

# Java运算符

所有的程序开发都是一种数据的处理游戏，那么对于数字的处理一定会有所谓的操作模式，而这些操作模式就称作为运算符

例如要进行加法运算肯定要使用“+”这样的运算符来完成，而对运算符而言，也是存在先后关系的，像小学学习四则运算。采用先乘除后加减的顺序来完成。

对于程序开发而言，会有大量的基础运算符提供，这些运算符也都会提供各自的优先顺序，可是从正常的角度而言，不必要记这些运算符的优先级

记住括号优先级最高。

关键性的问题是，对于程序的开发而言，不建议编写很复杂的计算

范例：以一个程序为例

```
public class JavaStudy {  
    public static void main(String args[]) {  
        int x = 20 ;  
        int y = 20 ;  
        int result = x -- + y++ * -- y / x / y * ++ x - --y + y ++ ;  
        System.out.println(result);  
    }  
}
```

如果在项目代码中按照这样的逻辑代码，会被其他开发人员骂死。诅咒，所以对于代码而言，已经告别了这种复杂程序逻辑代码而言，而是更希望去编写一个简单代码易懂

总结：清晰，代码，易懂才是时代的发展方向。

## 2.数学运算符

在Java语言中，数学运算提供了标准支持，包括四则运算。

范例：实现一个简单的四则运算

```
public class JavaStudy {  
    public static void main(String args[]) {  
        int result = 890234890 * 2902390 + (100 * 2);  
        System.out.println(result);  
    }  
}
```

在进行变量计算的时候，编程语言一般会提供简化的运算符（+=、\*=、-=、/=、%=）支持。

范例：简化运算符

```
public class JavaStudy {  
    public static void main(String args[]) {  
        int num = 10 ;  
        num = num + 20 ;  
        System.out.println(num);  
    }  
}
```

含义：相当于取出num数据加上20，又去改变num数据，在内存小的时代，是不建议使用的，所以提供了一系列的简化运算。

```
public class JavaStudy {  
    public static void main(String args[]) {  
        int num = 10 ;  
        num += 20 ;
```

```

        System.out.println(num);
    }
}

```

简化的运算符处理，相对于前者这个比较简化，且计算的答案一致。

范例：求模

```

public class JavaStudy {
    public static void main(String[] args) {
        int num = 10;
        num %= 3;
        System.out.println(num);
    }
}

```

在数学计算里面中，最头疼的就是“++”、“--”，因为这两种运算符有两类使用方式

++变量、变量--：先进行变量的自增加或者自减，然后再进行数字计算。

变量 ++、变量 --：先使用变量进行计算，然后在进行自增或者自减

范例：实现自增和自减

```

public class JavaStudy {
    public static void main(String[] args) {
        int x = 10;
        int y = 20;
        //1、++ x:首先x的内容首先要自增1，为11;
        //2、y--: 先进行计算，使用的内容是20; 计算完成后进行自减
        int result = ++x - y--;
        System.out.println("计算结果: " + result); // -9
        System.out.println("X = " + x); // 11
        System.out.println("y = " + y); // 19
    }
}

```

以上整个代码是当初在内存量不大的时代所提出的处理方式，但是在现在来看。这种计算就变的非常繁琐，尤其是在硬件成本降低的今天。这类操作如果不爽特别低端的情况下，建议不要这种使用。

```

public class JavaStudy {
    public static void main(String[] args) {
        int x = 10;
        int y = 20;
        ++x;
        int result = x - y;
        y--;
        System.out.println("计算结果: " + result); // -9
        System.out.println("X = " + x); // 11
        System.out.println("y = " + y); // 19
    }
}

```

### 3.关系运算符

关系运算符主要特征极速从大到小的比较进行处理，包括大于（>）号，小于号（<）以及大于等于（≥/>=），小于等于（≤/<=），不等（!=）和相等（==）

范例：大小关系判断

```

public class JavaStudy {
    public static void main(String[] args) {
        int x = 10;
    }
}

```

```

        int y = 10 ;
        boolean flag = x > y ; //flags
        System.out.println(flag) ;
    }
}

```

其输出的返回结构的判断结果都将是布尔类型数据。

在进行关系判断的时候特别需要注意的相等的判断问题。在编程语言Java里面，“=”所表示的是赋值运算，而内容相同的比较是“==”

