Bài: Các toán tử trong Java

Xem bài học trên website để ủng hộ Kteam: Các toán tử trong Java

Mọi vấn đề về lỗi website làm ảnh hưởng đến bạn hoặc thắc mắc, mong muốn khóa học mới, nhằm hỗ trợ cải thiện Website. Các bạn vui lòng phản hồi đến Fanpage How Kteam nhé!

Dẫn nhập

Ở bài trước, Kteam đã giới thiệu cho các bạn về CÁC KIỂU DỮ LIỆU TRONG JAVA. Trong Java có cung cấp nhiều toán tử để thao tác các biến. Trong bài viết này Kteam sẽ hướng dẫn các bạn về các **Toán tử trong Java**.

Nội dung

Để đọc hiểu bài này, tốt nhất các bạn nên có kiến thức cơ bản về các phần sau:

- CÁC BIẾN TRONG JAVA
- CÁC KIỂU DỮ LIỆU TRONG JAVA

Bài này chúng ta sẽ tìm hiểu những vấn đề sau:

- Danh sách các toán tử, ý nghĩa.
- Độ ưu tiên các toán tử

Danh sách các toán tử, ý nghĩa

Toán tử toán học

Các **toán tử toán học** được sử dụng trong các biểu thức toán học như trong đại số.

- **Ví dụ:** giả sử biến A có giá trị 1, biến B có giá trị 2

Toán tử	Miêu tả	Ví dụ
+	Phép cộng	A + B trả về 3
-	Phép trừ	A – B trả về -1
*	Phép nhân	A * B trả về 2
/	Phép chia lấy nguyên	A / B trả về 0
%	Phép chia lấy dư	A / B trả về 1
++	Tăng giá trị lên 1 đơn vị	A++ trả về 2
--	Giảm giá trị xuống 1 đơn vị	A-- trả về 0

Sự khác biệt a++ (a--) và ++a (--a):

Việc đưa toán tử phía sau có nghĩa là sau khi thực hiện các việc khác thì nó mới tăng (giảm) giá trị, còn đưa lên phía trước thì nó phải tăng (giảm) giá trị rồi mới thực hiện các việc khác.

- **Ví dụ:**

:

```
public class HelloWorld {

    public static void main(String[] args) {
        int a = 1;
        //In rồi mới tăng
        System.out.println(a++);
        //Tăng rồi mới in
        System.out.println(++a);
        //Cộng 1 và in ra rồi mới tăng
        System.out.println(a+++1);
        //Tăng rồi mới cộng 1 và in ra
        System.out.println(++a+1);

    }

}
```

Kết quả:

```
$javac HelloWorld.java
$java  HelloWorld
1
3
4
6
```

Toán tử quan hệ

Các **toán tử quan hệ** được sử dụng để kiểm tra 2 giá trị của 2 biến.

- **Ví dụ:** giả sử biến A có giá trị 1, biến B có giá trị 2

Toán tử	Miêu tả	Ví dụ
==	Kiểm tra xem 2 toán hạng có **bằng nhau hay không**. Đúng trả true, sai trả false	A == B trả về false
!=	Kiểm tra xem 2 toán hạng có **không bằng nhau không**. Đúng trả true, sai trả false	A != B trả về true
>	Kiểm tra xem toán hạng **trái có lớn hơn** toán hạng **phải** không. Đúng trả true, sai trả false	A > B trả về false
<	Kiểm tra xem toán hạng **trái có nhỏ hơn** toán hạng **phải** không. Đúng trả true, sai trả false	A < B trả về true
>=	Kiểm tra xem toán hạng **trái có lớn hơn hoặc bằng** toán hạng **phải** không. Đúng trả true, sai trả false	A >= B trả về false
<=	Kiểm tra xem toán hạng **trái có nhỏ hơn hoặc bằng** toán hạng **phải** không. Đúng trả true, sai trả false	A <= B trả về true

Toán tử thao tác bit

Các **toán tử thao tác bit** chỉ được áp dụng cho các kiểu dữ liệu integer, long, int, short, char và byte. Ta sẽ chuyển các giá trị trong biến sang giá trị integer rồi thao tác trên đó.

