

3 Процедуры и функции – методы класса

Задание 1. Разработать метод $f(x)$, который возвращает вторую справа цифру натурального числа x . Вычислить с помощью него значение $z=f(a) + f(b)-f(c)$. Перегрузите метод f из предыдущего раздела так, чтобы его сигнатура(заголовок) соответствовала виду `static void f (double x, out double y)`. Продемонстрируйте работу перегруженных методов.

Листинг программы:

```
namespace MyNamespace
{
    class Program
    {
        static double F(double x, out double y)
        {
            if (x < 0) y = -4;
            else if (x >= 0 && x < 1) y = x * x;
            else y = 2;
            return y;
        }

        static void Main(string[] args)
        {
            Console.WriteLine("Введите x: ");
            double x = Convert.ToDouble(Console.ReadLine());

            double y = F(x, out y);
            Console.WriteLine("y = " + y);
        }
    }
}
```

Таблица 3.1 – Входные и выходные данные

Входные данные	Выходные данные
4	2

					УП 2-40 01 01.37ТП.227.23.03								
Изм.	Лист	№ док	Подпись	Дата									
Разраб.		Сорокина Е.А.			Процедуры и функции – методы класса				Лит	Лист	Листов		
Проверил.		Новик А.И.									12		
									Гродненский ГКТТиД				
Н.контр.													
Утвердил.													

Анализ результатов:

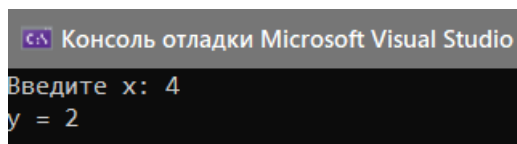


Рисунок 3.1 – Результат работы программы

Задание 2. Описать класс, реализующий тип данных «вещественная матрица» и работу с ними. Класс должен реализовывать следующие операции над матрицами:

методы, реализующие проверку типа матрицы (квадратная, диагональная, нулевая, единичная, симметричная, верхняя треугольная, нижняя треугольная);
операции сравнения на равенство/неравенство;
доступ к элементу по индексам.

Написать программу, демонстрирующую все разработанные элементы класса.

Листинг программы:

```
namespace MySpace
{
    class DoubleMatrix
    {
        private double[,] matrix;
        public int rows, cols;
        private int Length;
        int r = 0;
        bool a = true, t = false;
        public DoubleMatrix(int rows, int cols)
        {
            this.rows = rows;
            this.cols = cols;
            matrix = new double[this.rows, this.cols];
            Length = rows * cols;
        }
        public double this[int index1, int index2]
        {
            get { return matrix[index1, index2]; }
            set { matrix[index1, index2] = value; }
        }
        public void Method()
        {
            for (int i = 0; i < rows; i++)
            {
```

```

        for (int j = 0; j < cols; j++)
        {
            if (matrix[i, j] == 0)
            { r = r + 1; }
        }
    }
    if (r == cols * rows)
    { Console.WriteLine("Матрица нулевая"); }
    else { Console.WriteLine("Матрица не нулевая"); }
}
public bool Method1()
{
    if (cols == rows)
    {
        for (int i = 0; i < matrix.GetLength(0); ++i)
        {
            for (int j = 0; j < matrix.GetLength(1); ++j)
            if (matrix[i, j] != matrix[j, i])
            {
                a = false;
                break;
            }
            if (!a) break;
        }
        return a;
    }
    else return false;
}
public void Method2()
{
    if (rows == cols)
    {
        Console.WriteLine("Матрица квадратная");
    }
    else Console.WriteLine("Матрица не квадратная");
}
public bool Method3()
{
    if (rows == cols)
    {
        for (int i = 0; i < rows; i++)
        {
            for (int j = 0; j < cols; j++)
            {

