

一、运算符介绍

1、简介

运算符是一种特殊的符号，用以表示数据的运算、赋值和比较

2、分类

- 1、算术运算符
- 2、关系运算符
- 3、逻辑运算符
- 4、赋值运算符
- 5、三元运算符
- 6、位运算符

二、算术运算符

1、简介

算术运算符是对数值类型的变量进行运算

2、算术运算符表

运算符	运算	案例	结果
+	正号	+7	7
-	负号	b=11;-b	-11
+	加	9+9	18
-	减	10-8	2
*	乘	7*8	56
/	除	9/9	1
%	取模（取余）	11%9	2
++	自增（前）：先运算后取值	a=2;b=++a	a=3;b=3
++	自增（后）：先取值后运算	a=2;b=a++	a=3;b=2
--	自减（前）：先运算后取值	a=2;b=--a	a=1;b=1
--	自减（后）：先取值后运算	a=2;b=a--	a=1;b=2
+	字符串相加	"he" + "lp"	"help"

三、关系运算符

1、简介

关系运算符的结果都是boolean型，要么为true，要么为false。关系表达式经常用在if结构的条件中或循环结构的条件中

2、关系运算符表

运算符	运算	案例	结果
==	相等	8==7	false
!=	不等	8!=7	true
<	小于	8<7	false
>	大于	8>7	true
<=	小于等于	8<=7	false
>=	大于等于	8>=7	true
instanceof	检查是否是类的对象	"wtf" instanceof String	true

四、逻辑运算符

1、简介

用于连接多个条件（多个关系表达式），最终的结果也是一个boolean值

2、逻辑运算符表

a	b	a&b	a&& b	a b	a b	!a	a^b
true	true	true	true	true	true	false	false
true	false	false	false	true	true	false	true
false	true	false	false	true	true	true	true
false	false	false	false	false	false	true	false

五、赋值运算符

1、简介

赋值运算符就是将某个运算后的值，赋给指定的变量

2、分类

基本赋值运算符

=

复合赋值运算符

+=

-=

*=

/=

%=

六、三元运算符

1、语法

条件表达式？表达式1:表达式2；

2、细节

- 1、如果条件表达式为**true**，运算后的结果是表达式1
- 2、如果条件表达式为**false**，运算后的结果是表达式2

七、运算符优先级

1、简介

运算符有不同的优先级，所谓优先级就是表达式运算中的运算顺序。只有单目运算符、赋值运算符是从右向左运算

2、优先级

- 1、() {}等
- 2、单目运算 ++ --
- 3、算术运算符
- 4、位移运算符
- 5、比较运算符
- 6、逻辑运算符
- 7、三元运算符
- 8、赋值运算符

八、键盘输入语句

1、简介

在编程中，需要接收用户输入的数据，就可以使用键盘输入语句（**Scanner**）来获取。

2、步骤

- 1、导入该类的所在包，`java.util.*`
- 2、创建该类对象（声明对象）
- 3、调用里面的功能

九、进制

1、简介

对于整数，有四种表示方式：

- 1、二进制：0,1，满2进1，以0b或0B开头
- 2、十进制：0-9，满10进1
- 3、八进制：0-7，满8进1
- 4、十六进制：0-9及A(10)-F(15)，满16进1，以0x或0X开头表示

2、二进制转换

二进制转十进制

规则：

从最低位（右边）开始，将每个位上的数提取出来，乘以2的（位数-1）次方，然后求和

案例：

$$0b1011 = 1*2^0 + 1*2^1 + 0*2^2 + 1*2^3 = 1 + 2 + 0 + 8 = 11$$

八进制转十进制

规则：

从最低位（右边）开始，将每个位上的数提取出来，乘以16的（位数-1）次方，然后求和

案例：

$$0234 = 4*8^0 + 3*8^1 + 2*8^2 = 4 + 24 + 128 = 156$$

十六进制转十进制

规则：

从最低位（右边）开始，将每个位的数提取出来，乘以16的（位数-1）次方，然后求和

案例：

$$0x23A = 10*16^0 + 3*16^1 + 2*16^2 = 10 + 48 + 512 = 570$$

十进制转二进制

规则：

将该数不断除以2，直到商为0为止，然后将每步得到的余数倒过来，就是对应的二进制

案例：

32 = 0B00100010

十进制转八进制

规则：

将该数不断除以8，直到商为0为止，然后将每步得到的余数倒过来，就是对应的八进制

案例：

131 = 0203

十进制转十六进制

规则：

将该数不断除以16，直到商为0为止，然后将每步得到的余数倒过来，就是对应的十六进制

案例：

237 = 0xED

二进制转八进制

规则：

从低位开始，将二进制数每三位一组，转成对应的八进制数即可

案例：

0b11010101 = 0325

二进制转十六进制

规则：

从低位开始，将二进制数每四位一组，转成对应的十六进制数即可

案例：

0b11010101 = 0xD5

3、原码、反码、补码

- 1、二进制的最高位是符号位：0表示整数，1表示负数
- 2、正数的原码、反码、补码都一样（三码合一）
- 3、负数的反码 = 它的原码符号位不变，其他位取反（0->1, 1->0）
- 4、负数的补码 = 它的反码+1，负数的反码 = 负数的补码 - 1
- 5、0的反码，补码都是0
- 6、Java没有无符号数，换言之，Java中的数都是有符号的
- 7、在计算机运算的时候，都是以补码的方式运算的
- 8、当我们看运算结果的时候，要看他的原码

4、位运算符

按位与(&)：两位全为1，结果为1，否则为0

按位或(|)：两位有一个为1，结果为1，否则为0

按位异或(^)：两位一个为0，一个为1，结果为1，否则为0

按位取反(~)：0->1, 1->0

算术右移(>>)：低位溢出，符号位不变，并用符号位补溢出的高位

算术左移(<<)：符号位不变，低位补0

逻辑右移(>>>)：低位溢出，高位补0