

Java Operators - Toán tử trong Java

Một chương trình thực tế bao hàm việc tạo ra các biến. Các toán tử kết hợp các giá trị đơn giản hoặc các biểu thức con thành những biểu thức mới, phức tạp hơn và có thể trả về các giá trị. Điều này có hàm ý tạo ra các toán tử luận lý, số học, quan hệ và so sánh trên các biểu thức.

Java cung cấp nhiều dạng toán tử. Chúng bao gồm:

- Toán tử số học
- Toán tử dạng bit
- Toán tử quan hệ
- Toán tử luận lý
- Toán tử điều kiện
- Toán tử gán

1. Các toán tử số học

Các toán hạng của các toán tử số học phải ở dạng số. Các toán hạng kiểu Boolean không sử dụng được, song các toán hạng ký tự cho phép sử dụng loại toán tử này. Một vài kiểu toán tử được liệt kê trong bảng dưới đây.

Toán tử	Mô tả
+	Cộng.Trả về giá trị tổng hai toán hạng Ví dụ 5+3 trả về kết quả là 8
-	Trừ Trả về giá trị khác nhau giữa hai toán hạng hoặc giá trị phủ định của toán hạng. Ví dụ 5-3 kết quả là 2 và -10 trả về giá trị âm của 10
*	Nhân Trả về giá trị là tích hai toán hạng. Ví dụ 5*3 kết quả là 15
/	Chia Trả về giá trị là thương của phép chia Ví dụ 6/3 kết quả là 2
%	Phép lấy modulo Giá trị trả về là phần dư của toán tử chia Ví dụ 10%3 giá trị trả về là 1

++	Tăng dần Tăng giá trị của biến lên 1. Ví dụ a++ tương đương với a=a+1
--	Giảm dần Giảm giá trị của biến 1 đơn vị. Ví dụ a-- tương đương với a=a-1
+=	Cộng và gán giá trị Cộng các giá trị của toán hạng bên trái vào toán hạng bên phải và gán giá trị trả về vào toán hạng bên trái. Ví dụ c+=a tương đương c=c+a
-=	Trừ và gán giá trị Trừ các giá trị của toán hạng bên trái vào toán toán hạng bên phải và gán giá trị trả về vào toán hạng bên trái. Ví dụ c-= a tương đương với c=c-a
*=	Nhân và gán Nhân các giá trị của toán hạng bên trái với toán toán hạng bên phải và gán giá trị trả về vào toán hạng bên trái. Ví dụ c *= a tương đương với c=c*a
/=	Chia và gán Chia giá trị của toán hạng bên trái cho toán toán hạng bên phải và gán giá trị trả về vào toán hạng bên trái. Ví dụ c /= a tương đương với c=c/a
%=	Lấy số dư và gán Chia giá trị của toán hạng bên trái cho toán toán hạng bên phải và gán giá trị số dư vào toán hạng bên trái. Ví dụ c%=a tương đương với c=c%a

Ví dụ:

```
class ArithmeticOp {
    public static void main(String args[]) {
        int p = 5, q = 12, r = 20, s;
        s = p + q;
        System.out.println("p + q is" + s);
        s = p % q;
        System.out.println("p % q is" + s);
        s *= r;
        System.out.println("s *= r is" + s);
        System.out.println("Value of p before operation is" + p);
        p++;
        System.out.println("Value of p after operation is" + p);
        double x = 25.75, y = 14.25, z;
        z = x - y;
        System.out.println("x - y is" + z);
    }
}
```

```

z -= 2.50;
System.out.println("z -= 2.50 is" + z);
System.out.println("Value of z before operation is" + z);
z--;
System.out.println("Value of z after operation is" + z);
Z = x / y;
System.out.println("x / y is" + z);
}
}

```

Output:

```

p+q is 17
p%q is 5
s*=r is 100
Value of p before operation is 9.0
Value of z after operation is 8.0
x/y is 1.8070175438596429

```

2. Toán tử Bit

Các toán tử dạng Bit cho phép ta tạo những Bit riêng biệt trong các kiểu dữ liệu nguyên thủy. Toán tử Bit dựa trên cơ sở đại số Boolean. Nó thực hiện phép tính trên hai đối số là các bit để tạo ra một kết quả mới. Một vài dạng toán tử kiểu này được liệt kê dưới đây

