

Java基础(2)¹

✧ 运算符

运算符就是参与运算的符号，Java提供的运算符有很多种，可以分为下面几种

- 基本算术运算符
- 自增自减运算符
- 赋值运算符
- 关系运算符
- 逻辑运算符
- 三元运算符

算术运算符

算术运算符有 `+` `-` `*` `/` `%`，其中 `*` 表示乘法，`/` 表示除法，`%` 表示取余数



`/`：两个整数相除，结果也是一个整数

`%`：表示两个数相除，取余数

符号	作用	说明
+	加	
-	减	
*	乘	与“x”相同
/	除	与“÷”相同，注意：再java中两个整数相除结果还是整数
%	取余	获取的是两个数据做除法的余数



`+` 符号除了用于加法运算，还可以作为连接符。`+` 符号与字符串运算的时候是用作连接符的，其结果依然是一个字符串。

```

2      public static void main(String[] args){
3          // 目标: 掌握基本的算术运算符的使用
4          int a = 10;
5          int b = 2;
6          System.out.println(a + b); // 12
7          System.out.println(a - b); // 8
8          System.out.println(a * b); // 20
9          System.out.println(a / b); // 5 整数 / 整数 = 整数
10         System.out.println(5 / 2); // 2
11         System.out.println(5.0 / 2); // 2.5
12
13         int i = 5;
14         int j = 2;
15         System.out.println(1.0 * i / j); // 2.5
16         System.out.println(a % b); // 0
17         System.out.println(3 % 2); // 1
18
19         System.out.println("-----");
20
21         // 目标2: 掌握使用+符号做连接符的情况。
22         int a2 = 5;
23         System.out.println("abc" + a2); // abc5
24         System.out.println(a2 + 5); // 10
25         System.out.println("kaifamiao" + a2 + 'a'); // kaifamiao5a
26         System.out.println(a2 + 'a' + "kaifamiao"); //
102kaifamiao
27     }
28 }

```

自增自减运算符

接下来，学习一种比较常用的运算符：**++** 和 **--**

++ 读作自增，**--** 读作自减； 运算规则如下

符号	作用
自增: ++	放在某个变量前面或者后面，对变量自身的值加1
自减: --	放在某个变量前面或者后面，对变量自身的值减一

```

1  1.单独使用: ++或者--放在变量前面没有区别
2      int a = 10;
3      a++; // 1
4      --a; // 10
5      System.out.println(a); // 10
6  2.混合使用: ++或者--放在变量或者前面运算规则稍有不通过 ++ 在后 先赋值 再自增
   ++ 在前 先自增再赋值
7      int a = 10;
8      int b = a++;
9      int c = ++a;
10     System.out.println(a); // 12
11     System.out.println(b); // 10
12     System.out.println(c); // 12

```

赋值运算符

基本的复制运算符 `=`,意思是把右边的数值赋值给左边的变量。

```
1  int a = 10; //将数据10赋值给左边的变量
```

除了基本的赋值运算符，我们在了解一下扩展的运算符，有：`+=`，`-=`，`*=`，`/=`，`%=`。

符号	用法	作用	底层代码形式
<code>+=</code>	<code>a+=b</code>	加后赋值	<code>a=(a的类型)(a+b);</code>
<code>-=</code>	<code>a-=b</code>	减后赋值	<code>a=(a的类型)(a-b);</code>
<code>*=</code>	<code>a*=b</code>	乘后赋值	<code>a=(a的类型)(a * b);</code>
<code>/=</code>	<code>a/=b</code>	除后赋值	<code>a=(a的类型)(a / b);</code>
<code>\%=</code>	<code>a%=b</code>	取余后赋值	<code>a=(a的类型)(a % b);</code>



以 `+=` 为例来看一下它的运算规则，其他的运算符运算同理分析即可。

```

1  int a = 10;
2  a += 5; // a = a + 5;
3  System.out.println(a); // 15

```

例：

```

1 // 问题1: 下面的代码否有问题?
2 byte x = 10;
3 byte y = 30;
4 x = x + y; // byte char short 转换成int类型进行计算
5 System.out.println(x); // 报错
6 // 问题2: 下面的代码是否有问题?
7 byte x = 10;
8 byte y = 30;
9 // x += 3;
10 x += y; // x = (byte) (x + y);
11 System.out.println(x); // 40

