

08. Các toán tử (operator) cơ bản trong C#



Nội dung của buổi chia sẻ

1. Toán tử trong C#
2. Các phép toán số học trên các kiểu số
3. Các phép toán so sánh
4. Phép toán điều kiện
5. Các phép gán phức hợp
6. Các phép toán với kiểu dữ liệu: type casting, is và as, typeof
7. Lời kết



1. Toán tử trong C#

NHÓM	TOÁN TỬ
Phép toán số học	+ - * / %
Phép toán logic và nhị phân	& ^ ~ && !
Phép toán ghép xâu	+
Phép toán tăng giảm	++ --
Phép toán dịch bit	<< >>
Phép toán so sánh	== != < > <= >=
Phép gán	= += -= *= /= %= &= = ^= <<= >>=
Phép toán truy xuất thành viên (object và struct)	.
Phép toán indexer (cho mảng)	[]
Ép kiểu (type casting)	()
Phép toán điều kiện	?:
Phép toán cho delegate (thêm/bớt)	+ -
Khởi tạo object	new
Lấy thông tin về kiểu dữ liệu	sizeof is typeof as
Kiểm soát lỗi tràn bộ đệm	checked unchecked
Phép toán liên kết null	??
Phép toán kiểm tra điều kiện null	?. ?[]
Lấy tên của phần tử	nameof()



2. Các phép toán số học trên các kiểu số

Phép toán	Ví dụ
Số dương	<code>+x</code>
Số âm	<code>-x</code>
Tăng sau (post increment)	<code>x++</code>
Giảm sau (post decrement)	<code>x--</code>
Tăng trước (pre-increment)	<code>++x</code>
Giảm trước (pre-decrement)	<code>--x</code>
Nhân	<code>x * y</code>
Chia	<code>x / y</code>
Chia lấy dư	<code>x % y</code>
Cộng	<code>x + y</code>
Trừ	<code>x - y</code>



2. Các phép toán số học trên các kiểu số



3. Các phép toán so sánh

- Còn gọi là các phép toán quan hệ, thực hiện được trên nhiều kiểu dữ liệu nhưng kết quả trả về luôn là kiểu bool.
- Các phép so sánh có thể thực hiện trên các kiểu số, kiểu ký tự và chuỗi.



3. Các phép toán so sánh

Tên	Ví dụ
Nhỏ hơn	$x < y$
Lớn hơn	$x > y$
Nhỏ hơn hoặc bằng	$x \leq y$
Lớn hơn hoặc bằng	$x \geq y$
So sánh bằng	$x == y$
Không bằng	$x \neq y$



4. Phép toán điều kiện

- Là một đặc sản của các ngôn ngữ tương tự C. Nếu bạn biết C, bạn chắc chắn đã biết phép toán này
- Phép toán điều kiện hoạt động gần giống như cấu trúc điều kiện if-else..



5. Các phép gán phức hợp

- Là nhóm phép toán đặc sản của các ngôn ngữ trong họ C.

Trong đó, phép toán này thực hiện một thao tác (như cộng, trừ, nhân, chia, v.v.) và gán ngược giá trị đã biến đổi về cho biến.



5. Các phép gán phức hợp

Phép toán	Ví dụ
<code>+=</code>	<code>x += 1</code> // tương đương <code>x = x + 1</code>
<code>-=</code>	<code>x -= 1</code> // tương đương <code>x = x - 1</code>
<code>*=</code>	<code>x *= 2</code> // tương đương <code>x = x * 2</code>
<code>/=</code>	<code>x /= 2</code> // tương đương <code>x = x / 2</code>



6. Các phép toán với kiểu dữ liệu: type casting, is và as, typeof



6. Các phép toán với kiểu dữ liệu: type casting, is và as, typeof



6.1 Type casting

- Là việc chuyển đổi giá trị của một biến sang một kiểu khác nhưng không làm thay đổi bản chất giá trị của nó.
- Lưu ý, nếu bạn chuyển đổi từ chuỗi “1234” thành số 1234 hay ngược lại, dữ liệu đã bị thay đổi về bản chất. Đây được gọi là **type conversion**.
- Để thực hiện type casting, bạn cần dùng phép toán casting theo cách sau:

```
(<kiểu-đích> <giá-trị>
```



6.2. Type casting với phép toán as

- Phép toán as thực hiện ép kiểu cho giá trị. Nếu không thành công sẽ trả về giá trị null.



6.3. Kiểm tra kiểu – phép toán is

- C# sử dụng phép toán is để kiểm tra kiểu của một giá trị (object).



6.4. Lấy thông tin về kiểu: phép toán typeof

- Phép toán typeof trả về một object chứa thông tin về kiểu dữ liệu.

Từ kết quả này bạn có thể lấy tất cả các thông tin cần thiết về chính kiểu dữ liệu.

- Lưu ý rằng, typeof chỉ nhận tham số là tên kiểu dữ liệu. Nếu muốn lấy thông tin về kiểu từ biến, bạn phải dùng phương thức GetType():



7. Lời kết

- Toán tử trong C#
- Các phép toán số học trên các kiểu số
- Các phép toán so sánh
- Phép toán điều kiện
- Các phép gán phức hợp
- Các phép toán với kiểu dữ liệu: type casting, is và as, typeof



Xin chào và hẹn gặp lại !

