

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

Завдання для самостійного виконання

Завдання 1

1. Реалізувати підпрограму, яка визначає максимальне з трьох чисел.

Код програми:

```
class Program
{
    static double FindMax(double a, double b, double c)
    {
        double max = a;

        if (b > max) max = b;

        if (c > max) max = c;

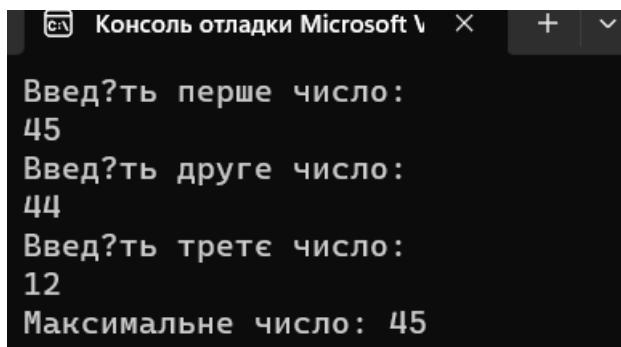
        return max;
    }
    static void Main(string[] args)
    {
        Console.WriteLine("Введіть перше число: ");
        double a = double.Parse(Console.ReadLine());

        Console.WriteLine("Введіть друге число: ");
        double b = double.Parse(Console.ReadLine());

        Console.WriteLine("Введіть третє число: ");
        double c = double.Parse(Console.ReadLine());

        double max = FindMax(a, b, c);
        Console.WriteLine("Максимальне число: " + max);
    }
}
```

Виводимо програму:



```
Консоль отладки Microsoft V  X  +  v
Введ?ть перше число:
45
Введ?ть друге число:
44
Введ?ть третє число:
12
Максимальне число: 45
```

2. Реалізувати підпрограму, яка визначає периметр трикутника.

Код програми:

```
class Program
{
    static void Main()
    {
        double a, b, c;

        Console.Write("Введіть довжину першої сторони: ");
        a = Convert.ToDouble(Console.ReadLine());

        Console.Write("Введіть довжину другої сторони: ");
        b = Convert.ToDouble(Console.ReadLine());

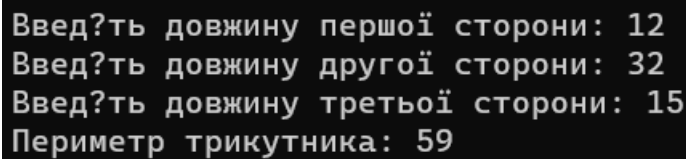
        Console.Write("Введіть довжину третьої сторони: ");
        c = Convert.ToDouble(Console.ReadLine());

        double p = CalculatePerimeter(a, b, c);

        Console.WriteLine("Периметр трикутника: " + p);
    }

    static double CalculatePerimeter(double a, double b, double c)
    {
        return a + b + c;
    }
}
```

Виводимо Програму:



```
Введ?ть довжину першої сторони: 12
Введ?ть довжину другої сторони: 32
Введ?ть довжину третьої сторони: 15
Периметр трикутника: 59
```

3. Реалізувати підпрограму визначення площі трикутника по трьом

Код програми:

```
class Program
{
    static void Main()
    {
        double a, b, c;

        Console.Write("Введіть довжину першої сторони трикутника: ");
        a = double.Parse(Console.ReadLine());

        Console.Write("Введіть довжину другої сторони трикутника: ");
        b = double.Parse(Console.ReadLine());

        Console.Write("Введіть довжину третьої сторони трикутника: ");
        c = double.Parse(Console.ReadLine());

        double s = CalculateTriangleArea(a, b, c);

        if (s > 0)
        {
            Console.WriteLine($"Площа трикутника дорівнює {s}");
        }
        else
        {
            // This block is empty in the original image
        }
    }
}
```

