**Лабораторная работа №18**

**РАБОТА С МАТРИЦАМИ**

**Цели:**

1.Познакомиться с алгоритмами ввода-вывода матриц в C#.

2.Закрепить навыки работы с текстовыми файлами

**Приложение Lab18\_01 Разработка и использование подпрограмм ввода-вывода двумерного массива вещественных чисел**

***Задание*:** Реализовать в классе Helper метод-функцию ReadMatrixOfDouble(), организующую ввод двумерного массива вещественных чисел. Добавить в класс Helper метод-процедуру WriteMatrixOfDouble(), принимающую двумерный массив в качестве параметра и выводящую его на консоль.

Сопроводить разработанные методы XML-комментариями.

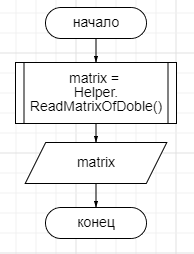
***Константы:*** отсутствуют.

***Переменные:*** matrix – переменная типа double[][] – матрица.

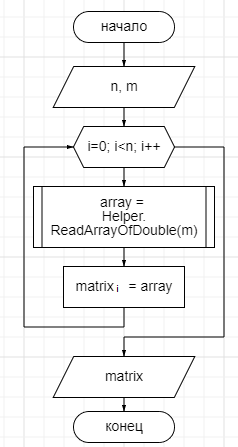
***Исходные данные:*** отсутствуют.

***Результат:*** Программа выводит матрицу.

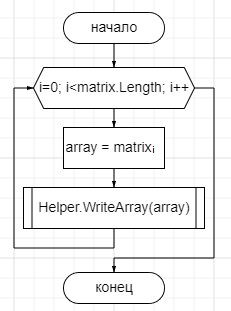
***Блок-схема:***



***Блок-схема ReadMatrixOfDouble():***



***Блок-схема WriteMatrix(double[][] matrix):***



***Код приложения:***

**Листинг 1.1 – Код консольного приложения Lab18\_01**

namespace PuzanovVE.OP.Lab18\_01

{

class Program

{

static void Main(string[] args)

{

//Стандартная шапка

int lab = 18; int app = 1; string info = "Вывод матрицы";

Helper.Head(lab, app, info);

//основной код программы

double[][] matrix = Helper.ReadMatrixOfDouble();

Helper.WriteMatrix(matrix);

Console.ReadLine();

}

}

}

**Листинг 1.2 – Код функции ReadMatrixOfDouble()**

/// <params>

/// Запись матрицы с клавиатуры

/// </params>

/// <returns>

/// Возвращает матрицу

/// </returns>

public static double[][] ReadMatrixOfDouble()

{

Console.WriteLine("Введите кол-во строк в матрице");

int n = int.Parse(Console.ReadLine());

Console.WriteLine("Введите кол-во столбцов в матрице");

int m = int.Parse(Console.ReadLine());

double[][] matrix = new double[n][];

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

{

Console.WriteLine($"{i + 1} строка:");

double[] array = Helper.ReadArrayOfDouble(m);

matrix[i] = array;

}

return matrix;

}

**Листинг 1.3 – Код функции WriteMatrix(double[][] matrix)**

/// <params>

/// Вывод матрицы

/// </params>

/// <param name="matrix">

/// Считываемая матрица

/// </param>

public static void WriteMatrix(double[][] matrix)

{

for (int i = 0; i < matrix.Length; i++)

{

double[] array = matrix[i];

Helper.WriteArray(array);

Console.WriteLine();

}

}

**Тестирование:**

Контрольный пример 1:

Исходные данные: n – кол-во строк - 3; m – кол-во столбцов - 4

Результат: Программа выводит матрицу введённую с клавиатуры.

Результат работы программы для указанных исходных данных приведён на рисунке 1.1

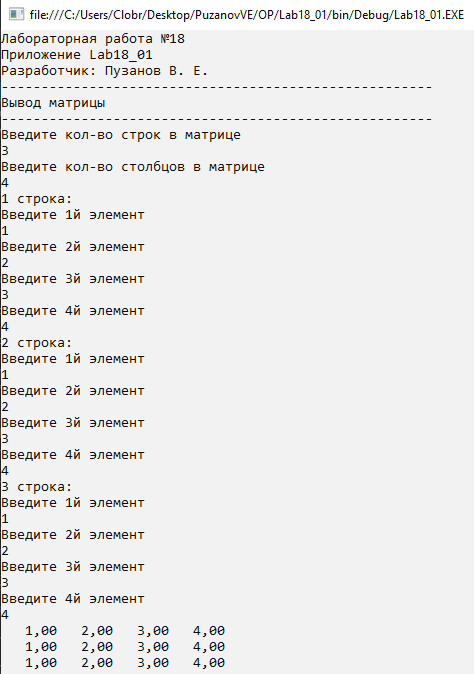


Рисунок 1.1 –Результат работы программы Lab18\_01

Контрольный пример 2:

Исходные данные: n – кол-во строк - 5; m – кол-во столбцов - 1

Результат: Программа выводит матрицу введённую с клавиатуры.