```

```

        if (i == j && matrix[i, j] != 1)
        {
            return false;
        }
        else if (i != j && matrix[i, j] != 0)
        { return false; }
    }
}
return true;
}
return false;
}
public bool Method4()
{
    if (rows == cols)
    {
        for (int i = 0; i < rows; i++)
        {
            for (int j = 0; j < cols; j++)
            {
                if (i != j)
                {
                    if (matrix[i, j] == 0)
                    {
                        t = true;
                    }
                    else t = false;
                    break;
                }
            }
        }
    }
    return t;
}
public bool Method5()
{
    if (rows == cols)
    {
        for (int i = 0; i < rows; i++)
        {
            for (int j = 0; j < cols; j++)
            {
                if (matrix[i, j] != 0 && i > j)
                {

```

```

        return false;
    }
}
return true;
}
return false;
}
public bool Method6()
{
    if (rows == cols)
    {
        for (int i = 0; i < rows; i++)
        {
            for (int j = 0; j < cols; j++)
            {
                if (matrix[i, j] != 0 && i < j)
                {
                    return false;
                }
            }
        }
        return true;
    }
    return false;
}
public static bool Equals(DoubleMatrix arr1, DoubleMatrix arr2)
{
    for (int i = 0; i < 10; i++)
        for (int j = 0; j < 10; j++)
            if (arr1[i, j] != arr2[i, j])
                return false;
    return true;
}
}
class Program
{
    static void Main(string[] args)
    {
        Console.Write("Введите количество строк матрицы: ");
        int n = Convert.ToInt32(Console.ReadLine());
        Console.Write("Введите количество столбцов матрицы: ");
        int m = Convert.ToInt32(Console.ReadLine());
        DoubleMatrix matrix = new DoubleMatrix(n, m);
    }
}

```

```

Random rand = new Random();
Console.WriteLine();
Console.WriteLine("Матрица :");
for (int i = 0; i < matrix.rows; i++)
{
    for (int j = 0; j < matrix.cols; j++)
    {
        matrix[i, j] = matrix[i, j] = rand.Next(0, 10);
        Console.Write(matrix[i, j] + " ");
    }
    Console.WriteLine();
}
Console.WriteLine();
matrix.Method();
Console.WriteLine();
if (matrix.Method1() == false)
{
    Console.WriteLine("Матрица не симметричная");
}
else Console.WriteLine("Матрица симметричная");
Console.WriteLine();
matrix.Method2();
Console.WriteLine();
if (matrix.Method3())
{
    Console.WriteLine("Матрица единичная");
}
else Console.WriteLine("Матрица не единичная");
Console.WriteLine();
if (matrix.Method4() == true)
{
    Console.WriteLine("Матрица диагональная");
}
else Console.WriteLine("Матрица не диагональная");
Console.WriteLine();
if (matrix.Method5())
{
    Console.WriteLine("Матрица верхняя треугольная ");
}
else Console.WriteLine("Матрица не верхняя треугольная ");
Console.WriteLine();
if (matrix.Method6())
{
    Console.WriteLine("Матрица нижняя треугольная ");
}

