

## Звіт лабораторної роботи №6

### Тема: Робота з методами у мові C#

Мета: отримати навички роботи з методами. Навчитися розв'язувати логічні задачі та реалізувати їх рішення на мові C#

#### Варіант 24

##### Завдання 1

1. Реалізувати підпрограму, яка визначає максимальне з трьох чисел.
2. Реалізувати підпрограму, яка визначає периметр трикутника.
3. Реалізувати підпрограму визначення площі трикутника по трьом сторонам.
4. Реалізувати підпрограму, яка перевірятиме введене з клавіатури число на парність.
5. Реалізувати підпрограму виведення імені у стовпчик декілька раз.
6. Реалізувати підпрограму визначення площі трикутника по координатам вершин.

```
using System;
```

```
namespace Program  
{
```

```
    public class Program  
    {
```

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

```
            Console.WriteLine("Завдання 1 №1-3");  
            Console.Write("A: ");  
            double a = Convert.ToDouble(Console.ReadLine());  
            Console.Write("B: ");  
            double b = Convert.ToDouble(Console.ReadLine());  
            Console.Write("C: ");  
            double c = Convert.ToDouble(Console.ReadLine());  
            Console.WriteLine(GetMax(a, b, c));  
            Console.WriteLine(Perimeter(a, b, c));  
            Console.WriteLine(Area(a, b, c));  
            Console.WriteLine("Завдання 1 №4");  
            EvenOrOdd();  
            Console.WriteLine("Завдання 1 №5");  
            PrintName();  
            Console.WriteLine("Завдання 1 №6");  
            Console.Write("x1: ");  
            double x1 = Convert.ToDouble(Console.ReadLine());  
            Console.Write("y1: ");  
            double y1 = Convert.ToDouble(Console.ReadLine());  
            Console.Write("x2: ");  
            double x2 = Convert.ToDouble(Console.ReadLine());  
            Console.Write("y2: ");  
            double y2 = Convert.ToDouble(Console.ReadLine());  
            Console.Write("x3: ");  
            double x3 = Convert.ToDouble(Console.ReadLine());  
            Console.Write("y3: ");  
            double y3 = Convert.ToDouble(Console.ReadLine());  
            AreaByCoords(x1, y1, x2, y2, x3, y3);  
        }  
    }  
}
```

```

        Console.ReadKey();
    }
    static string GetMax(double a, double b, double c)
    {
        double[] nums = new double[3];
        nums[0] = a;
        nums[1] = b;
        nums[2] = c;
        return $"Максимальне: {nums.Max()}";
    }
    static string Perimeter(double a, double b, double c)
    {
        return $"Периметр: {a + b + c}";
    }
    static string Area(double a, double b, double c)
    {
        double s = (a + b + c) / 2;
        return $"Площа: {Math.Sqrt(s * (s - a) * (s - b) * (s - c))}";
    }
    static void EvenOrOdd()
    {
        while (true)
        {
            Console.Write("Число (0 щоб вийти): ");
            double num = Convert.ToDouble(Console.ReadLine());
            if (num == 0) break;
            if (num % 2 == 0) Console.WriteLine("Парне");
            else Console.WriteLine("Непарне");
        }
    }
    static void PrintName()
    {
        Console.Write("Ім'я: ");
        string name = Console.ReadLine();
        Console.Write("Кількість повторів: ");
        int i = Convert.ToInt32(Console.ReadLine());
        for (int j = 0; j < i; j++)
        {
            Console.WriteLine(name);
        }
    }
    static void AreaByCoords(double x1, double y1, double x2, double y2, double
x3, double y3)
    {
        Console.WriteLine($"Площа (через векторний добуток): {Math.Abs((x1 - x3) *
(y2 - y3) - (x2 - x3) * (y1 - y3)) / 2}");
    }
}

```

```

Завдання 1 №1-3
А: 75
В: 87
С: 36
Максимальне: 87
Периметр: 198
Площа: 1340,2447537670125
Завдання 1 №4
Число (0 щоб вийти): 1
Непарне
Число (0 щоб вийти): 2
Парне
Число (0 щоб вийти): 3
Непарне
Число (0 щоб вийти): 4
Парне
Число (0 щоб вийти): 0

```

```

Завдання 1 №5
Ім'я: Alex
Кількість повторів: 5
Alex
Alex
Alex
Alex
Alex
Завдання 1 №6
x1: 65
y1: 45
x2: 76
y2: 89
x3: 54
y3: 56
Площа (через векторний добуток): 302,5

