



# Вступ до мови програмування C#

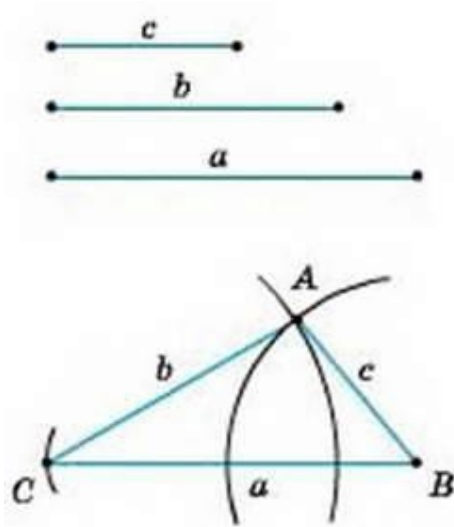


# Вступ до мови програмування C#

Тема 5. Методи (функції).

# Загальне поняття методу

- Метод (від грец. «шлях дослідження, теорія, вчення») - спосіб досягнення будь-якої мети, вирішення конкретного завдання
- Наприклад, в геометрії є метод побудови трикутника за трьома сторонами, а в алгебрі є метод розв'язання системи лінійних рівнянь.



Розв'язування систем лінійних рівнянь  
методом додавання.

$$\begin{cases} 2(4x-5) - 3(3+4y) = 5, \\ 7(6y-1) - (4+3x) = 21y-86; \end{cases} \quad \begin{cases} 8x-10-9-12y=5, \\ 41y-7-4-3x=21y-86; \end{cases}$$

$$\begin{cases} 8x-12y=24, \\ -3x+21y=-75; \end{cases} \quad \begin{cases} 2x-3y=6, \\ -x+7y=-25; \end{cases} \quad \begin{cases} 2x-3y=6 \\ -2x+14y=-50 \end{cases}$$
$$11y=-44, y=-4$$
$$2x+12=6$$

# Методи в мові програмування C#

---

- Метод – це іменований блок коду. Методи забезпечують функціональність програми. За допомогою методів в програмі виконуються різні дії.
- Методи містять більшу частину коду, який забезпечує діяльність програми. При написанні методів можна використовувати усі типи даних (int, double, string) та конструкції для керування потоком виконання програми: умовний оператор, цикли тощо.
- Код, представлений у методі, можна виконати з будь-якого місця програми в області видимості, використовуючи назву методу. Під час роботи програми з методом можна обмінюватися даними, передаючи їх у метод та отримуючи їх від нього.

# Оголошення методу

---

- Оголошення методу містить дві головних секції: заголовок методу та тіло методу. Заголовок методу задає його характеристики: тип результату (якщо він є), назву, дані, які передаються у метод і отримуються від нього. Тіло методу містить послідовність інструкцій, які виконують необхідні дії.
- Виконання завжди починається з першої інструкції методу і послідовно продовжується до виходу з нього.

## Приклад методу без параметрів.

```
class Program
```

```
{
```

```
1 reference
```

```
static void HelloEpam()
```

```
{
```

```
    Console.WriteLine("Hello, EPAM!!!");
```

```
}
```

```
0 references
```

```
static void Main(string[] args)
```

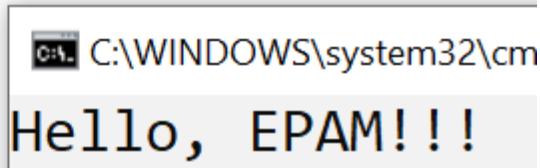
```
{
```

```
    HelloEpam();
```

```
}
```

```
}
```

**Метод не повертає значення**



C:\WINDOWS\system32\cm  
Hello, EPAM!!!