Toán tử	Mô tả
~	Phủ định (NOT) Trả về giá trị phủ định của một số. Ví dụ a=10 thì ~a=-10
&	Toán tử AND Trả về giá trị là 1 nếu các toán hạng là 1 và 0 trong các trường hợp khác. Ví dụ nếu a=1 và b=0 thì a&b trả về giá trị 0
	Toán tử OR Trả về giá trị là 1 nếu một trong các toán hạng là 1 và 0 trong các trường hợp khác. Ví dụ nếu a=1 và b=0 thì a b trả về giá trị 1
^	Exclusive OR Trả về giá trị là 1 nếu chỉ một trong các toán hạng là 1 và trả về 0 trong các trường hợp khác. Ví dụ nếu a=1 và b=1 thì a^b trả về giá trị 0

>>	Dịch sang phải Chuyển toàn bộ các bit của một số sang phải một vị trí, giữ nguyên dấu của số âm. Toán hạng bên trái là số bị dịch còn số bên phải chỉ số vị trí mà các bit cần dịch. Ví dụ x=37 tức là 00011111 vậy x>>2 sẽ là 00000111.
<<	Dịch sang trái Chuyển toàn bộ các bit của một số sang trái một vị trí, giữ nguyên dấu của số âm. Toán hạng bên trái là số bị dịch còn số bên phải chỉ số vị trí mà các bit cần dịch.

3. Các toán tử quan hệ

Các toán tử quan hệ kiểm tra mối quan hệ giữa hai toán hạng. Kết quả của một biểu thức có dùng các toán tử quan hệ là những giá trị Boolean (logic "đúng" hoặc "sai"). Các toán tử quan hệ được sử dụng trong các cấu trúc điều khiển.

Toán tử	Mô tả
=	So sánh bằng Toán tử này kiểm tra sự tương đương của hai toán hạng Ví dụ if (a= b) trả về giá trị “True” nếu giá trị của a và b như nhau
!=	So sánh khác Kiểm tra sự khác nhau của hai toán hạng Ví dụ if(a!=b) Trả về giá trị “true” nếu a khác b
>	Lớn hơn Kiểm tra giá trị của toán hạng bên phải lớn hơn toán hạng bên trái hay không Ví dụ if(a>b) . Trả về giá trị “true” nếu a lớn hơn b, ngược lại (nhỏ hơn hoặc bằng), trả về ‘False’
<	Nhỏ hơn Kiểm tra giá trị của toán hạng bên phải có nhỏ hơn toán hạng bên trái hay không Ví dụ if(a<b) . Trả về giá trị “true” nếu a nhỏ hơn b, ngược lại (lớn hơn hoặc bằng trả về ‘False’
>=	Lớn hơn hoặc bằng Kiểm tra giá trị của toán hạng bên phải có lớn hơn hoặc bằng toán hạng bên trái hay không Ví dụ if(a>=b) . Trả về giá trị "true" nếu a lớn hơn hoặc bằng b, ngược lại (nhỏ hơn trả về ‘False’
<=	Nhỏ hơn hoặc bằng Kiểm tra giá trị của toán hạng bên phải có nhỏ hơn hoặc bằng toán hạng bên trái hay không Ví dụ if(a<=b) . Trả về giá trị “true” nếu a nhỏ hơn hoặc bằng b , ngược lại (lớn hơn trả về 'False'

Ví dụ:

```

class RelationalOp {
    public static void main(String args[]) {
        float a = 10.0 F;
        double b = 10.0;
        if (a == b)
            System.out.println(a and b are equal");
        else
            System.out.println("a and b are not equal");
    }
}

```

Output:

```
a and b are not equal
```

Trong chương trình trên cả a và b là những số có dấu phẩy động, dạng dữ liệu có khác nhau, a là kiểu float còn b là kiểu double. Tuy vậy chúng không phải là cùng một kiểu. Bởi vậy khi kiểm tra giá trị của các toán hạng, kiểu dữ liệu cần phải được kiểm tra.