```

## Завдання 2

1. Програма повинна зчитувати з клавіатури в масив числа до тих пір, поки не буде введено від'ємне число. Для послідовності чисел, що введена, знайти максимальне та мінімальне числа, а різницю їх додати до кожного числа введеної послідовності. Після чого видати на екран послідовність, що введена, максимальний та мінімальний елементи, а також послідовність чисел, що отримали у результаті додавання.

```

using System;

namespace Program
{
    public class Program
    {
        static void Main(string[] args)
        {
            double[] nums = new double[100];
            int i = 0;
            while (true)
            {
                Console.Write("Число: ");
                double num = Convert.ToDouble(Console.ReadLine());
                if (num < 0) break;
                nums[i] = num;
                i++;
            }
            Func(nums, i);
            Console.ReadKey();
        }
        static void Func(double[] nums, int i)
        {
            Console.WriteLine("Введена послідовність:");
            for (int j = 0; j < i; j++)
            {
                Console.WriteLine(nums[j]);
            }
            double min = nums.Min();
            double max = nums.Max();
            Console.WriteLine($"Мінімальне: {min}");
            Console.WriteLine($"Максимальне: {max}");
            for (int j = 0; j < i; j++)
            {
                nums[j] += max - min;
            }
        }
    }
}

```

```

    }
    Console.WriteLine($"Після додавання {max - min}:");
    for (int j = 0; j < i; j++)
    {
        Console.WriteLine(nums[j]);
    }
}
}
}

```

```

Число: 1
Число: 2
Число: 3
Число: 4
Число: 5
Число: -1
Введена послідовність:
1
2
3
4
5
Мінімальне: 0
Максимальне: 5
Після додавання 5:
6
7
8
9
10

```

2. У матриці  $m \times n$  знайти всі елементи, що є простими числами і замінити їх числом 0. Видати на екран обидві матриці.

```

using System;

namespace Program
{
    public class Program
    {
        static void Main(string[] args)
        {
            Random random = new Random();
            int m = 5, n = 4;
            int[,] nums = new int[m,n];
            Console.WriteLine("Матриця до замін:");
            for (int i = 0; i < nums.GetLength(0); i++)
            {
                for (int j = 0; j < nums.GetLength(1); j++)
                {
                    nums[i, j] = random.Next(0, 100);
                    Console.Write($"{nums[i, j]}\t");
                }
                Console.WriteLine();
            }
            Console.WriteLine("Матриця після замін:");
            for (int i = 0; i < nums.GetLength(0); i++)
            {
                for (int j = 0; j < nums.GetLength(1); j++)
                {
                    Console.Write($"{IsSimple(nums[i, j])}\t");
                }
            }
        }
    }
}

```

```

        Console.WriteLine();
    }
    Console.ReadKey();
}
static int IsSimple(int num)
{
    int j = 0;
    for (int i = 1; i <= Math.Sqrt(num); i++)
    {
        if (num == 1) return num;
        if (num % i == 0) j++;
    }
    if (j > 1) return num;
    else return 0;
}
}
}

```

```

Матриця до замін:
63      52      82      61
2        57      18      88
9        90      31      69
92       53      29      21
57       23      38      21
Матриця після замін:
63      52      82      0
0        57      18      88
9        90      0       69
92       0       0       21
57       0       38      21

```

### Контрольні запитання:

#### 1. Які типи методів у мові C#?

- Інстанційні методи - це методи, що прив'язані до екземпляра класу і можуть бути викликані через посилання на об'єкт.
- Статичні методи - це методи, що не пов'язані з конкретним екземпляром класу і можуть бути викликані через сам клас.
- Віртуальні методи - це методи, що можуть бути перевизначені в похідних класах.
- Абстрактні методи - це методи без тіла, що потрібно перевизначити в похідних класах

#### 2. Які існують ідентифікатори доступу у мові C#?

- public: доступ до змінної або методу відкритий для всіх.
- private: доступ до змінної або методу обмежений лише до того ж класу.
- protected: доступ до змінної або методу обмежений лише до того ж класу і його похідних класів.

- `internal`: доступ до змінної або методу обмежений до того ж проекту.
- `protected internal`: доступ до змінної або методу обмежений до того ж проекту та його похідних класів

### 3. Як викликати методи у мові C#?

Методи викликаються за допомогою імені методу та його параметрів, які передаються у дужках. Якщо метод є членом класу, то спочатку потрібно створити екземпляр класу або викликати статичний метод через ім'я класу.

### 4. Для чого використовується ключове слово `return` в методах?

Ключове слово `return` використовується для повернення значення з методу.

### 5. Що таке сигнатура методу?

Сигнатура методу - це комбінація імені методу та його параметрів. Сигнатура визначає, який саме метод буде викликаний, якщо імена методів співпадають, але різняться їх параметри. У мові C# сигнатура методу складається з імені методу та типів його параметрів.

### 6. Що таке перезавантаження методів?

Перезавантаження методів - це можливість визначити декілька методів з однаковим іменем, але з різною сигнатурою. Коли метод викликається, компілятор вирішує, який саме метод потрібно викликати, залежно від переданих параметрів.