```

关系运算符

下表是每一种关系运算符的符号及作用，每一个关系运算符的结果都是boolean值（true、false）

符号	例子	作用	结果
>	a > b	判断a是否大于b	成立返回true，不成立返回false
>=	a >= b	判断a是否大于等于b	成立返回true，不成立返回false
<	a < b	判断a是否小于b	成立返回true，不成立返回false
<=	a <= b	判断a是否小于等于b	成立返回true，不成立返回false
==	a == b	判断a是否等于b	成立返回true，不成立返回false
!=	a != b	判断a是否不等于b	成立返回true，不成立返回false

下面通过代码来家演示一下，各种关系运算符的效果：

```

1 public class Demohhhhjkhghjhjj {
2     public static void main(String[] args) {
3         // 目标: 掌握关系运算符的基本使用。
4         int a = 10;
5         int b = 5;
6         boolean rs = a > b;
7         System.out.println(rs); // true
8         System.out.println(a ≥ b); // true
9         System.out.println(2 ≥ 2); // true
10        System.out.println(a < b); // false
11        System.out.println(a ≤ b); // false
12        System.out.println(2 ≤ 2); // true
13        System.out.println(a == b); // false
14        System.out.println(5 == 5); // true

```

```

15 // 注意了：判断是否相等一定是用 ==，!=是用来赋值的。
16 System.out.println(a == b); // 5 报错
17 System.out.println(a != b); // true
18 System.out.println(10 != 10); // false
19 }
20 }

```



关系运算符在程序中常用于条件判断，根据条件判断的结果是true还是false，来决定后续该执行哪些操作。

逻辑运算符

逻辑运算符是用来将多个条件放在一起运算，最终结果是true或者false。

符号	称呼	例子	运算逻辑
&	逻辑与	$2 > 1 \ \& \ 3 < 2$	多个条件必须是true，结果才是true；有一个是
	逻辑或	$2 > 1 \ \ 3 > 5$	多个条件中主要有一个是true。结果就是true
!	逻辑非	$!(2 > 1)$	就是取反：你真我假，你假我真。
^	逻辑异或	$2 > 1 \ ^ \ 3 > 1$	前后条件的结果相同，就直接返回false，前后条件的结果不同，才返回true
&&	短路与	$2 > 10 \ \&\& \ 3 > 2$	判断结果与"&"一样，过程不同：左边为false，右边就不执行
	短路或	$2 > 1 \ \ 3 < 5$	判断结果与" "一样，过程不同：左边为true，右边就不执行



逻辑运算符在程序中常用于组合几个条件判断，根据条件判断的结果是true还是false，来决定后续该执行哪些操作。

位运算符

在计算及内部，数据以二进制（0和1）的形式存储和处理。计算机通过数据转换为二进制形式，将其存储在内存中，并对其进行操作和计算和操作。不同的数据类型在内存中占据不同的二进制位，以便正确的表示和处理数据。

- ① 源码：源码是一种最简单的机器数表示法，其中最高表示符号位，其余位表示该数的二进制的绝对值。
- ② 反码：正数的反码与与原码相同，而负数的反码是其原码除符号位外，按位取反的结果。
- ③ 补码：正数的补码等于其补码，而负数的补码等于其反码加1。

运算符	含义	运算规则
&	按位与	两个操作数都是1，结果才是1。
	按位或	两个操作数任意一个是1，结果就是1。
^	按位异或	两个操作数相同，结果为0；两个操作数不同，结果为1
~	按位非	操作数为1，结果为0；操作数为0，结果为1。
<<	左移	右侧空位补0。
>>	右移	左侧空位补最高位，即符号位。
>>>	无符号右移	左侧空位补0。

```

1 // 计算5&6的结果
2 // 计算5|6的结果
3 // 计算3<<2的结果
4 // 计算16>>2的结果
5 // 计算-12 | 2的结果

```

三元运算符

三元运算符格式：

```
1 数据类型 变量名 = 关系表达式 ? 值1 : 值2;
```



三元运算的执行流程：首先计算关系表达式的值，如果关系表达式的值为true，则返回 值1；如果关系表达式的值为false, 则返回值2。

运算优先级

优先级	运算符	综合性
1	()、[]	从左向右
2	!、~、++、--	从右向左
3	*/、/、%	从左向右
4	+、-	从左向右
5	<<、>>、>>>	从左向右
6	<、<=、>、>=、instanceof	从左向右
7	==、!=	从左向右
8	&	从左向右
9	^	从左向右
10		从左向右
11	&&	从左向右
12		从左向右
13	?:	从右向左
14	=、+=、-=、*=、/=、%=、&=、 =、^=、&=、<<=、>>=、>>>=	从右向左



在实际开发中，其实我们很少考虑运算优先级，因为如果你想让某些数据先运算，其实加 **()** 就可以了，这样阅读性更高。

1. 第二天学习笔记 ↩