Результат работы программы для указанных исходных данных приведён на рисунке 1.1

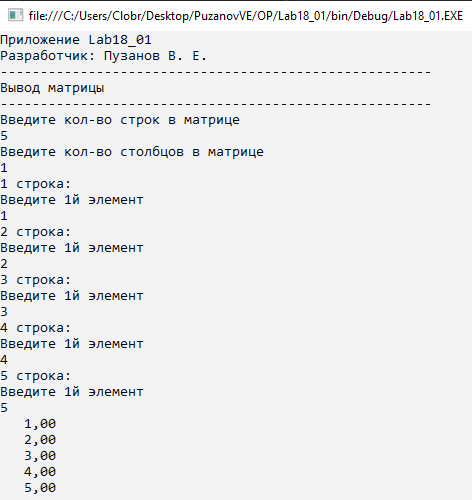


Рисунок 1.1 –Результат работы программы Lab18\_01

Контрольный пример 3:

Исходные данные: n – кол-во строк - 2; m – кол-во столбцов - 2

Результат: Программа выводит матрицу введённую с клавиатуры.

Результат работы программы для указанных исходных данных приведён на рисунке 1.1

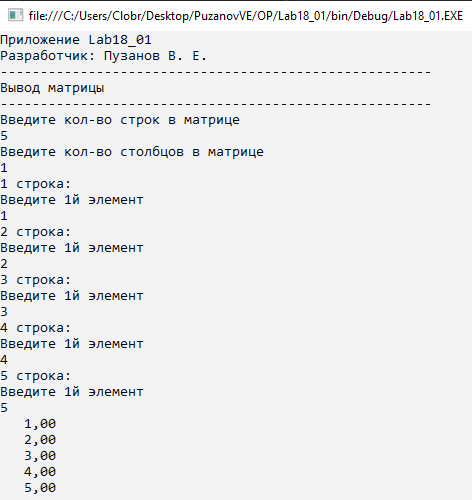


Рисунок 1.1 –Результат работы программы Lab18\_01

**Приложение Lab18\_02.Разработка и использование метода записи двумерного массива в текстовый файл**

***Задание*:** Добавить в класс Helper процедуру WriteMatrixFile(), выводящую двумерный массив в указанный текстовый файл. Процедура должна принимать в качестве параметров путь к файлу и двумерный ступенчатый массив вещественных чисел. Сопроводить разрабатываемый метод XML-комментариями.

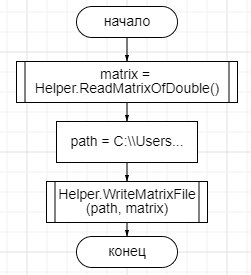
***Константы:*** path – переменная типа string – хранит в себе ссылку на файл.

***Переменные:*** matrix – переменная типа double[][] – матрица.

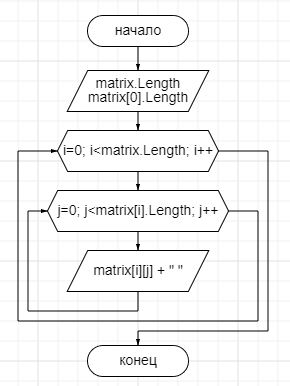
***Исходные данные:*** Введённая матрица;

***Результат:*** Программа записывает матрицу в txt файл.

***Блок-схема:***



***Блок-схема WriteMatrixFile(string path, double[][] matrix):***



***Код приложения:***

**Листинг 2.1 – Код консольного приложения Lab18\_02**

namespace PuzanovVE.OP.Lab18\_02

{

class Program

{

static void Main(string[] args)

{

//Стандартная шапка

int lab = 18; int app = 2; string info = " Вывод матрицы в txt файл";

Helper.Head(lab, app, info);

//основной код программы

double[][] matrix = Helper.ReadMatrixOfDouble();

string path = "C:\\Users\\Clobr\\Desktop\\PuzanovVE\\matrix.txt";

Helper.WriteMatrixFile(path, matrix);

Console.ReadLine();