4. Các toán tử logic

Các toán tử logic làm việc với các toán hạng Boolean. Một vài toán tử kiểu này được chỉ ra dưới đây:

Toán tử	Mô tả
&	Và (AND) Trả về một giá trị “Đúng” (True) nếu chỉ khi cả hai toán tử có giá trị “True” Ví dụ: if(science>90) AND (math>75) thì gán “Y” cho biến “được nhận học bổng”
	Hoặc (OR) Trả về giá trị “True” nếu một giá trị là True hoặc cả hai đều là True Ví dụ Nếu age_category is ‘Senior_citizen’ OR special_category is ‘handicapped’ thì giảm giá tua lữ hành. Giá cũng sẽ được giảm nếu cả hai điều kiện đều được thỏa mãn
^	XOR Trả về giá trị True nếu chỉ một trong các giá trị là True, các trường hợp còn lại cho giá trị False (sai)
!	Toán hạng đơn tử NOT. Chuyển giá trị từ True sang False và ngược lại. Ví dụ: Quá trình thực thi các dòng lệnh tiếp tục cho đến khi kết thúc chương trình.

5. Các toán tử điều kiện

Toán tử điều kiện là một loại toán tử đặc biệt vì nó gồm ba thành phần cấu thành biểu thức điều kiện:

```
Cú pháp:
biểu thức 1?biểu thức 2: biểu thức 3;
biểu thức 1
Điều kiện luận lý (Boolean) mà nó trả về giá trị True hoặc False
biểu thức 2
Giá trị trả về nếu biểu thức 1 xác định là True
biểu thức 3
Giá trị trả về nếu biểu thức 1 xác định là False
```

Câu lệnh sau đây kiểm tra có những người đi làm bằng vé tháng có tuổi lớn hơn 65 không và gán một tiêu chuẩn cho họ. Nếu những người này có tuổi là 55, tiêu chuẩn gán là "Regular"

```
CommuterCategory=(CommuterAge>65)?"Senior Citizen": "Regular"
```

6. Toán tử gán

Toán tử gán (=) dùng để gán một giá trị vào một biến. Bạn nên gán nhiều giá trị đến nhiều biến cùng một lúc.

Ví dụ đoạn lệnh sau gán một giá trị cho biến num. Thì giá trị trong biến num được gán cho nhiều biến trên một dòng lệnh đơn.

```
int num = 20000;
int p, q, r, s;
p=q=r=s=num;
```

Dòng lệnh cuối cùng được thực hiện từ phải qua trái. Đầu tiên giá trị ở biến num được gán cho 's', sau đó giá trị của 's' được gán cho 'r' và cứ tiếp như vậy.

7. Thứ tự ưu tiên của các toán tử

Các biểu thức được viết ra nói chung gồm nhiều toán tử. Thứ tự ưu tiên quyết định trật tự thực hiện các toán tử trên các biểu thức. Bảng dưới đây liệt kê thứ tự thực hiện các toán tử trong Java

Thứ tự	Toán tử
1.	Các toán tử đơn như +,-,++,--
2.	Các toán tử số học và các toán tử dịch như *,/,+,-,<<,>>
3.	Các toán tử quan hệ như >,<,>=,<=,=,!=

4.	Các toán tử logic và Bit như &&, ,&, ,^
5.	Các toán tử gán như =,*=,/=,+=,-=

Để thay đổi thứ tự ưu tiên trên một biểu thức, bạn có thể sử dụng dấu ngoặc đơn (). Từng phần của biểu thức được giới hạn trong ngoặc đơn được thực hiện trước tiên. Nếu bạn sử dụng nhiều ngoặc đơn lồng nhau thì toán tử nằm trong ngoặc đơn phía trong sẽ thực thi trước, sau đó đến các vòng phía ngoài. Nhưng trong phạm vi một ngoặc đơn thì quy tắc thứ tự ưu tiên vẫn giữ nguyên tác dụng.

Revision #2

Created 28 September 2019 08:22:31 by Laptrinh.vn

Updated 12 April 2020 14:39:04 by Laptrinh.vn