

Типове данни в C#

1. Стойностни типове.

Променлива от определен стойностен тип винаги съдържа стойност от този тип.

C# ви задължава да инициализирате променливите, преди да можете да ги използвате в някакви изчисления - вече не съществуват проблеми с неинициализирани променливи, тъй като компилатора ще ви съобщи когато се опитвате да използвате такива.

Стойностните типове в C# могат да бъдат групирани по следният начин:

- o Прости типове
- o Типове struct
- o Изброени типове

2. Прости типове

Простите типове, които притежава C#, имат някои общи характеристики. Първо, всички те са псевдоними на .NET системните типове. Второ, изразите с константи от прост тип се изчисляват само при компилирането, не и при стартирането. На края, простите типове могат да бъдат инициализирани с литерали.

Простите типове в C# са групирани по следният начин:

Целочислени типове

Тип `bool`

Тип `char` (специален случай на целичислен тип)

Типове с плаваща запетая

Типът `decimal`

3. Целочислени типове

Съществуват девет целочислени типа в C#: sbyte, byte, short, ushort, int, uint, long, ulong и char (обсъден в отделна секция). Те имат следните характеристики:

Типът sbyte представлява 8 битови цели числа със знак, чиито стойности са между -128 и 127.

Типът byte представлява 8 битови цели числа без знак, чиито стойности са между 0 и 255.

Типът short представлява 16 битови цели числа със знак, чиито стойности са между -32,768 и 32,767.

Типът ushort представлява 16 битови цели числа без знак, чиито стойности са между 0 и 65,535.

Типът int представлява 32 битови цели числа със знак, чиито стойности са между -2,147,483,648 и 2,147,483,647.

Типът uint представлява 32 битови цели числа без знак, чиито стойности са между 0 и 4,294,967,295.

Типът long представлява 64 битови цели числа със знак, чиито стойности са между -9,223,372,036,854,775,808 и 9,223,372,036,854,775,807.

Типът ulong представлява 64 битови цели числа без знак, чиито стойности са между 0 и 18,446,744,073,709,551,615

Типът bool представлява булевите променливи true и false. На булева променлива можете да присвоите една от двете стойности (true или false), или път да присвоите израз, чийто резултат се свежда отново до една от тях:

```
bool bTest = (100 > 90);
```

За разлика от C и C++, в C# стойността true вече не се представя от която и да била нулева стойност. Не съществува възможност за конвертиране на целочислени типове в булев такъв, за да се нарушава тази конвенция.

4. Символни литерати. – служат за представяне на някои специални символи, така както е и в C:

Означение	Символно име
'	Единична кавичка
"	Двойна кавичка
\	Обратно наклонена черта
\0	Нула
\a	Бипване
\b	Символ за <code>backspace</code>
\f	Зареждане (<code>form feed</code>)
\n	Нов ред
\r	Връщане в началото на реда
\t	Хоризонтална табулация
\v	Вертикална табулация

5. Типове с плаваща запетая

Два типа за данни спадат към категорията типове с плаваща запетая: `float` и `double`. Разликата между тях е в обхвата от стойности и броя цифри след десетичната запетая:

`float`: Обхватът от стойности е от 1.5×10^{-45} до 3.4×10^{38} степен, с точност от 7 знака след десетичната точка.

`double`: Обхватът от стойности е от 5.0×10^{-324} до 1.7×10^{308} степен с точност от 15-16 знака след десетичната точка.

При извършването на изчисления с които и да било от двата типа с плаваща запетая, могат да се получат следните стойности:

Положителна или отрицателна нула

Положителна или отрицателна безкрайност

Не числена стойност (Not-a-Number - NaN)

Крайно множество от не нулеви стойности

Друго правило при изчисленията на стойностите е, че ако една променлива в даден израз е от тип плаваща запетая, всички останали типове се конвертират в тип с плаваща запетая преди извършването на изчисленията.

6. Типът `decimal`.

Типът `decimal` е много прецизен, 128 битов тип за данни, който е предназначен за употреба при финансови и парични изчисления. Той може да представя стойности, вариращи приблизително от 1.0×10^{-28} до 7.9×10^{28} с 28 до 29 значими цифри. Важно е да се отбележи, че точността се определя от цифри, а не от десетични знаци. Операциите са точни, максимум до 28 десетични знака.

Както можете да видите, обхватът от стойности е по-малък от типа `double`, за сметка на това обаче е доста по-точен. Ето защо не е възможно безусловно конвертиране между типовете `decimal` и `double` - в едната посока можете да предизвикате препълване; в другата да загубите точността. Затова трябва изрично да заявите конвертиране посредством образец.

Когато дефинирате променлива и присвоявате стойност на нея, използвайте суфикса `m`, за да обозначите че стойността е от тип `decimal`:

```
decimal decMyValue = 1.0m;
```

Ако пропуснете суфикса `m`, променливата ще бъде третирана като `double` от компилатора, преди присвояването на стойност.