## Приклад методу з одним параметром типу string

```
class Program
```

```
{
```

```
1 reference
```

```
static void HelloName(string name)
```

```
{
```

```
    Console.WriteLine($"Hello, {name}!!!");
```

```
}
```

```
0 references
```

```
static void Main(string[] args)
```

```
{
```

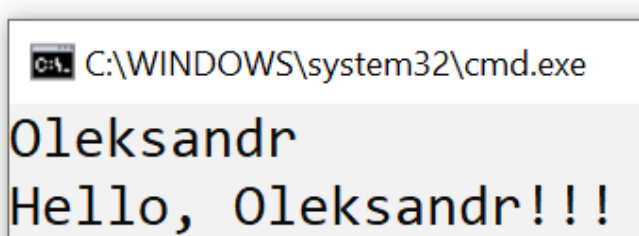
```
    string myname = Console.ReadLine();
```

```
    HelloName(myname);
```

```
}
```

```
}
```

Метод не повертає значення



C:\WINDOWS\system32\cmd.exe

Oleksandr

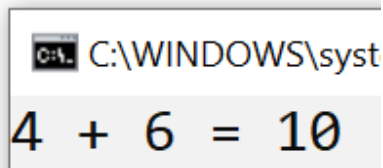
Hello, Oleksandr!!!

## Приклад методу з двома параметрами типу int

```
class Program
```

```
{  
    1 reference  
    static void Sum(int x, int y)  
    {  
        int z = x + y;  
        Console.WriteLine($"{x} + {y} = {z}");  
    }  
  
    0 references  
    static void Main(string[] args)  
    {  
        Sum(4, 6);  
    }  
}
```

Метод не повертає значення





## Приклад методу з двома параметрами

```
class Program
```

```
{
```

```
    1 reference
```

```
    static int Sum(int x, int y)
```

```
    {
```

```
        return x + y;
```

```
    }
```

```
    0 references
```

```
    static void Main(string[] args)
```

```
    {
```

```
        int a = int.Parse(Console.ReadLine());
```

```
        int b = int.Parse(Console.ReadLine());
```

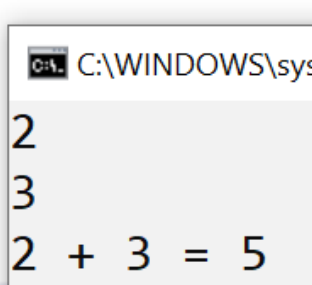
```
        int c = Sum(a, b);
```

```
        Console.WriteLine($"{a} + {b} = {c}");
```

```
    }
```

```
}
```

**Метод повертає значення типу int**



## Приклад методу, параметром якого є одновимірний масив

```
class Program
```

```
{
```

```
    1 reference
```

```
    static int Sum(int [] mas)
```

```
    {
```

```
        int s = 0;
```

```
        foreach(int n in mas)
```

```
        {
```

```
            s = s + n;
```

```
        }
```

```
        return s;
```

```
    }
```

```
    0 references
```

```
    static void Main(string[] args)
```

```
    {
```

```
        int[] massiv = { 3, 5, 1, 8, 2 };
```

```
        int c = Sum(massiv);
```

```
        Console.WriteLine($"Сума = {c}");
```

```
    }
```

```
}
```

**Метод повертає значення типу int**

```
C:\WINDOWS\Sy
```

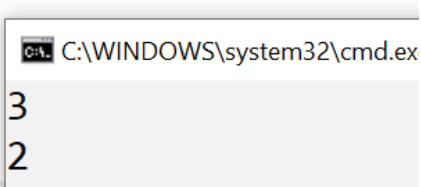
```
Сума = 19
```

## Приклад методу, параметром якого є двовимірний масив

```
class Program
```

```
{  
    static int[] DiagonalElements(int[,] mas)  
    {  
        int rows = mas.GetLength(0);  
        int[] diagonals = new int[rows];  
        for (int i = 0; i < rows; i++)  
            diagonals[i] = mas[i, i];  
        return diagonals;  
    }  
  
    static void Main(string[] args)  
    {  
        int[,] array = { {3, 6},  
                          {4, 2} };  
        foreach(int x in DiagonalElements(array))  
            Console.WriteLine(x);  
    }  
}
```

**Метод повертає масив елементів типу int**



## Домашнє завдання

---

- Напишіть метод, який виводить на екран середнє арифметичне значення двох цілих чисел
- Напишіть метод, який повертає максимальне значення серед елементів одновимірного масиву цілих чисел.
- (\*) Напишіть метод, який виконує паралельний зсув вниз рядків двовимірного масиву цілих чисел. Тобто, нульовий рядок стає першим, перший рядок стає другим і так далі. При цьому останній рядок стає нульовим.

Приклад:

1	3	2	4			2	6	3	9
4	1	5	7	перетвориться на		1	3	2	4
2	6	3	9			4	1	5	7