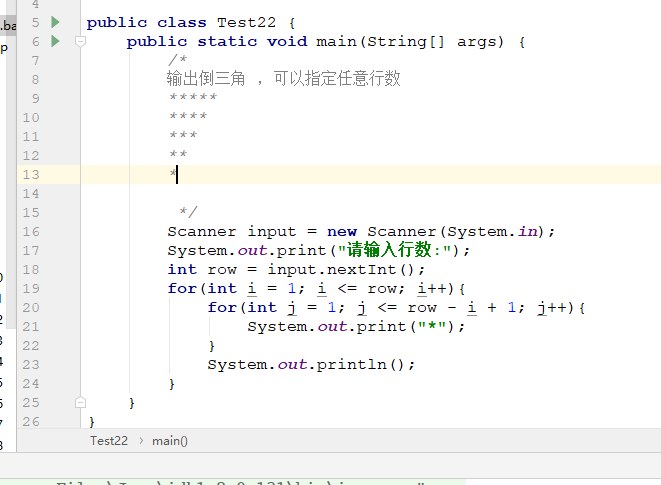
循环的嵌套

考题中

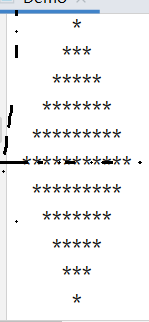
打印乘法口诀表



特殊图形



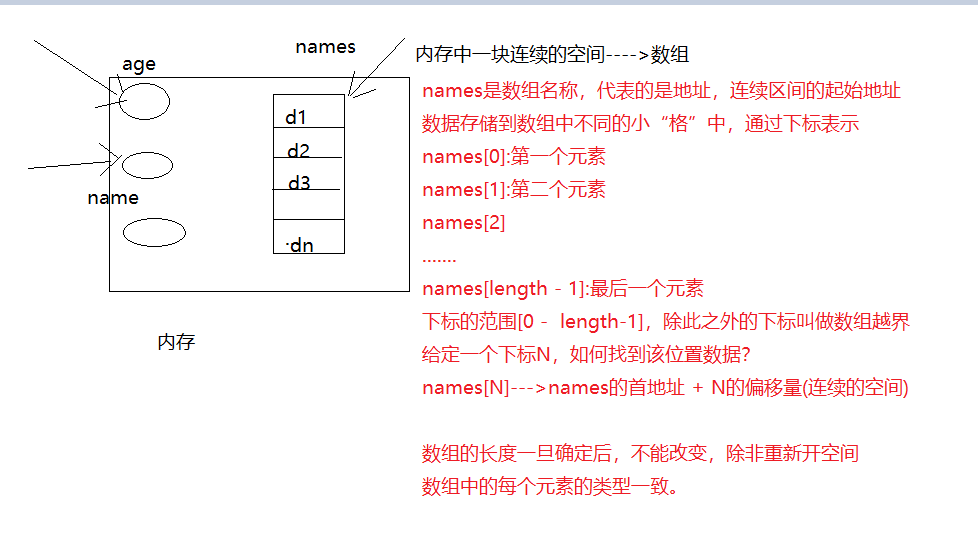
课后去多找几个特殊图形，关键：(1)外层循环是什么 (2)内层循环又是什么 (3) i 和j之间的关系



