



ASICODE

Курс Основи - C#

ЗАНЯТТЯ № 4 - АРИФМЕТИЧНІ ОПЕРАЦІЇ



Оператор **try-catch**

При возникновении исключения общеязыковая среда выполнения (CLR) ищет оператор `catch`, который обрабатывает это исключение. Если текущий выполняемый метод не содержит такой блок `catch`, среда CLR выполняет поиск в методе, который вызвал текущий метод, и так далее вверх по стеку вызовов. Если блок `catch` не находится, то среда CLR отображает пользователю сообщение о необработанном исключении и останавливает выполнение программы.

Блок `try` содержит защищенный код, который может вызвать исключение. Этот блок выполняется, пока не возникнет исключение или пока он не будет успешно завершен.



Byte



Integer

Integer



Byte

Приведення типів

Необхідність перетворення типів виникає у висловлюваннях, присвоюваннях, заміні формальних аргументів методу фактичними. У C# перетворення поділяються на неявні та явні. До неявних відносяться ті перетворення, результат виконання яких завжди успішний і не призводить до втрати точності даних. До явних відносяться дозволені перетворення, успіх виконання яких не гарантується або може призводити до втрати точності.



переповнення

Якщо переповнення є критичним моментом для певної ділянки коду вашої програми, то можете використовувати перевірку переповнення за допомогою ключового слова `checked`. Суть його роботи полягає в тому, що якщо в рамках контексту, позначеного через `checked` відбувається переповнення, то буде викинуто виключення `OverflowException`.





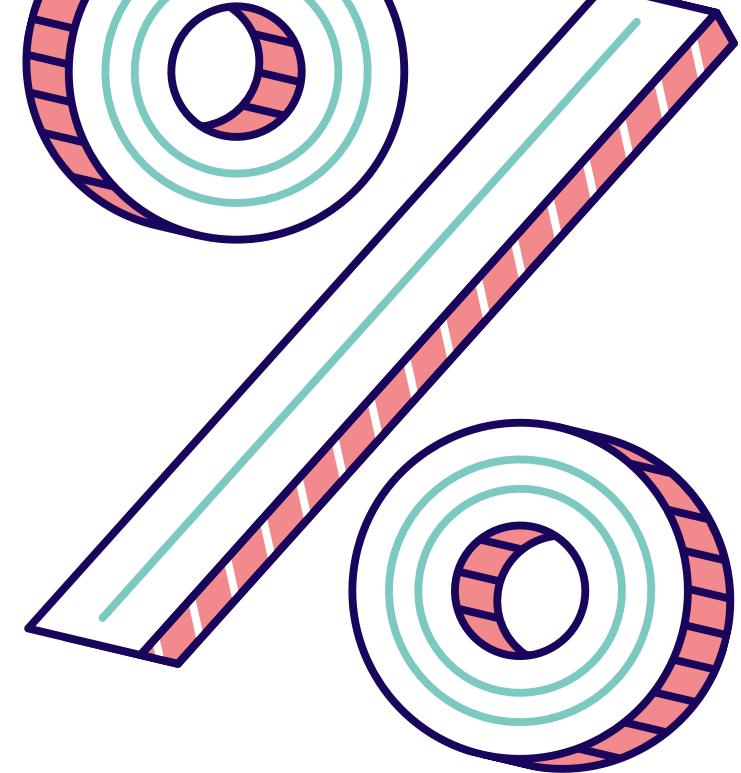
Вбудована конвертація

Для приведення типів можна скористатися класом `System . Convert` , який містить методи приведення одного типу до іншого. Наведемо кілька прикладів його використання. Варіант, коли приведення типу не призводить до переповнення:

```
short s7 = 100 ;
byte b6 = System. Convert . ToByte ( s7 ) ;
Console. WriteLine ( $ "byte value: {b6} , short value: {s7} " );
```

Варіант, коли приведення типу призводить до переповнення, у цьому випадку буде викинуто виняток:

```
short s8 = 500 ;
byte b7 = System. Convert . ToByte ( s8 ) ;
Console. WriteLine ( $ "byte value: {b7} , short value: {s8} " );
```



Базові операції

Для сложения двух чисел в математике мы пишем, например, $3 + 4$. В программировании — то же самое.

Кроме сложения доступны следующие операции:

- * — умножение
- / — деление
- - — вычитание
- % — остаток от деления



Інкремент та декремент

Інкремент бывает префиксным: $++x$ - сначала значение переменной x увеличивается на 1, а потом ее значение возвращается в качестве результата операции.

И также существует постфиксный инкремент: $x++$ - сначала значение переменной x возвращается в качестве результата операции, а затем к нему прибавляется 1.

Операция декремента или уменьшения значения на единицу. Также существует префиксная форма декремента ($--x$) и постфиксная ($x--$).

The diagram shows a sequence of handwritten-style symbols: a small '1' with a downward arrow, followed by a large 'X' with a diagonal cross through it, then another '1' with a downward arrow, and finally an equals sign '=' followed by a large '2'. This illustrates the sequence of operations: first, the value of variable X is increased by 1 (prefix increment), then the result is assigned to a variable (assignment operator), resulting in the value 2.



Складне привласнення

Кроме базовой операции присвоения в C# есть еще ряд операций:

- $+=$: присваивание после сложения. Присваивает левому операнду сумму левого и правого операндов: выражение $A += B$ равнозначно выражению $A = A + B$
- $-=$: присваивание после вычитания. Присваивает левому операнду разность левого и правого операндов: $A -= B$ эквивалентно $A = A - B$
- $*=$: присваивание после умножения. Присваивает левому операнду произведение левого и правого операндов: $A *= B$ эквивалентно $A = A * B$
- $/=$: присваивание после деления. Присваивает левому операнду частное левого и правого операндов: $A /= B$ эквивалентно $A = A / B$
- $\% =$: присваивание после деления по модулю. Присваивает левому операнду остаток от целочисленного деления левого операнда на правый: $A \% = B$ эквивалентно $A = A \% B$



ASICODE

Дякую за увагу