

Часть 1. Массивы

Цель работы: изучить массивы в C#.

Выполнение: объектная декомпозиция изображена на рисунке 1

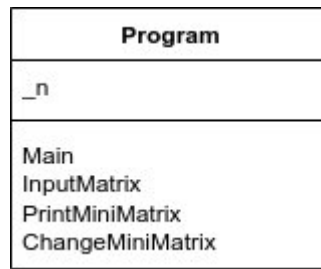


Рисунок 1 — Объектная декомпозиция

Схемы алгоритмов изображены на рисунках 2 и 3

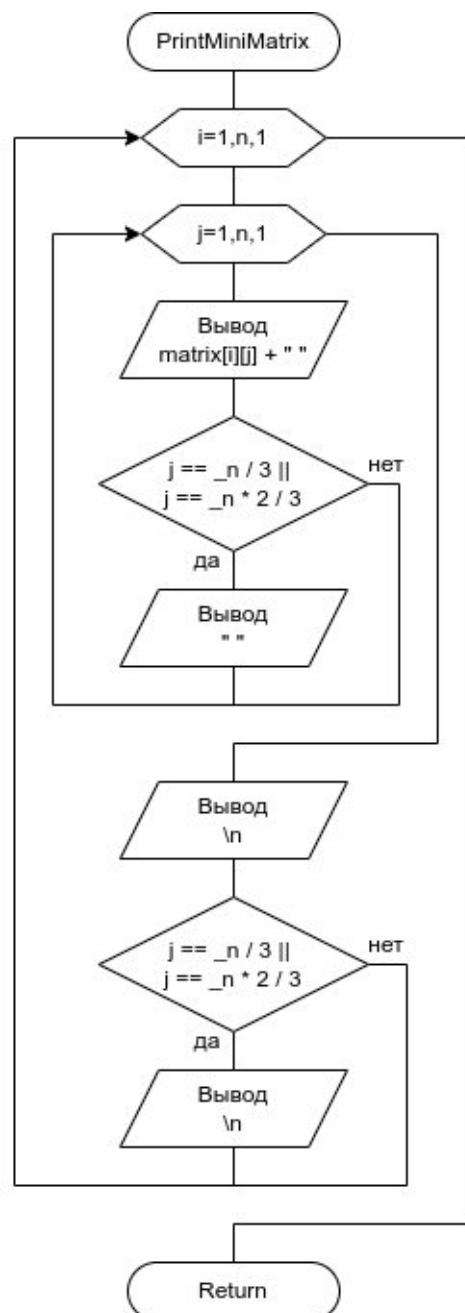


Рисунок 2 — Схема алгоритма вывода матрицы

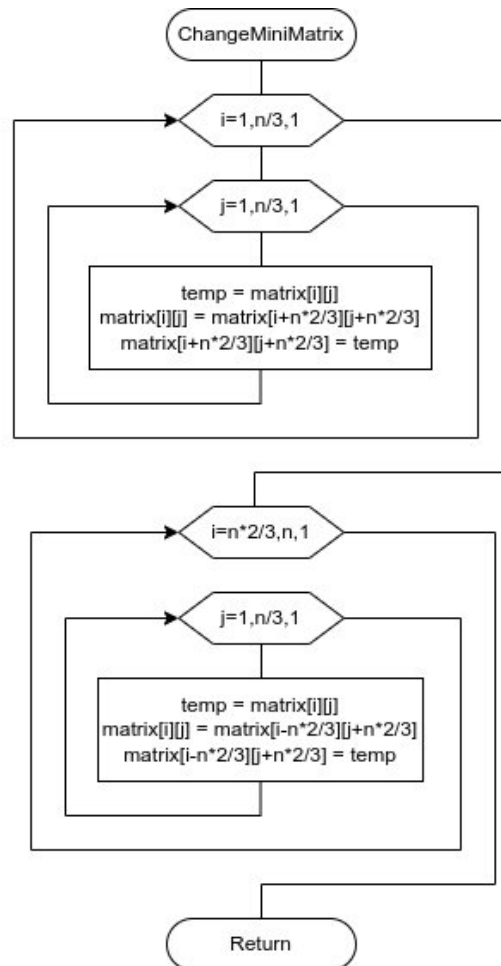


Рисунок 3 — Схема алгоритма изменения матрицы

Текст программы изображён на рисунках 4-6

```

public static int Main()
{
    Console.WriteLine("Введите n (размер матрицы, кратно 3): ");
    if (!int.TryParse(Console.ReadLine(), out _n))
    {
        Console.WriteLine("Ошибка при вводе числа!");
        return 1;
    }

    if (_n < 3 || _n % 3 != 0)
    {
        Console.WriteLine("Число должно быть положительным и кратным 3!");
        return 1;
    }

    var matrix = new int[_n][];
    if (InputMatrix(matrix) != 0) return 1;

    PrintMiniMatrix(matrix);
    ChangeMiniMatrix(matrix);
    Console.WriteLine("\n");
    PrintMiniMatrix(matrix);

    return 0;
}