```

```

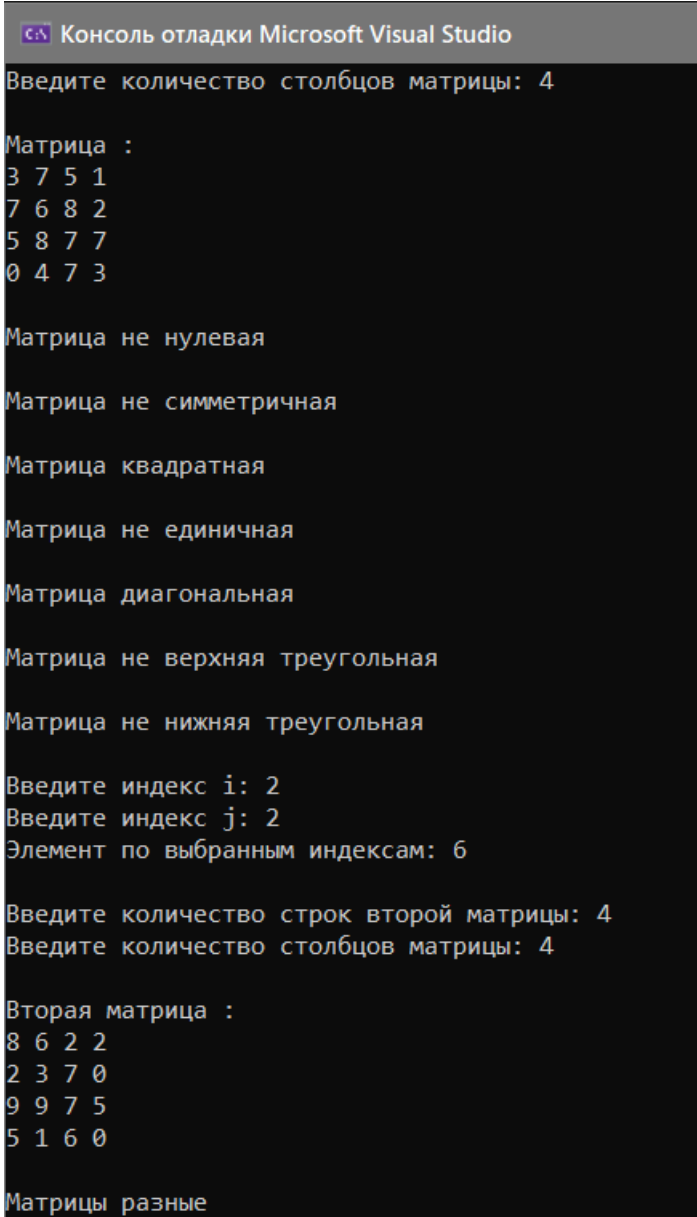
    }
    else Console.WriteLine("Матрица не нижняя треугольная ");
    Console.WriteLine();
    Console.Write("Введите индекс i: ");
    int ii = Convert.ToInt32(Console.ReadLine()) - 1;
    Console.Write("Введите индекс j: ");
    int jj = Convert.ToInt32(Console.ReadLine()) - 1;
    Console.Write("Элемент по выбранным индексам: ");
    Console.WriteLine(matrix[ii, jj]);
    Console.WriteLine();
    Console.Write("Введите количество строк второй матрицы: ");
    int n2 = Convert.ToInt32(Console.ReadLine());
    Console.Write("Введите количество столбцов матрицы: ");
    int m2 = Convert.ToInt32(Console.ReadLine());
    DoubleMatrix matrix2 = new DoubleMatrix(n2, m2);
    Console.WriteLine();
    Console.WriteLine("Вторая матрица :");
    for (int i = 0; i < matrix2.rows; i++)
    {
        for (int j = 0; j < matrix2.cols; j++)
        {
            matrix2[i, j] = matrix2[i, j] = rand.Next(0, 10);
            Console.Write(matrix2[i, j] + " ");
        }
        Console.WriteLine();
    }
    Console.WriteLine();
    if (DoubleMatrix.Equals(matrix, matrix2) == false)
    {
        Console.WriteLine("Матрицы разные");
    }
    else Console.WriteLine("Матрицы одинаковые");
    Console.ReadKey();
}
}
}

```

Таблица 3.1 – Входные и выходные данные

Входные данные	Выходные данные
4, 4	Матрица
2, 2	6
4, 4	Матрица

Анализ результатов:



```
Консоль отладки Microsoft Visual Studio
Введите количество столбцов матрицы: 4

Матрица :
3 7 5 1
7 6 8 2
5 8 7 7
0 4 7 3

Матрица не нулевая
Матрица не симметричная
Матрица квадратная
Матрица не единичная
Матрица диагональная
Матрица не верхняя треугольная
Матрица не нижняя треугольная

Введите индекс i: 2
Введите индекс j: 2
Элемент по выбранным индексам: 6

Введите количество строк второй матрицы: 4
Введите количество столбцов матрицы: 4

Вторая матрица :
8 6 2 2
2 3 7 0
9 9 7 5
5 1 6 0

Матрицы разные
```

Рисунок 3.2 – Результат работы программы