

🕒 Cập nhật tháng 8 năm 2024

[Bài đọc] Toán tử

1. Toán tử số học (Arithmetic Operators):

Toán tử số học được sử dụng để thực hiện các phép tính toán học cơ bản.

Toán tử	Ý nghĩa	Ví dụ	Kết quả
+	Phép cộng	a + b	Tổng của a và b
-	Phép trừ	a - b	Hiệu của a và b
*	Phép nhân	a * b	Tích của a và b
/	Phép chia	a / b	Thương của a và b
%	Phép chia dư	a % b	Số dư của a / b

Ví dụ:

```
public class ArithmeticExample {  
    public static void main(String[] args) {  
        int a = 10;  
        int b = 3;  
        System.out.println("Tổng: " + (a + b)); // Kết quả: 13  
        System.out.println("Hiệu: " + (a - b)); // Kết quả: 7  
        System.out.println("Tích: " + (a * b)); // Kết quả: 30  
        System.out.println("Thương: " + (a / b)); // Kết quả: 3  
        System.out.println("Số dư: " + (a % b)); // Kết quả: 1  
    }  
}
```

Lưu ý: Khi thực hiện phép chia (/) trên các số nguyên, kết quả trả về sẽ là số nguyên. Để nhận kết quả số thực, ít nhất một trong hai toán hạng phải là kiểu float hoặc double.

2. Toán Tử So Sánh (Relational Operators):

Toán tử so sánh được sử dụng để so sánh hai giá trị và trả về kết quả là **true** hoặc **false**.

Toán tử	Ý nghĩa	Ví dụ	Kết quả
==	Bằng nhau	a == b	true nếu a bằng b
!=	Khác nhau	a != b	true nếu a khác b
>	Lớn hơn	a > b	true nếu a lớn hơn b
<	Nhỏ hơn	a < b	true nếu a nhỏ hơn b
>=	Lớn hơn hoặc bằng	a >= b	true nếu a lớn hơn hoặc bằng b
<=	Nhỏ hơn hoặc bằng	a <= b	true nếu a nhỏ hơn hoặc bằng b

Ví dụ:

```
public class RelationalExample { new *
    public static void main(String[] args) { new *
        int a = 5;
        int b = 10;
        System.out.println(a == b); // false
        System.out.println(a != b); // true
        System.out.println(a > b); // false
        System.out.println(a < b); // true
        System.out.println(a >= b); // false
        System.out.println(a <= b); // true
    }
}
```

3. Toán Tử Logic (Logical Operators):

Toán tử logic được sử dụng để kết hợp hoặc đảo ngược các biểu thức điều kiện, thường được sử dụng trong cấu trúc điều kiện và vòng lặp.

Toán tử	Ý nghĩa	Ví dụ	Kết quả
&&	AND (và)	(a > b) && (c > d)	true nếu cả hai điều kiện đều đúng

	OR (hoặc)	(a > b) (c > d)	OR (Hoặc)
!	NOT (phủ định)	!(a > b)	Đảo ngược kết quả điều kiện

Ví dụ:

```
public class LogicalExample { new *
    public static void main(String[] args) { new *
        int a = 5, b = 10, c = 20;
        System.out.println((a > b) && (b < c)); // false (cả hai điều kiện không đúng)
        System.out.println((a > b) || (b < c)); // true (một điều kiện đúng)
        System.out.println(!(a > b)); // true (phủ định điều kiện a > b)
    }
}
```

4. Lưu ý Khi Sử Dụng Toán Tử:

- **Toán tử ưu tiên:** Các toán tử số học được thực hiện trước toán tử so sánh, và toán tử logic được thực hiện sau cùng.
- Ví dụ:

```
boolean result = (5 + 3 > 6) && (10 - 5 == 5); // true
```

- **Phép chia dư:** Hữu ích trong việc kiểm tra tính chẵn lẻ hoặc tuần hoàn.
- Ví dụ:

```
if (x % 2 == 0) {
    System.out.println("x là số chẵn");
}
```

- **Toán tử logic ngắn mạch (Short-circuit):**
- Trong toán tử && và ||, Java dừng đánh giá ngay khi kết quả được xác định.
- Ví dụ:

```
int x = 5;
if (x > 10 && x++ > 3) { // Điều kiện đầu tiên sai, x không tăng
    System.out.println("Không bao giờ chạy");
}
```

Link tài nguyên đọc thêm: <https://www.javatpoint.com/operators-in-java>