```

Рисунок 4 — Код основной функции

```

private static void PrintMiniMatrix(int[][] matrix)
{
    for (int i = 0; i < _n; i++)
    {
        for (int j = 0; j < _n; j++)
        {
            Console.Write(matrix[i][j] + " ");
            if (j == (_n / 3 - 1) || j == (_n / 3 * 2 - 1))
                Console.Write(" ");
        }

        Console.WriteLine();
        if (i == (_n / 3 - 1) || i == (_n / 3 * 2 - 1))
            Console.WriteLine();
    }
}

```

Рисунок 5 — Код вывода матрицы

```

private static void ChangeMiniMatrix(int[][] matrix)
{
    // Меняем матрицы на главной диагонали
    for (var i = 0; i < _n / 3; i++)
    {
        for (var j = 0; j < _n / 3; j++)
        {
            var temp:int = matrix[i][j];
            matrix[i][j] = matrix[i + _n * 2 / 3][j + _n * 2 / 3];
            matrix[i + _n * 2 / 3][j + _n * 2 / 3] = temp;
        }
    }

    // Меняем матрицы на побочной диагонали
    for (var i:int = _n * 2 / 3; i < _n; i++)
    {
        for (var j = 0; j < _n / 3; j++)
        {
            var temp:int = matrix[i][j];
            matrix[i][j] = matrix[i - _n * 2 / 3][j + _n * 2 / 3];
            matrix[i - _n * 2 / 3][j + _n * 2 / 3] = temp;
        }
    }
}

```

Рисунок 6 — Код изменения матрицы

Тестовые данные и результаты тестирования: тестовые данные и результаты тестирования изображены на рисунке 7

Ввод	Вывод
3 1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9 9 2 7 4 5 6 3 8 1
6 1 4 6 4 2 3 1 1 2 4 5 6 9 9 4 1 3 4 6 3 3 8 9 2 4 5 6 9 8 4 1 2 9 3 4 5	1 4 6 4 2 3 1 1 2 4 5 6 9 9 4 1 3 4 6 3 3 8 9 2 4 5 6 9 8 4 1 2 9 3 4 5 8 4 6 4 4 5 4 5 2 4 1 2 9 9 4 1 3 4 6 3 3 8 9 2 2 3 6 9 1 4 5 6 9 3 1 1

Рисунок 7 — Тестовые данные и результаты тестирования

Вывод: в данной лабараторной работе была создана программа на C#, использующая массивы и выполняющая операции над матрицами, реализованных с их помощью.

Часть 2. Строки

Цель работы: изучить строки в C#.

Выполнение: объектная декомпозиция изображена на рисунке 8.

