

第三章：运算符

教学内容：

1. 算术运算符
2. 逻辑运算符
3. 关系运算符
4. 位运算符
5. 赋值运算符
6. 条件运算符

一、 算术运算符

加号 (+)、 减号 (-)、 乘号 (*)、 除号 (/)、 取模 (%)、 自增运算符 (++)、 自减运算符 (--)

1. 除号 (/)

java 中将算术运算的结果默认位 int 类型，直接去除小数点；

```
int a = 10;  
int b = 3;  
System.out.println(a / b); // 3
```

2. 取模 (%)

```
int a = 10;  
int b = 3;  
System.out.println(a % b); // 1
```

可以操作浮点数：

3. 自增运算符 (++)、 自减运算符 (--)

4. 前缀++、--与后缀++、--的区别:

++i (前缀++) 在使用 i 之前，使 i 的值加 1，因此执行完++i 后，整个表达式和 i 的值均为 i+1； i++ (后缀++) 在使用 i 之后，使 i 的值加 1，因此执行完 i++ 后，整个表达式的值为 i，而 i 的值变为 i+1

--i 在使用 i 之前，使 i 的减 1，因此执行完 --i 后，整个表达式和 i 的值均为 i-1；i-- 在使用 i 之后，使 i 的减 1，因此执行完 i-- 后，整个表达式的值为 i，而 i 的值变为 i-1

练习：

1)

```
int i = 1;
System.out.println(i++);
```

2)

```
int i = 1;
for (; i < 5; i++) {
    System.out.println(i);
}
System.out.println(i);
```

二、关系运算符

运算符	含义	示例
==	等于	a==b
!=	不等于	a!=b
>	大于	a>b
<	小于	a=	大于或等于	a>=b
<=	小于或等于	a<=b

关系运算符用于操作两个数或者表达式之间的关系，返回一个布尔类型的值。关系运算符都是二元运算符。

三、逻辑运算符

&、|、!、&&、||

Java 中的逻辑运算符

op1	op2	!op1	op1&&op2	op1 op2
false	false	true	false	false
false	true	true	false	true
true	false	false	false	true
true	true	false	true	true

&&、||、! (短路运算方式) 对“或运算”如果运算符左边的表达式的值为 true，则整个表达式的结果为 true，不必对运算符右边的表达式再进行运算；同样，对“与运算”，如果左边表达式的值为 false，则不必对右边的表达式求值，整个表达式的结果为 false。

```
int a = 1;
int b = 2;
int c = 3;
//System.out.println(b++ < a++ && c++ < a++);
//System.out.println(a + "\t" + b + "\t" + c);
//System.out.println(b++ < a++ & c++ < a++);
//System.out.println(a + "\t" + b + "\t" + c);
```

四、 位运算符

按位与运算符 (&)

按位或运算符 (|)

按位异或运算符 (^)

按位取反运算符 (~)

左移位运算符 (<<) 和右移位运算符 (>>)

1. 按位与运算 (&)

参与运算的两个值，如果两个相应位都为 1，则该位的结果为 1，否则为 0。

即：0&0=0，0&1=0，1&0=0，1&1=1

2. 按位或运算 (|)

参与运算的两个值，如果两个相应位都是 0，则该位结果为 0，否则为 1。

即：0|0=0，0|1=1，1|0=1，1|1=1

3. 按位异或运算 (^)

参与运算的两个值，如果两个相应位的某一个是 1，另一个是 0，那么按位异或 (^) 在该位的结果为 1。也就是说如果两个相应位相同，输出位为 0，否则为 1。

即：0^0=0, 0^1=1, 1^0=1, 1^1=0

4. 按位取反运算 (~)

按位取反运算 (~) 属于一元运算符，它只对一个自变量进行操作（其他所有运算符都是二元运算符）。按位取反生成与输入位的相反的值——若输入 0，则输出 1；输入 1，则输出 0。

即：~0=1, ~1=0

5. 左移位运算符 (<<)

运算符 << 执行一个左移位。作左移位运算时，高位溢出，低位补零，右边的空位补 0。在不产生溢出的情况下，数据左移 1 位相当于乘以 2。例如：

int a=64,b; b=a<<1; //b=128

6. 右移位运算符 (>> 与 >>>)

运算符 >> 执行一个右移位（带符号），操作数的构成二进制位右移若干位，地位溢出舍弃，高位补充原来高位的值。（右移一位相当于除 2 取商）

例如：

int a=16,b; b=a>>2; //b=4

五、 赋值运算符： '='

一次可以给多个变量赋值

```
int i = 1000;  
int a, b, c, d;  
a = b = c = d = i;  
System.out.println(a);  
System.out.println(b);  
System.out.println(c);  
System.out.println(d);
```

六、三元运算符

条件表达式 ? 值 1(表达式 1) : 值 2(表达式 2);

当条件表达式为 true 是取值值 1，false 取值值 2；

练习:

使用三元运算符完成：一个 5 分制的评分系统，如果 5 分为 A，4 分为 B，3 分为 C，其它为 D；

PS：如何避免多个运算符带来的问题

- 不要在一行中编写太复杂的表达式，也就是不要在一行中进行太多的运算
- 在一行中进行太多的运算并不能为你带来什么好处，相反只能带来坏处，它并不比改成几条语句的运行速度快，它除可读性差外，还极易出错。

- 对于优先级顺序，读者不用刻意去记，有个印象就行如果你实在弄不清这些运算先后关系的话，就用括号或是分成多条语句来完成你想要的功能，因为括号的优先级是最高的。

- 这也是软件编码规范和企业级开发的一个要求。