**算术运算符**

算术运算符用在数学表达式中，它们的作用和在数学中的作用一样。下表列出了所有的算术运算符。

表格中的实例假设整数变量A的值为10，变量B的值为20：

操作符 描述 例子

+ 加法 - 相加运算符两侧的值 A + B 等于 30

- 减法 - 左操作数减去右操作数 A – B 等于 -10

\* 乘法 - 相乘操作符两侧的值 A \* B等于200

/ 除法 - 左操作数除以右操作数 B / A等于2

％ 取余 - 左操作数除以右操作数的余数 B%A等于0

++ 自增: 操作数的值增加1 B++ 或 ++B 等于 21（区别详见下文）

-- 自减: 操作数的值减少1 B-- 或 --B 等于 19（区别详见下文）

实例

下面的简单示例程序演示了算术运算符。复制并粘贴下面的 Java 程序并保存为 Test.java 文件，然后编译并运行这个程序：

实例

public class Test {

public static void main(String[] args) {

int a = 10;

int b = 20;

int c = 25;

int d = 25;

System.out.println("a + b = " + (a + b) );

System.out.println("a - b = " + (a - b) );

System.out.println("a \* b = " + (a \* b) );

System.out.println("b / a = " + (b / a) );

System.out.println("b % a = " + (b % a) );

System.out.println("c % a = " + (c % a) );

System.out.println("a++ = " + (a++) );

System.out.println("a-- = " + (a--) );

// 查看 d++ 与 ++d 的不同

System.out.println("d++ = " + (d++) );

System.out.println("++d = " + (++d) );

}

}

运行实例 »

以上实例编译运行结果如下：

a + b = 30

a - b = -10

a \* b = 200

b / a = 2

b % a = 0

c % a = 5

a++ = 10

a-- = 11

d++ = 25

++d = 27

自增自减运算符

1、自增（++）自减（--）运算符是一种特殊的算术运算符，在算术运算符中需要两个操作数来进行运算，而自增自减运算符是一个操作数。

实例

public class selfAddMinus{

public static void main(String[] args){

int a = 3;//定义一个变量；

int b = ++a;//自增运算

int c = 3;

int d = --c;//自减运算

System.out.println("进行自增运算后的值等于"+b);

System.out.println("进行自减运算后的值等于"+d);

}

}

运行结果为：

进行自增运算后的值等于4

进行自减运算后的值等于2

解析：

int b = ++a; 拆分运算过程为: a=a+1=4; b=a=4, 最后结果为b=4,a=4

int d = --c; 拆分运算过程为: c=c-1=2; d=c=2, 最后结果为d=2,c=2

2、前缀自增自减法(++a,--a): 先进行自增或者自减运算，再进行表达式运算。

3、后缀自增自减法(a++,a--): 先进行表达式运算，再进行自增或者自减运算 实例：

实例

public class selfAddMinus{

public static void main(String[] args){

int a = 5;//定义一个变量；

int b = 5;

int x = 2\*++a;

int y = 2\*b++;

System.out.println("自增运算符前缀运算后a="+a+",x="+x);

System.out.println("自增运算符后缀运算后b="+b+",y="+y);

}

}

运行结果为：

自增运算符前缀运算后a=6，x=12

自增运算符后缀运算后b=6，y=10