package 八月十号;

import java.util.Scanner;

import javax.jws.soap.SOAPBinding;

public class test {

public static void f(int x){

}

public static int MOD=1000000007;

//main Alt+/ Enter

public static void main(String[] args) {

// //21亿 21亿\*21亿 字符 小数 布尔

// //int long char double boolean

// double PI = 3.1415926;

//// PI+=1;

//// long a=15674863;

// System.out.println(a%MOD);

// //\*\*\*\*\*Scanner sc=new Scanner(System.in);

// Scanner sc=new Scanner(System.in);

//

// //int 21亿

// int a=sc.nextInt();

// int A=sc.nextInt();

// a/A;

// //syso Alt+/ Enter

// for(int i=0;i<10;i++){

// int v=10;

// System.out.println(a);

// if(a==1){

// int u=9;

// }

//

// }

Scanner sc=new Scanner(System.in);

//// double b=sc.nextDouble();

// System.out.println((a<5)?("小"):("大"));

// System.out.println((1+2)\*3);

for(int i=0;i<10;i+=2){

System.out.println(i);

}

int a=sc.nextInt();

while(a>0){

a--;

}

a=-1;

do{

}while(a>0);

if(a<5){

}else if(a>5){

}else{

}

}

}

## 基本语法（大小写敏感）

编写 Java 程序时，应注意以下几点：

* **大小写敏感**：Java 是大小写敏感的，这就意味着标识符 Hello 与 hello 是不同的。
* **类名**：对于所有的类来说，类名的首字母应该大写。如果类名由若干单词组成，那么每个单词的首字母应该大写，例如 **MyFirstJavaClass** 。
* **方法名**：所有的方法名都应该以小写字母开头。如果方法名含有若干单词，则后面的每个单词首字母大写。
* **源文件名**：源文件名必须和类名相同。当保存文件的时候，你应该使用类名作为文件名保存（切记 Java 是大小写敏感的），文件名的后缀为 **.java**。（如果文件名和类名不相同则会导致编译错误）。
* **主方法入口**：所有的 Java 程序由 **public static void main(String[] args)** 方法开始执行。

## Java 标识符（全部用字母）

Java 所有的组成部分都需要名字。类名、变量名以及方法名都被称为标识符。

关于 Java 标识符，有以下几点需要注意：

* 所有的标识符都应该以字母（A-Z 或者 a-z）,美元符（$）、或者下划线（\_）开始
* 首字符之后可以是字母（A-Z 或者 a-z）,美元符（$）、下划线（\_）或数字的任何字符组合
* 关键字不能用作标识符
* 标识符是大小写敏感的
* 合法标识符举例：age、$salary、\_value、\_\_1\_value
* 非法标识符举例：123abc、-salary

## Java修饰符（只记public）

像其他语言一样，Java可以使用修饰符来修饰类中方法和属性。主要有两类修饰符：

* 访问控制修饰符 : default, public , protected, private
* 非访问控制修饰符 : final, abstract, static, synchronized

## Java 变量（★搞清楚变量作用范围）

Java 中主要有如下几种类型的变量

* 局部变量
* 类变量（静态变量）
* 成员变量（非静态变量）

## Java注释

类似于 C/C++、Java 也支持单行以及多行注释。注释中的字符将被 Java 编译器忽略。

## Java 空行

空白行或者有注释的行，Java 编译器都会忽略掉。

## 内置数据类型（基本数据类型）

Java语言提供了八种基本类型。六种数字类型（四个整数型，两个浮点型），一种字符类型，还有一种布尔型。

**byte：**

* byte 数据类型是8位、有符号的，以二进制补码表示的整数；
* 最小值是 **-128（-2^7）**；
* 最大值是 **127（2^7-1）**；
* 默认值是 **0**；
* byte 类型用在大型数组中节约空间，主要代替整数，因为 byte 变量占用的空间只有 int 类型的四分之一；
* 例子：byte a = 100，byte b = -50。