二重循环还会通常使用在排序中

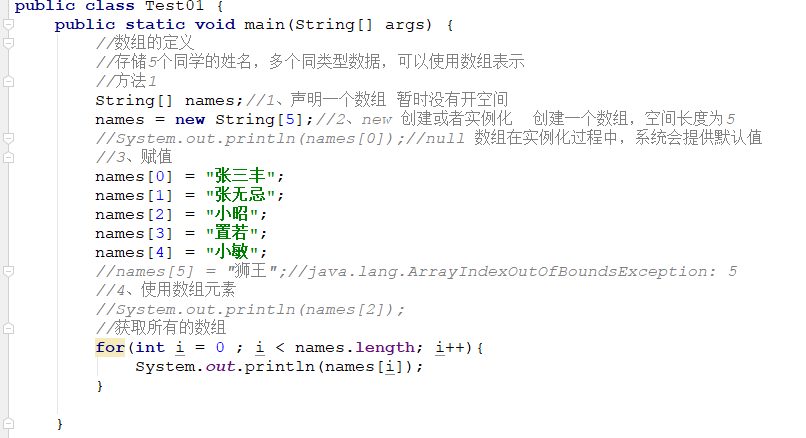
## 3.10 数组

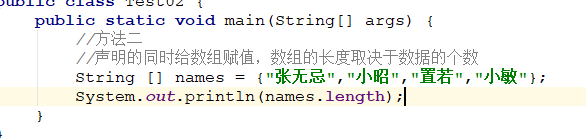
### 3.10.1 定义

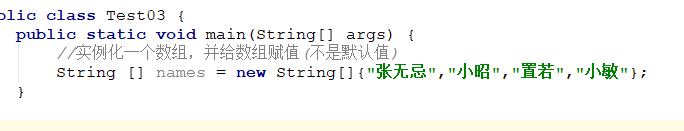


数组：一组连续的区间

### 3.10.2 数组的定义

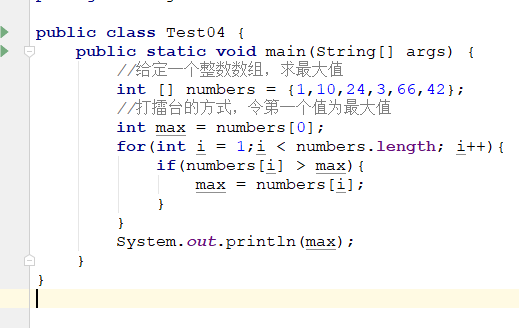






### 3.10.3 数组的应用

(1)给定一个整数数组，求最大值

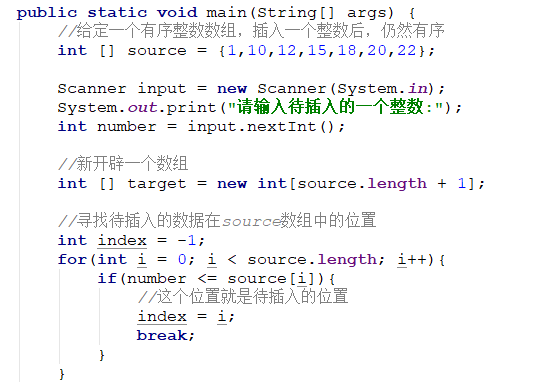


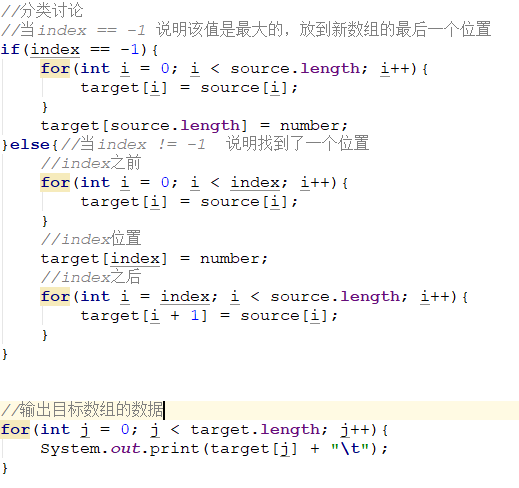
(2)给定一个整数数组,如:{26,12,47,9,11,32},按照如下格式输出(中兴软创笔试题)

