

Bài 2.7: Các toán tử toán học và toán tử gán

- ✓ Tổng quan về các toán tử
- ✓ Các toán tử toán học
- ✓ Các toán tử gán
- ✓ Ví dụ minh họa

Tổng quan về các toán tử

- ✓ Toán tử trong lập trình chính là các phép toán.
- ✓ Các phép toán được chia làm phép toán 1 ngôi, 2 ngôi.
- ✓ Phép toán 1 ngôi là phép toán chỉ cần 1 toán hạng đã có thể hoàn tất nhiệm vụ.
- ✓ Phép toán 2 ngôi là phép toán cần đủ cả 2 toán hạng để hoàn thành nhiệm vụ.
- ✓ Ta sẽ tìm hiểu về các toán tử toán học, toán tử gán, so sánh, toán tử logic.

Các toán tử toán học

- ✓ Chúng dùng để thực hiện các thao tác với các kiểu số để tính toán toán học như cộng trừ nhân chia.
- ✓ Danh sách các toán tử:

Toán tử	Ý nghĩa	Ví dụ
+	Phép cộng hai toán hạng	$c = a + b$
-	Phép trừ hai toán hạng	$c = a - b$
*	Phép nhân hai toán hạng	$c = a * b$
/	Phép chia hai toán hạng	$c = a / b$
%	Phép chia lấy dư hai toán hạng	$c = a \% b$
++	Phép tăng giá trị của một toán hạng lên 1 đơn vị	$c = a++$
--	Phép giảm giá trị của một toán hạng đi 1	$c = b--$

- ✓ Trong số này, toán tử +, -, ++, -- là toán tử 1 ngôi.
- ✓ Trừ ++ và -- ra tất cả đều là toán tử 2 ngôi.

Toán tử +, -, *, /

- ✓ Các toán tử trên thực hiện cộng, trừ, nhân hai số và kết quả lưu vào biến ở kiểu lớn nhất trong 2 toán hạng.
- ✓ Ví dụ: 1 số float cộng 1 số int, kết quả lưu vào kiểu float.

```
0 references
static void Main()
{
    float a = 20.25f;
    int b = 30;
    var sum = a + b; // sum có kiểu float
    Console.WriteLine("Sum = " + sum); // 50.25
    Console.WriteLine(sum.GetType().Name); // Single -> float
}
```

- ✓ Với phép chia /:
 - ✓ Nếu tử và mẫu số cùng là số nguyên, kết quả sẽ là phép chia nguyên.
 - ✓ Nếu tử hoặc mẫu là số thực thì kết quả sẽ ở dạng số thực.
 - ✓ Mẫu số của phép chia phải đảm bảo khác 0. Nếu không kết quả có thể là +- vô cực.

Ví dụ phép chia /

- ✓ Kết quả phép chia:

```
Console.WriteLine(2 / 3); // 0
Console.WriteLine(a / b); // 0.675
Console.WriteLine(2 / 0.0); // vô cực
```

- ✓ Từ đó suy ra, nếu muốn có kết quả chính xác của phép chia, ta phải đưa tử hoặc mẫu số về số thực.
- ✓ Điều trên dễ dàng thực hiện được qua thao tác nhân với 1.0 hoặc ép kiểu về float hoặc double.

```
Console.WriteLine(1.0f * 2 / 3); // cách 1
int x = 2;
int y = 3;
Console.WriteLine((float)x / y); // cách 2, ép kiểu sang float
```

Phép chia lấy dư

- ✓ Áp dụng cho cả số nguyên và số thực.
- ✓ Dấu của kết quả chỉ phụ thuộc dấu của tử số.
- ✓ Kết quả là phần dư còn lại của tử số chia mẫu số.
- ✓ Ví dụ:

```
Console.WriteLine(2 % 3); // 2  
Console.WriteLine(a % b); // 20.25
```

Phép tăng, giảm

- ✓ Ta có thể đặt ++, -- ở trước và sau toán hạng.
- ✓ Toán hạng của các toán tử này phải là các biến.
- ✓ Khi sử dụng với phép gán, ++x và --x có thứ tự ưu tiên cao hơn x++ và x--.

```
// phép ++ và --  
x = 1;  
y = x++; // y = 1, x = 2  
int z = --y; // z = 0, y = 0  
Console.WriteLine($"x = {x}, y = {y}, z = {z}");
```

Các phép gán

- ✓ Các phép gán dùng để gán giá trị cho biến.
- ✓ Toán hạng vế trái phép gán luôn phải là các biến.
- ✓ Toán hạng bên phải có thể tùy ý.
- ✓ Với phép toán $x += 5$ tương đương $x = x + 5$. Tương tự cho các phép gộp khác.

Toán tử	Ý nghĩa	Ví dụ
=	Phép gán	$a = 5$
+=	Phép cộng và gán lại kết quả vào chính nó	$a += 5, a = 10$
-=	Phép trừ và gán lại kết quả vào chính nó	$a -= 5, a = 0$
*=	Phép nhân và gán lại kết quả vào chính nó	$a *= 5, a = 25$
/=	Phép chia và gán lại kết quả vào chính nó	$a /= 5, a = 1$
%=	Phép chia lấy dư và gán lại kết quả vào chính nó	$a \% = 5, a = 0$



Nội dung tiếp theo

Các toán tử(tiếp)