范例：判定相等

```

public class JavaStudy {
    public static void main(String[] args) {
        int x = 10 ;
        int y = 10 ;
        boolean flag = x == y ; //true
        System.out.println(flag) ;
    }
}

```

在进行关系运算的时候除了可以针对所有的数据类型，也可直接使用字符进行处理

范例：直接使用字符进行处理关系 运算

```

public class JavaStudy {
    public static void main(String[] args) {
        char c = '樱' ;
        boolean flag = 27185 == c ; //true
        System.out.println(flag) ;
    }
}

```

数据类型之间提供有转型支持，所以在判断的时候自负会自动转换为Int而后进行关系运算。

#### 4.条件运算符（三目运算符）

三目运算符，又称条件运算符，是计算机语言（C,C++,Java等）的重要组成部分。它是唯一有3个操作数的运算符，所以有时又称为三元运算符。一般来说，三目运算符的结合性是右结合的。

在进行程序开发的时候，条件运算符使用的非常多，合理的利用条件运算符可以避免一些大范围的程序编写。条件运算符一种赋值运算的处理，它是需要一个设置一个大小的逻辑关系判断后才可以进行的赋值操作。

范例：判断两个数字的大小，将最大的值保存

```

public class JavaStudy {
    public static void main(String[] args) {
        int x = 10 ;
        int y = 20 ;
        //判断x与y的大小关系来决定最终max变量的内容
        int max = x > y ? x : y ;
        System.out.println(max) ;
    }
}

```

以上代码为使用条件运算符进行运算，下面讲列举不使用条件运算符进行运算

```

public class JavaStudy {
    public static void main (String args[]) {
        int x = 10 ;
        int y = 20 ;
        //判断x与y的大小关系来决定最终Mac变量内容
    }
}

```

```

        int max = 0 ;
        if (x > y) {
            max = x ;
        } else {
            max = y ;
        }
        System.out.println(max) ;
    }
}

```

对于程序开发而言，代码越少是越少的，条件运算符也可以进行嵌套处理，也就是说在赋值语句上可以进一步编写条件运算符。

范例：条件运算符嵌套

```

public class JavaStudy {
    public static void main (String args[]) {
        int x = 10 ;
        int y = 20 ;
        int z = 15 ;
        int max = x > y ? (x > z ? x : z) : (y > z ? y : z) ;
        System.out.println(max) ;
    }
}

```

虽然可以进行嵌套处理，但是程序可读性差，所欲根据实际情况使用

## 5.位运算符

位运算指的是二进制数据的处理，主要有：与（&）、或（|）、异或（^）、反码（~）、移位处理  
想要理解为操作，则需要清楚十进制于二进制转换处理的逻辑：数字除2取余

范例：观察于（&）计算，如果两位都是1结果才是1

```

public class JavaStudy {
    public static void main(String mase[]) {
        int x = 13 ;
        int y = 7 ;
        System.out.println(x & y) ;
    }
}

```

13的二进制 00001101

7的二进制是 00000111

于（&）运算 00000101 ——> 5

范例：观察或（|）运算，有1位是1，结果就是1，2都是0的时候结果才是0

```

public class JavaStudy {
    public static void main(String mase[]) {
        int x = 13 ;
        int y = 7 ;
        System.out.println(x & y) ;
    }
}

```

13的二进制 00001101

7的二进制是 00000111

或（|）运算 00001111 ——> 15

在进行位运算处理的时候还可以进行移位的计

例如：以2的3次方计算为例

```
public class JavaStudy {  
    public static void main(String[] args) {  
        int x = 2;  
        int y = 2;  
        System.out.println(x << y);  
        System.out.println(x);  
    }  
}
```

2的进制： 00000010

向左移位：00001000 ——> 8

面试题：请解释&&、| 和 || 的区别？

& 和 | 两个运算符可以进行关系运算符于逻辑运算；

在进行逻辑运算的时候，所有的判断条件都要执行；

在进行位运算的时候只针对当前的数据进行与和或的处理；

在 逻辑运算上还可以使用 &&、||

&& :在若干条件判断的时候，如果前面的条件返回false，后续所有的条件都不在判断，最终的结果就是false；

||：在若干的条件判断的时候，如果前面条件返回了true，后续条件不在执行，最终就是true