**short：**

* short 数据类型是 16 位、有符号的以二进制补码表示的整数
* 最小值是 **-32768（-2^15）**；
* 最大值是 **32767（2^15 - 1）**；
* Short 数据类型也可以像 byte 那样节省空间。一个short变量是int型变量所占空间的二分之一；
* 默认值是 **0**；
* 例子：short s = 1000，short r = -20000。

**int：**

* int 数据类型是32位、有符号的以二进制补码表示的整数；
* 最小值是 **-2,147,483,648（-2^31）**；
* 最大值是 **2,147,483,647（2^31 - 1）**；
* 一般地整型变量默认为 int 类型；
* 默认值是 **0** ；
* 例子：int a = 100000, int b = -200000。

**long：**

* long 数据类型是 64 位、有符号的以二进制补码表示的整数；
* 最小值是 **-9,223,372,036,854,775,808（-2^63）**；
* 最大值是 **9,223,372,036,854,775,807（2^63 -1）**；
* 这种类型主要使用在需要比较大整数的系统上；
* 默认值是 **0L**；
* 例子： long a = 100000L，Long b = -200000L。  
  "L"理论上不分大小写，但是若写成"l"容易与数字"1"混淆，不容易分辩。所以最好大写。

**float：**

* float 数据类型是单精度、32位、符合IEEE 754标准的浮点数；
* float 在储存大型浮点数组的时候可节省内存空间；
* 默认值是 **0.0f**；
* 浮点数不能用来表示精确的值，如货币；
* 例子：float f1 = 234.5f。

**double：**

* double 数据类型是双精度、64 位、符合IEEE 754标准的浮点数；
* 浮点数的默认类型为double类型；
* double类型同样不能表示精确的值，如货币；
* 默认值是 **0.0d**；
* 例子：double d1 = 123.4。

**boolean：**

* boolean数据类型表示一位的信息；
* 只有两个取值：true 和 false；
* 这种类型只作为一种标志来记录 true/false 情况；
* 默认值是 **false**；
* 例子：boolean one = true。

**char：**

* char类型是一个单一的 16 位 Unicode 字符；
* 最小值是 **\u0000**（即为0）；
* 最大值是 **\uffff**（即为65,535）；
* char 数据类型可以储存任何字符；
* 例子：char letter = 'A';。

## 引用类型（非基本数据类型）

* 在Java中，引用类型的变量非常类似于C/C++的指针。引用类型指向一个对象，指向对象的变量是引用变量。这些变量在声明时被指定为一个特定的类型，比如 Employee、Puppy 等。变量一旦声明后，类型就不能被改变了。
* 对象、数组都是引用数据类型。
* 所有引用类型的默认值都是null。
* 一个引用变量可以用来引用任何与之兼容的类型。
* 例子：Site site = new Site("Runoob")。

## Java 常量（不变量）

常量在程序运行时是不能被修改的。

final double PI = 3.1415927;

### **自动类型转换（加减乘除自动由低到高转换）**

必须满足转换前的数据类型的位数要低于转换后的数据类型，例如: short数据类型的位数为16位，就可以自动转换位数为32的int类型，同样float数据类型的位数为32，可以自动转换为64位的double类型。

### **强制类型转换（由高到低，由低到高任意转换）**

1. 条件是转换的数据类型必须是兼容的。

2. 格式：(type)value type是要强制类型转换后的数据类型

# Java 运算符

计算机的最基本用途之一就是执行数学运算，作为一门计算机语言，Java也提供了一套丰富的运算符来操纵变量。我们可以把运算符分成以下几组：

* 算术运算符（+ - \* \ ++ --）
* 关系运算符( == != > < >= <= )
* 逻辑运算符( && || ! )
* 赋值运算符



* 其他运算符( ? : )

## Java运算符优先级(不管优先级，直接加括号)

循环（for while do...while break continue）

条件（if if-else if-else if-else ）

\*\*数组

\*\*API

\*\*函数