**3 Процедуры и функции – методы класса**

Задание №1. Описать класс для работы с двумерным массивом строк. Обеспечить перегрузку операции + для построчного соединения элементов.

Листинг программы:

using System;

class StringMatrix

{

private string[,] matrix;

private int rows;

private int cols;

public StringMatrix(int rows, int cols)

{

this.rows = rows;

this.cols = cols;

matrix = new string[rows, cols];

}

public string this[int i, int j]

{

get => matrix[i, j];

set => matrix[i, j] = value;

}

public static StringMatrix operator +(StringMatrix a, StringMatrix b)

{

if (a.rows != b.rows || a.cols != b.cols)

throw new ArgumentException("Размеры матриц должны совпадать!");

StringMatrix result = new StringMatrix(a.rows, a.cols);

for (int i = 0; i < a.rows; i++)

{

for (int j = 0; j < a.cols; j++)

{

result[i, j] = $"{a[i, j]} {b[i, j]}";

}

}

return result;

}

public void Print()

{

for (int i = 0; i < rows; i++)

{

for (int j = 0; j < cols; j++)

{

Console.Write(matrix[i, j].PadRight(15));

}

Console.WriteLine();

}

}

}

class Program

{

static void Main()

{

StringMatrix matrix1 = new StringMatrix(2, 2)

{

[0, 0] = "Привет",

[0, 1] = "Мир",

[1, 0] = "Как",

[1, 1] = "дела?"

};

StringMatrix matrix2 = new StringMatrix(2, 2)

{

[0, 0] = "!",

[0, 1] = "?",

[1, 0] = "у",

[1, 1] = "тебя."

};

StringMatrix result = matrix1 + matrix2;

Console.WriteLine("Результирующая матрица:");

result.Print();

}

}

Таблица 3.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Привет Мир Как дела? ! ? у тебя. | Результирующая матрица |

Анализ результатов:

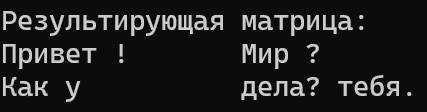


Рисунок 3.1 – Результаты работы программы