- **Ví dụ:** giả sử biến A có giá trị 1 thì giá trị nhị phân là **0000 0001**, biến B có giá trị 2 thì giá trị nhị phân là **0000 0010**

Toán tử	Miêu tả	Ví dụ
&	Sao chép bit 1 đến kết quả nếu nó **tồn tại 2 toán hạng** tại cùng vị trí, ngược lại bit kết quả bằng 0	A & B trả về 0 , hay 0000 0000
|	Sao chép bit 1 đến kết quả nếu nó **tồn tại 1 trong 2 toán hạng** tại cùng vị trí, ngược lại bit kết quả bằng 0	A & B trả về 3, hay 0000 0011
^	Sao chép bit 1 đến kết quả nếu nó **tồn tại chỉ 1 toán hạng tại** cùng vị trí, ngược lại bit kết quả bằng 0	A ^ B trả về 3, hay 0000 0011
~	Dùng để đảo bit 0 thành 1 và ngược lại	~A trả về 254, hay 1111 1110
<<	Dịch trái n bit với n là giá trị bên phải toán hạng	A<<2 trả về 4, hay 0000 0100
>>	Dịch phải n bit với n là giá trị bên phải toán hạng	B>>1 trả về 1, hay 0000 0001

Toán tử logic

Các **toán tử logic** dùng để đưa ra giá trị đúng sai trong một loạt các mệnh đề với nhau

- **Ví dụ:** giả sử biến A có giá trị true, biến B có giá trị false

Toán tử	Miêu tả	Ví dụ
&&	Nếu hai toán tử khác false (hoặc 0) thì trả về true	A && B trả về false
||	Nếu một trong hai toán tử là true (hoặc khác 0) thì trả về true	A || B trả về true
!	Phủ định lại trạng thái toán tử đó	!A trả về false

Toán tử gán

Các **toán tử gán** dùng để thực hiện toán tử số học đồng thời gán cho giá trị mình muốn

Toán tử	Miêu tả	Ví dụ
=	Gán giá trị hạng phải cho hạng trái	A = B
+=	Cộng 2 toán hạng rồi gán giá trị cho toán hạng trái	A += B tương đương A = A + B
-=	Trừ 2 toán hạng rồi gán giá trị cho toán hạng trái	A -= B tương đương A = A – B
*=	Nhân 2 toán hạng rồi gán giá trị cho toán hạng trái	A *= B tương đương A = A * B
/=	Chia lấy nguyên 2 toán hạng rồi gán giá trị cho toán hạng trái	A /= B tương đương A = A / B
%=	Chia lấy dư 2 toán hạng rồi gán giá trị cho toán hạng trái	A %= B tương đương A = A % B
<<=	Dịch trái toán hạng trái sang số vị trí bằng toán hạng phải rồi	A<<=B tương đương A=A<<B
>>=	Dịch phải toán hạng trái sang số vị trí bằng toán hạng phải rồi	A>>=B tương đương A=A>>B
&=	Phép AND Bit	A&=B tương đương A=A&B
^=	Phép OR loại trừ bit	A^=B tương đương A = A^B
|=	Phép OR bit	A!=B tương đương A = A | B

Toán tử khác

Toán tử instanceof

Là toán tử chỉ dùng cho các biến của kiểu dữ liệu tham chiếu. Mục đích để kiểm tra xem biến đó có đúng là kiểu dữ liệu mình dự đoán không, nếu đúng là **true** ngược lại trả về **false**. Ta có cú pháp:

> **<Đối tượng** hoặc **biến> instanceof <class** hoặc **interface>**

- **Ví dụ:**

:

```
public class HelloWorld {

    public static void main(String[] args) {
        String name = “HowKteam”;
        Boolean isString = name instanceof String;
        System.out.println(isString);
    }

}
```

Kết quả trả về là **true**

Toán tử điều kiện

Là toán tử trả về dựa vào điều kiện **đúng hay sai**. Ta có cú pháp như sau:

> **Biến = <điều kiện> ? <giá trị nếu đúng> : <giá trị nếu sai>**

- **Ví dụ:**

:

```
public class HelloWorld {

    public static void main(String[] args) {
        int a = 1;
        int b = 2;
        int c = a<b ? 3 : 4;
        System.out.println(c);
    }

}
```

Kết quả trả về là 3

Độ ưu tiên các toán tử

Đây là bảng độ ưu tiên dựa vào các hạng toán tử được liệt kê phía trên (những trường hợp phức tạp khác đã lược bỏ) với ưu tiên từ trên xuống, từ trái sang phải:

Toán tử	Ghi chú
++ --	Toán tử phía sau
! ~	
++ --	Toán tử phía trước
* / %	
+ -	
<< >>	
< > <= >= instanceof	
== !=	
& ^ |	
&& ||	
?:	Toán tử điều kiện
>>= <<= |= ^= &= %= /= *= -= += =	

Kết luận

Như vậy chúng ta đã tìm hiểu các hạng toán tử trong Java

Ở bài sau, Kteam sẽ giới thiệu đến bạn về **HẰNG TRONG JAVA**

Cảm ơn các bạn đã theo dõi bài viết. Hãy để lại bình luận hoặc góp ý của mình để phát triển bài viết tốt hơn. Đừng quên "**Luyện tập – Thử thách – Không ngại khó**".