

# Лабораторна робота 3. Операції з даними

## Мета роботи

Вивчити основні компоненти і набір інструментів інтегрованого середовища розробки Microsoft Visual Studio, які використовуються для розробки застосунків мовою C#.

Вивчити структуру проектів програм, створюваних автоматично на мові C# з використанням бібліотеки базових класів платформи .NET.

Навчитись створювати власні проекти консольних додатків та додатків Windows Forms.

## Зміст

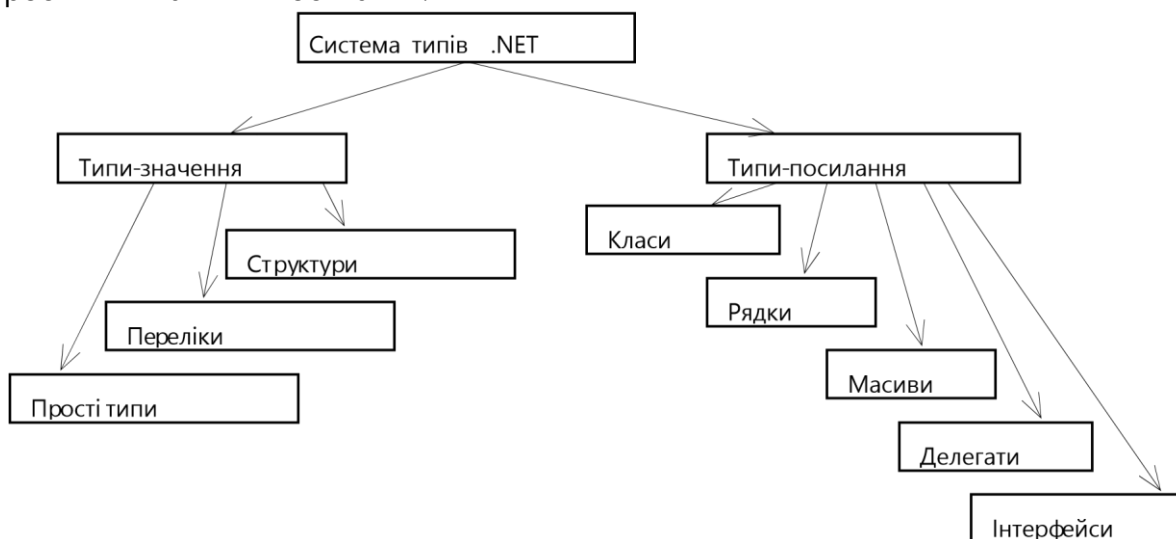
### Основи мови програмування C#

1. Система типів .NET
2. Змінні та константи
3. Основні правила для вибору імен змінним та константам
4. Введення та виведення даних
5. Форматне виведення
6. Операції з даними

## Основи програмування мовою C#

### 1. Система типів .NET

Платформа .NET використовує різні типи даних. Є дві основні категорії вбудованих типів у C# - прості типи та типи-посилання.



C# - мова з суворим контролем типів.

### 2. Змінні та константи

Для збереження даних у програмі використовуються змінні та константи. Змінна - іменована область пам'яті, в яку під час виконання програми можна записувати та зчитувати дані. Константа - іменована область пам'яті, яка містить значення певного типу, яке можна лише зчитувати.

Для того, щоб використовувати змінну її треба оголосити та ініціювати. Для констант оголошення відбувається одночасно з ініціюванням.

Оголошення й ініціалізація змінних:

**Тип\_змінної ім'я\_змінної [=значення];**

```
int x; //оголошення змінної x x=100; //ініціалізація змінної x
long w,z=100; //оголошення змінних w та z й ініціалізація z q=100*z;
//оголошення змінних з динамічною ініціалізацією
```

Оголошення й ініціалізація констант:

Тип\_константи ім'я\_константи [=значення];

```
const double pi=3.14; //оголошення та ініціювання константи pi
```