9 47 11 32 12 26

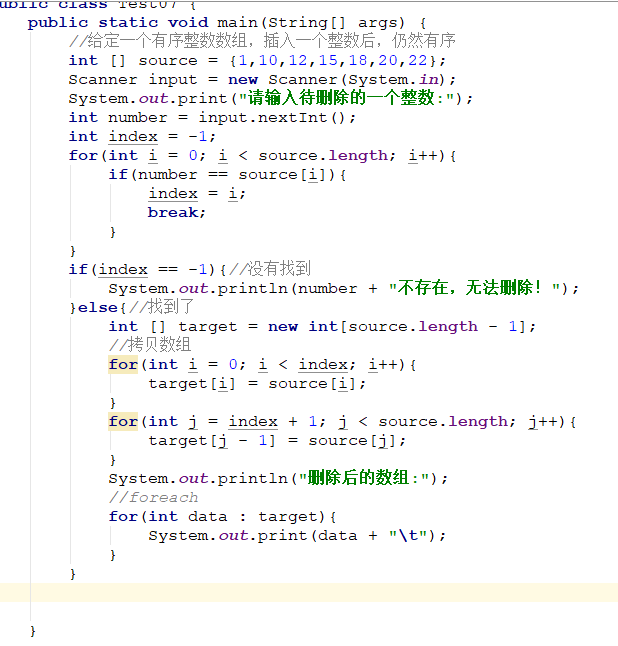


(3)给定一个有序整数数组，插入一个整数后，仍然有序



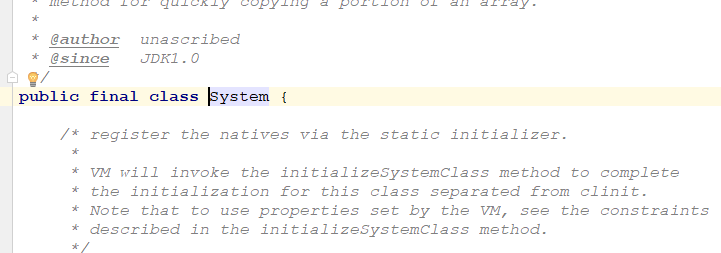


(4)给定一个有序的整数数组，删除一个整数后，仍然有序

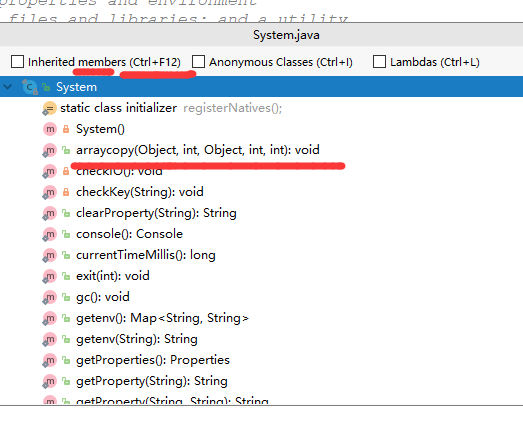


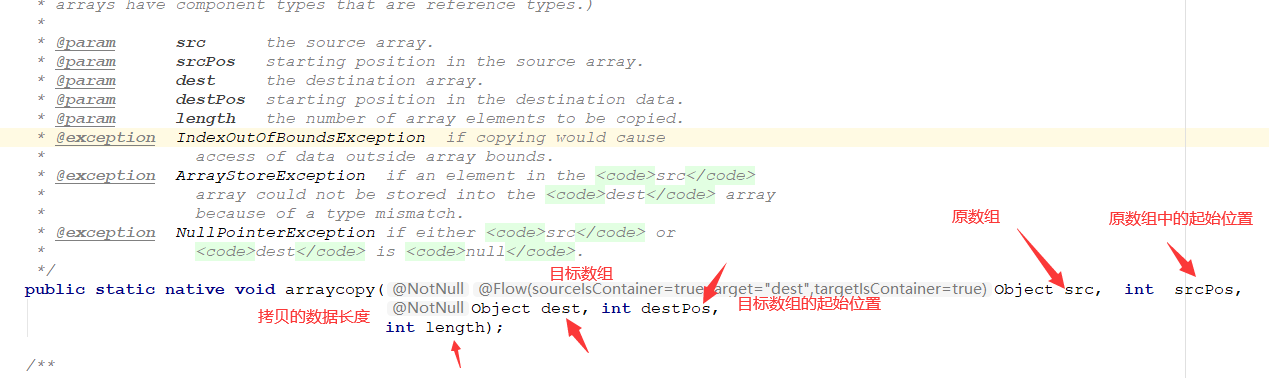
JDK内部已经对数组的拷贝做了封装

System底层的代码：ctrl + 单击进入到System源码中



ctrl + F12可以弹出System中的所有成员





直接使用



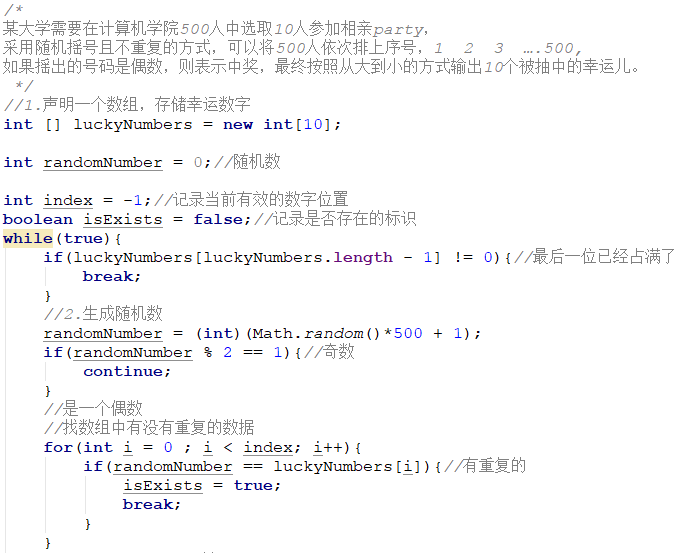


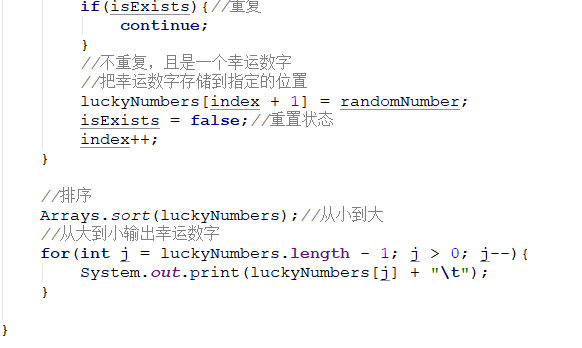