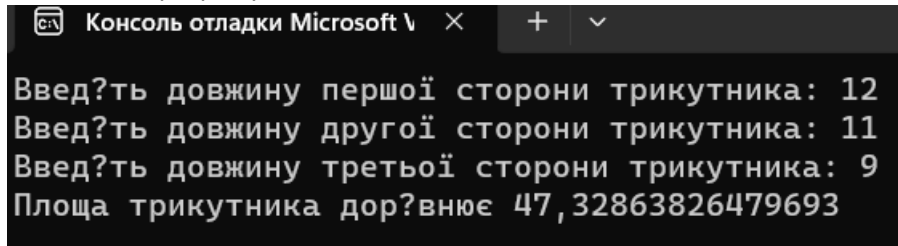
```

        Console.WriteLine("Трикутник з такими сторонами не існує.");
    }
}

static double CalculateTriangleArea(double a, double b, double c)
{
    double s = (a + b + c) / 2;
    double area = Math.Sqrt(s * (s - a) * (s - b) * (s - c));
    return area;
}
}

```

Виводимо програму:



```

Консоль отладки Microsoft \
Введ?ть довжину першої сторони трикутника: 12
Введ?ть довжину другої сторони трикутника: 11
Введ?ть довжину третьої сторони трикутника: 9
Площа трикутника дор?внює 47,32863826479693

```

4. Реалізувати підпрограму, яка перевірятиме введене з клавіатури число на парність.

Код програми:

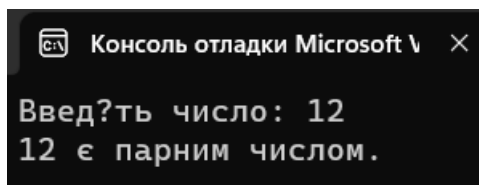
```

class Program
{
    static void Main()
    {
        Console.Write("Введіть число: ");
        int a = int.Parse(Console.ReadLine());

        if (IsEven(a)) Console.WriteLine(a + " є парним числом.");
        else Console.WriteLine(a + " є непарним числом.");
    }
    static bool IsEven(int a)
    {
        return a % 2 == 0;
    }
}

```

Виводимо програму:



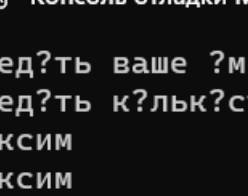
```

Консоль отладки Microsoft \
Введ?ть число: 12
12 є парним числом.

```

Код програми:

Виводимо програму:



```
Консоль отладки Microsoft V
Введ?ть ваше ?м'я: Максим
Введ?ть к?льк?сть раз?v: 10
Максим
Максим
Максим
Максим
Максим
Максим
Максим
Максим
```

6. Реалізувати підпрограму визначення площі трикутника по координатам вершин

Код програми:

```
class Program
{
    static void Main()
    {
        Console.WriteLine("Введіть координати вершин трикутника:");

        double x1, y1, x2, y2, x3, y3;
        Console.Write("x1: ");
        x1 = double.Parse(Console.ReadLine());
        Console.Write("y1: ");
        y1 = double.Parse(Console.ReadLine());

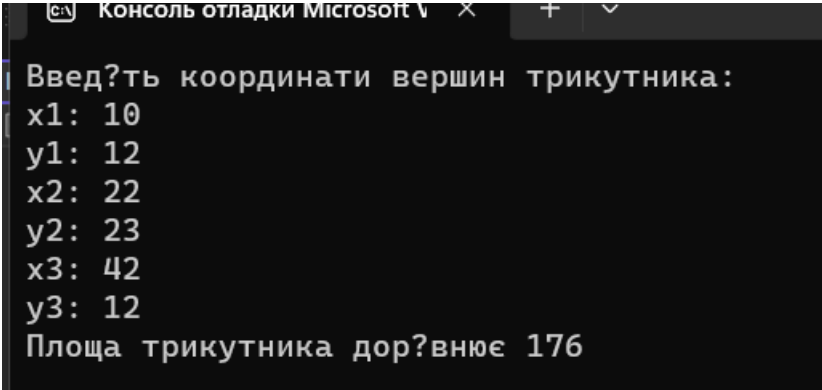
        Console.Write("x2: ");
        x2 = double.Parse(Console.ReadLine());
        Console.Write("y2: ");
        y2 = double.Parse(Console.ReadLine());

        Console.Write("x3: ");
        x3 = double.Parse(Console.ReadLine());
        Console.Write("y3: ");
        y3 = double.Parse(Console.ReadLine());

        double a = CalculateTriangleArea(x1, y1, x2, y2, x3, y3);
        Console.WriteLine($"Площа трикутника дорівнює {a}");
    }

    static double CalculateTriangleArea(double x1, double y1, double x2, double y2,
double x3, double y3)
    {
        double side1 = Math.Sqrt(Math.Pow(x2 - x1, 2) + Math.Pow(y2 - y1, 2));
        double side2 = Math.Sqrt(Math.Pow(x3 - x2, 2) + Math.Pow(y3 - y2, 2));
        double side3 = Math.Sqrt(Math.Pow(x1 - x3, 2) + Math.Pow(y1 - y3, 2));
        double s = (side1 + side2 + side3) / 2;
        double a = Math.Sqrt(s * (s - side1) * (s - side2) * (s - side3));
        return a;
    }
}
```