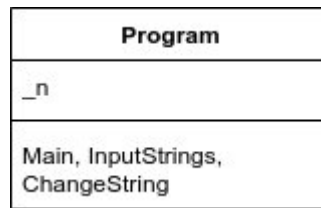


Рисунок 8 — Объектная декомпозиция

Схема основного алгоритма изображена на рисунке 9

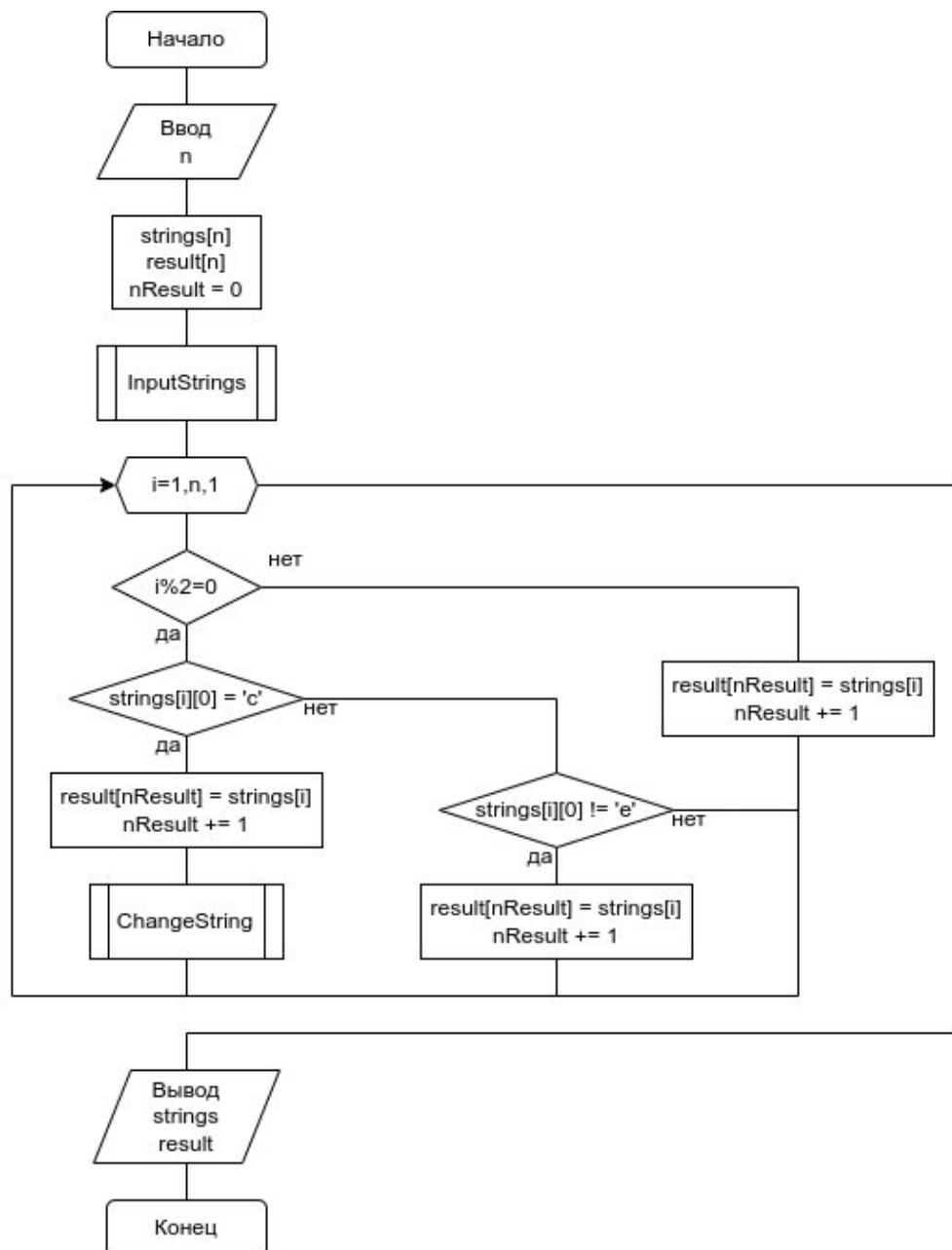


Рисунок 9 — Схема основного алгоритма программы

Код программы изображен на рисунках 10-12

```
private static int Main()
{
    Console.WriteLine("Введите число строк: ");
    if (!int.TryParse(Console.ReadLine(), out _n)) {...}

    if (_n <= 0) {...}

    var strings = new string[_n];
    if (InputStrings(strings) != 0)
        return 1;

    var result = new string[_n];
    var nResult = 0;
    for (var i = 0; i < _n; i++)
    {
        if ((i + 1) % 2 == 0)
        {
            if (strings[i][0] == 'c')
            {
                result[nResult++] = strings[i];
                if (ChangeString(ref result[nResult - 1]) != 0) return 1;
            }
            else if (strings[i][0] != 'e')
                result[nResult++] = strings[i];
        }
        else
        {
            result[nResult++] = strings[i];
        }
    }

    Console.WriteLine("Исходная последовательность:");
    foreach (var s :string in strings)
        Console.WriteLine(s + " ");
    Console.WriteLine("\nИзменённая последовательность:");
    for (int i = 0; i < nResult; i++)
        Console.WriteLine(result[i] + " ");

    return 0;
}
```

Рисунок 10 — Код функции Main

```

private static int InputStrings(string[] strings)
{
    Console.WriteLine("Введите строки:");
    for (var i = 0; i < _n; i++)
    {
        var input :string? = Console.ReadLine();
        if (input == null)
        {
            Console.WriteLine("Ошибка при вводе строки!");
            return 1;
        }

        strings[i] = input;
    }

    return 0;
}

```

Рисунок 11 — Код функции ввода строк

```

private static int ChangeString(ref string s)
{
    Console.Write($"Введите строку, замещающую {s}: ");
    var input :string? = Console.ReadLine();
    if (input == null)
    {
        Console.WriteLine("Ошибка при вводе строки!");
        return 1;
    }

    s = input;
    return 0;
}

```

Рисунок 12 — Код изменения строки

Тестовые данные и результаты тестирования: тестовые данные и результаты тестирования изображены на рисунке 13.

Ввод	Вывод
2 hello world	Исходная последовательность: hello world Изменённая последовательность: hello world
4 hello e_word_to_delete c_but_ignore c_word_to_change	Введите строку, замещающую c_word_to_change: new_string Исходная последовательность: hello e_word_to_delete c_but_ignore c_word_to_change Изменённая последовательность: hello c_but_ignore new_string

Рисунок 13 — Тестовые данные и результаты тестирования

Вывод: при выполнении данной лабораторной работы были изучены строки в C#, а также базовые операции с ними.