Jdk封装的排序算法



按照自然顺序 ，从小到大。

(5) 某大学需要在计算机学院500人中选取10人参加相亲party，采用随机摇号且不重复的方式，可以将500人依次排上序号，1 2 3 ….500, 如果摇出的号码是偶数，则表示中奖，最终按照从大到小的方式输出10个被抽中的幸运儿。



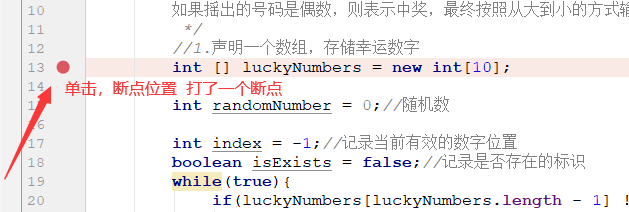


***注意：***

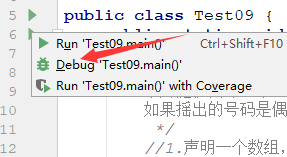
定义问题的解决方案

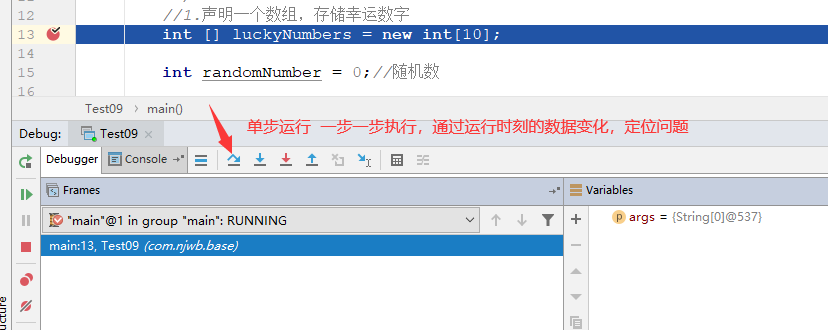
(1)通过异常信息，如：数组越界

(2)通过debug工具 + 单步运行 ，使用程序不报错，但是预期的结果不一致。



运行的时候，采用debug模式运行





# 4.人机猜拳

需求：