Виводимо програму:



```
Консоль отладки Microsoft V
Введіть координати вершин трикутника:
x1: 10
y1: 12
x2: 22
y2: 23
x3: 42
y3: 12
Площа трикутника дорівнює 176
```

Завдання 2 Реалізувати наступні завдання у одній програмі використовуючи методи, та перезавантаження методів. В основній програмі повинні бути рядки введення даних для першого і другого завдання, та рядки виведення результатів виконання першого завдання. Всі інші дії повинні бути реалізовані всередині методів. Дозволити циклічне виконання програми по запиту до введення N/n (нескінченний цикл).

ВАРІАНТ № 8 1. Програма повинна зчитувати з клавіатури в масив цілі числа до тих пір, поки сума непарних чисел, що введені, більша за суму парних чисел, після чого повинна видати на екран суму чисел, що попадають в діапазон між першим та другим введеним числом. 2. У матриці $m \times n$ знайти всі від'ємні числа. Видати на екран матрицю і всі від'ємні числа, розмістивши їх по збільшенню.

Код програми:

```
class Program
{
    static void Main(string[] args)
    {
        char choice;
        do
        {
            Console.WriteLine("Виберіть завдання: ");
            Console.WriteLine("1 - Знайти суму чисел в діапазоні");
            Console.WriteLine("2 - Знайти від'ємні числа в матриці");
            Console.WriteLine("N - Завершити програму");

            choice = Console.ReadKey().KeyChar;
            Console.WriteLine();

            switch (choice)
            {
                case '1':
                    Task1();
                    break;
                case '2':
                    Task2();
                    break;
                case 'n':
                case 'N':
                    Console.WriteLine("Програму завершено.");
                    break;
                default:
                    Console.WriteLine("Невірний вибір. Спробуйте ще раз.");
                    break;
            }
        } while (choice != 'n' && choice != 'N');
    }

    static void Task1()
    {
        int[] a = new int[100];

        int b = 0;
        int c = 0;
        int d = 0;
        int e = 0;
        int f = 0;

        while (true)
        {
            Console.Write("Введіть число (або 0 для завершення введення): ");
            int num = int.Parse(Console.ReadLine());
```

```

        if (num == 0)
            break;

        a[f] = num;
        f++;

        if (num % 2 == 0)
            c += num;
        else
            b += num;

        if (b > c && d == 0)
            d = num;

        if (d != 0)
            e = num;
    }

    int g = CalculateSumInRange(a, d, e);
    Console.WriteLine($"Сума чисел у діапазоні від {d} до {e}: {g}");
}

static int CalculateSumInRange(int[] a, int h, int i)
{
    int j = 0;
    bool k = false;

    for (int l = 0; l < a.Length; l++)
    {
        if (a[l] == h)
            k = true;

        if (k)
            j += a[l];

        if (a[l] == i)
            k = false;
    }

    return j;
}

static void Task2()
{
    Console.Write("Введіть кількість рядків матриці: ");
    int m = int.Parse(Console.ReadLine());

    Console.Write("Введіть кількість стовпців матриці: ");
    int n = int.Parse(Console.ReadLine());

    int[,] matrix = new int[m, n];

    Console.WriteLine("Введіть елементи матриці:");

    for (int o = 0; o < m; o++)
    {
        for (int p = 0; p < n; p++)
        {
            Console.Write($"matrix[{o},{p}] = ");
            matrix[o, p] = int.Parse(Console.ReadLine());
        }
    }

    Console.WriteLine("Матриця:");
}