**Основні прості типи даних в C#:**

Ім'я типу	Граничні значення	Приклад
<b>bool</b>	<b>true</b> або <b>false</b>	true
<b>char</b>	Один символ	'A'
<b>sbyte</b>	Число від -28 до 127	-16
<b>byte</b>	Число від 0 до 255 (8 розрядів)	32
<b>short</b>	Число від -32768 до 32767 (16 розрядів)	-256
<b>ushort</b>	Число від 0 до 65535 (16 розрядів)	128
<b>int</b>	Число від -2147483648 до 2147483648 (32 розряди)	-100000
<b>uint</b>	Число від 0 до 4294967295 (32 розряди)	200000
<b>long</b>	Число від -9223372036854775808 до 9223372036854775807 (64 розряди)	-50000000000L
<b>ulong</b>	Число від 0 до 18446744073709951615 (64 розряди)	100000000000L
<b>float</b>	Число від $1,5 \cdot 10^{-45}$ до $3,4 \cdot 10^{38}$ (32 розряди, точність до 7 десяткових розрядів)	375.95f
<b>double</b>	Число від $5,0 \cdot 10^{-324}$ до $1,7 \cdot 10^{308}$ (64 розряди, точність до 15-16 десяткових розрядів)	1.0000000001d
<b>Dsmal</b>	Число від $1,0 \cdot 10^{-28}$ до $7,9 \cdot 10^{28}$ (128 розряди, точність до 15-16 десяткових розрядів)	75.95m
<b>string</b>	Послідовність символів Unicode	"Welcome"
<b>object</b>	Базовий для всіх інших типів тип	

### 3. Основні правила вибору імен змінним та константам.

1. Першим символом має бути латинська літера, знак підкреслення «\_», або символ «@». Наступними символами імені можуть бути будь-які символи латиниці, знак підкреслення чи цифри. Символи можуть бути як з верхнього так і з нижнього регістру.
2. Для імен не можна використовувати ключові слова мови програмування (Main, int, if, for тощо). Вони виділяються кольором в редакторі коду.  
Приклади допустимих імен: myBigScore, \_qwer, a12.  
Приклади недопустимих імен: \$qw, 54r, 1if, for, x-12.
3. Рекомендується імена простих змінних та констант розпочинати з символів нижнього регістру, імена складних змінних, методів та функцій з верхнього.
4. Якщо ім'я складене з декількох слів, то перша літера другого слова пишеться великою.  
Приклади імен змінних та констант: myWord, myString, myBestSolution.  
Приклади імен методів та функцій: Fib(), Main(), Step(), MaxCount(), Max3().
5. Небажано використовувати символ підкреслення для розділення частин імені (запис word\_count виглядає незграбніше ніж wordCount).
6. Мова C# є чутливою до регістра, якщо ви замість static наберете Static, то виникне помилка, імена myWord та MyWord є різними..
7. Назви просторів імен, типів - розпочинаються з великої літери (Console.WriteLine, System.Windows.Forms.MessageBox).
8. Усі ключові слова мови C# пишуться в нижньому регістрі (наприклад, for, if, static, global).

### 4. Операції з даними

В програмі з прикладу 2 використовується операція присвоєння

```
s = Console.ReadLine();
```

Дані з правої від знака «=» частини записуються до змінної, розміщеної зліва від «=».

Результат виконання функції Console.ReadLine() запишеться до змінної s.

Над дійсними, цілими, натуральними числами можна здійснювати арифметичні операції та обчислювати значення математичних функцій.

**Приклад 1.** Розрахуємо площу трикутника за відомими сторонами.

Введемо значення довжин сторін a, b, c, обчислимо півпериметра трикутника p та за формулою. Герона знайдемо площу трикутника. Результати виведемо в консоль.

```
using System;
namespace triangle
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Введіть сторону a"); // (1)
            int a = Convert.ToInt32(Console.ReadLine()); // (2)
            Console.WriteLine("Введіть сторону b");
            int b = Convert.ToInt32(Console.ReadLine());
            Console.WriteLine("Введіть сторону c");
```

```

int c = Convert.ToInt32(Console.ReadLine());
int p = (a + b + c) / 2;
Console.WriteLine("Півпериметр трикутника p={0}", p);
double s = Math.Sqrt(p * (p - a) * (p - b) * (p - c));
Console.WriteLine("Площа трикутника s={0}", s);
Console.ReadLine();
}
}

```

В рядку (1) виводиться в консоль рядок "Введіть сторону a".

В рядку (2) введений рядок символів конвертується в ціле число.

Після зчитування даних про сторони трикутника в змінні a, b, c у рядку 18 обчислюється та присвоюється змінній p значення півпериметра, а в рядку 20 присвоюється змінній s значення площі трикутника.