```

```

    for (int q = 0; q < m; q++)
    {
        for (int r = 0; r < n; r++)
        {
            Console.Write($"{matrix[q, r]}\t");
        }
        Console.WriteLine();
    }

    Console.WriteLine("Від'ємні числа в матриці:");

    int[] s = GetNegativeNumbers(matrix);
    Array.Sort(s);

    foreach (int t in s)
    {
        Console.WriteLine(t);
    }
}

static int[] GetNegativeNumbers(int[,] matrix)
{
    int u = matrix.GetLength(0);
    int v = matrix.GetLength(1);

    int w = 0;

    for (int x = 0; x < u; x++)
    {
        for (int y = 0; y < v; y++)
        {
            if (matrix[x, y] < 0)
            {
                w++;
            }
        }
    }

    int[] negatives = new int[w];
    int z = 0;

    for (int x = 0; x < u; x++)
    {
        for (int y = 0; y < v; y++)
        {
            if (matrix[x, y] < 0)
            {
                negatives[z] = matrix[x, y];
                z++;
            }
        }
    }

    return negatives;
}
}

```

Виводимо програму:


```

N - Завершити програму
1
Введ?ть число (або 0 для завершення введення): 1
Введ?ть число (або 0 для завершення введення): 99
Введ?ть число (або 0 для завершення введення): 0
Сума чисел у д?апазон? в?д 1 до 99: 100
Вибер?ть завдання:
1 - Знайти суму чисел в д?апазон?
2 - Знайти в?д'ємн? числа в матриц?
N - Завершити програму
2
Введ?ть к?льк?сть рядк?в матриц?: 3
Введ?ть к?льк?сть стовпц?в матриц?: 3
Введ?ть елементи матриц?:
matrix[0,0] = -2
matrix[0,1] = 24
matrix[0,2] = -5
matrix[1,0] = 5
matrix[1,1] = 24
matrix[1,2] = -56
matrix[2,0] = 54
matrix[2,1] = 76
matrix[2,2] = 88
Матриця:
-2      24      -5
5       24      -56
54      76      88
В?д'ємн? числа в матриц?:
-56
-5
-2

```

Контрольні запитання

1. Які типи методів у мові C#?
2. Які існують ідентифікатори доступу у мові C#?
3. Як викликати методи у мові C#?
4. Для чого використовується ключове слово return в методах?
5. Що таке сигнатура методу?
6. Що таке перезавантаження методів?

Відповіді

1. статичні, нестатичні (екземплярні), віртуальні та абстрактні.
2. public, private, protected, internal та protected internal.
3. Методи у мові C# можна викликати, використовуючи ім'я методу, застосовуючи необхідні аргументи, інкапсульовані в дужках. Наприклад: **methodName(argument1, argument2);**
4. Ключове слово **return** в методах використовується для повернення значення з методу та завершення виконання методу.
5. Сигнатура методу - це унікальний ідентифікатор методу, який включає в себе ім'я методу та типи його параметрів.
6. Перевантаження методів (method overloading) - це можливість мати кілька методів з однаковим ім'ям в одному класі, але з різними списками параметрів.

Висновок:

Освоєння нових команд було складним, але виконавши завдання я отримав навички роботи з методами. Навчився розв'язувати логічні задачі та реалізувати їх рішення на мові C#