**Завдання 1.** Вдосконалити програму. Здійснити перевірку існування трикутника, та здійснювати валідацію даних за допомогою методу TryParse. В усіх випадках виводити необхідні повідомлення.

**Завдання 2\*.** Використовуючи технології Windows Forms намалювати трикутник та вивести його площу під малюнком. Трикутник задати координатами вершин. Дані водити в текстове поле, трикутник малювати за натиском кнопки.

З мови C++ мова C# успадкувала префіксні та постфіксні операції. У наступному прикладі до значення змінної X додамо значення змінної Y, а результат запишемо до змінної X

- $x = ++u$ ; //значення u збільшується на 1, після чого нове значення записується до змінної x;
- $x = u++$ ; // значення u записується до змінної x, а потім значення змінної u збільшується на 1.

## 5. Операції з даними логічного типу

Крім арифметичних виразів у мовах програмування ще один тип виразів – логічний. Логічним назвемо такий вираз, результатом обчислення якого може бути істина (True) або хибна (False).

Для опису такого типу даних використовується слово bool (скорочення від boolean).

В якості операторів у логічних виразах використовуються операції порівняння:

Назва	Позначення
Більше	
Менше	
Менше рівне	> <
Більше рівне	<= >=
рівне	= =
не рівне	!=

та логічні операції

Назва	Позначення
Умовне (логічне множення) I	&&
Умовне (логічне додавання) АБО	
Заперечення (інверсія) НІ	!

Логічна операція && (I) дає істинний результат лише тоді, коли обидва операнди істинні.  
 Логічна операція || (АБО) дає істинний результат тоді, коли хоча б один з операндів істинний.  
 Логічна операція ! (НІ) завжди дає результат протилежний значенню операнда.  
 Приклади простих логічних виразів:

Вираз	Результат		
5==7	False		
4==4	True		
6>=6	True		
6!=8	True		

Замість чисел можуть використовуватися змінні, яким присвоєно числа. Приклади складних логічних виразів

Вираз	Результат
5>2 && 3==5	False
5>2    3==5	True

При складанні як арифметичних, так і логічних виразів слід урахувати пріоритет виконання різних операцій. Для зміни порядку використовуються дужки

Пріоритет	Категорія	Операція	Порядок
0	Первинні	(expr), x y, x->y, f(x), a[x], x++, x--, new, typeof(t), checked(expr), unchecked(expr)	Зліва направо
1	Унарні	+, -, !, ~, ++x, --x, (T)x, sizeof(t)	Зліва направо
2	Мультиплікативні (Множення)	*, /, %	Зліва направо
3	Адитивні (Додавання)	+, -	Зліва направо
4	Зсув	<<, >>	Зліва направо
5	Відношення, перевірка типів	<, >, <=, >=, is, as	Зліва направо
6	Еквівалентність	==, !=	Зліва направо
7	Логічне I (AND)	&	Зліва направо
8	Логічне виключаючи АБО (XOR)	^	Зліва направо
9	Логічне АБО (OR)		Зліва направо

10 Умовне логічне І	&&	Зліва направо
11 Умовне логічне АБО		Зліва направо
12 Умовний вираз	? :	Справа наліво
13 Присвоювання	=, *=, /=, % =, +=, -=, <<=, >>=, &=, ^=,  =	Справа наліво
Склеювання з null	??	
14 Лямбда-оператор	=>	Справа наліво

Розглянемо задачу визначення можливості існування трикутника за довжинами його сторін

### Приклад 3. Використання логічних виразів

```
using System;
namespace c_bool_1
{
    class Program
    {
        public static void Main(string[] args)
        {
            Console.WriteLine("Уведіть довжини трьох сторін трикутника щоразу натискаючи Enter");
            double a=Convert.ToDouble(Console.ReadLine());
            double b=Convert.ToDouble(Console.ReadLine());
            double c=Convert.ToDouble(Console.ReadLine());
            bool sol = a+b>c && a+c>b && b+c>a;
            Console.WriteLine("Результат {0}",sol);
            Console.ReadKey(true);
        }
    }
}
```

**Завдання 3.** Обчисліть значення виразу:

А)  $x > y \ \&\& \ x == 2$ , де  $x=2$ ,  $y=4$ .

Б)  $x == y \ \&\& \ x != 2$ , де  $x=12$ ,  $y=5$ .

В)  $k > v \ || \ k <= 2$ , де  $k=2$ ,  $v=4$ .