}

}

}

**Листинг 2.2 – Код функции WriteMatrixFile(string path, double[][] matrix)**

/// <params>

/// Выводит матрицу вещественных чисел в текстовый файл

/// </params>

/// <param name="matrix">

/// Матрица, выводимая в текстовый файл

/// </param>

/// <param name="path">

/// Путь к файлу

/// </param>

public static void WriteMatrixFile(string path, double[][] matrix)

{

StreamWriter f = new StreamWriter(path, false);

f.WriteLine(matrix.Length);

f.WriteLine(matrix[0].Length);

for (int i = 0; i < matrix.Length; i++)

{

for (int j = 0; j < matrix[i].Length; j++)

{

f.Write(matrix[i][j] + " ");

}

f.WriteLine();

}

f.Close();

}

**Тестирование:**

Контрольный пример 1:

Исходные данные:

Введённая в консоль матрица

Результат:

Программа записывает матрицу в txt файл.

Результат работы программы для указанных исходных данных приведён на рисунках (2.1.1, 2.1.2)

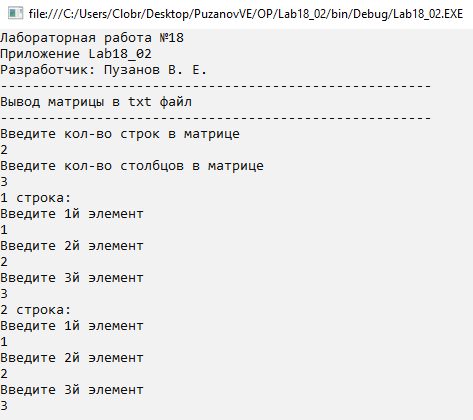


Рисунок 2.1.1 –Результат работы программы Lab18\_02

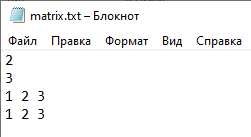


Рисунок 2.1.2 –txt файл после выполнения программы Lab18\_03

Контрольный пример 2:

Исходные данные:

Введённая в консоль матрица

Результат:

Программа записывает матрицу в txt файл.

Результат работы программы для указанных исходных данных приведён на рисунках (2.2.1, 2.2.2)

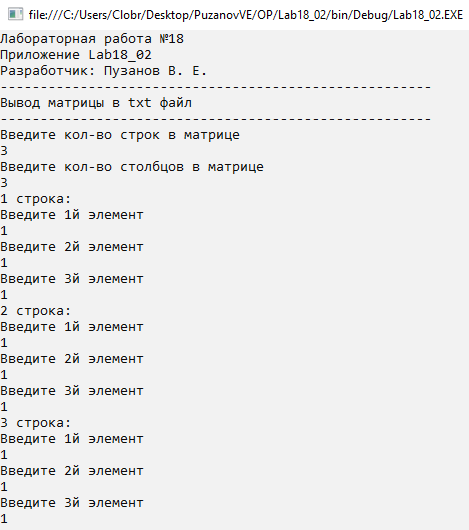


Рисунок 2.2.1 –Результат работы программы Lab18\_02

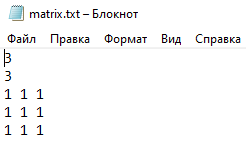


Рисунок 2.2.2 –txt файл после выполнения программы Lab18\_03

Контрольный пример 3:

Исходные данные:

Введённая в консоль матрица

Результат:

Программа записывает матрицу в txt файл.

Результат работы программы для указанных исходных данных приведён на рисунках (2.3.1, 2.3.2)

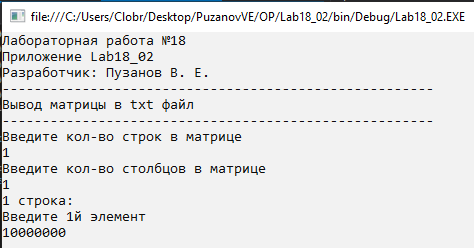


Рисунок 2.3.1 –Результат работы программы Lab18\_02

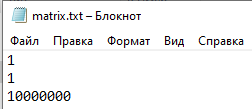


Рисунок 2.3.2 –txt файл после выполнения программы Lab18\_03

**Приложение Lab18\_03.Разработка и использование метода чтения двумерного массива из текстового файла**

***Задание*:** Реализовать в классе Helper функцию ReadMatrixFile(), читающую двумерный массив из указанного текстового файла. Функция должна принимать путь к файлу в качестве параметра и возвращать в точку вызова считанный двумерный массив (матрицу). Сопроводить метод-функцию XML- комментариями.

Разработать консольное приложение, которое, используя разработанный метод ReadMatrixFile(), читало бы из файла «H:\матрица.txt» двумерный массив и выводило его на консоль в виде таблицы.

***Константы:*** path – переменная типа string – хранит в себе ссылку на txt файл.

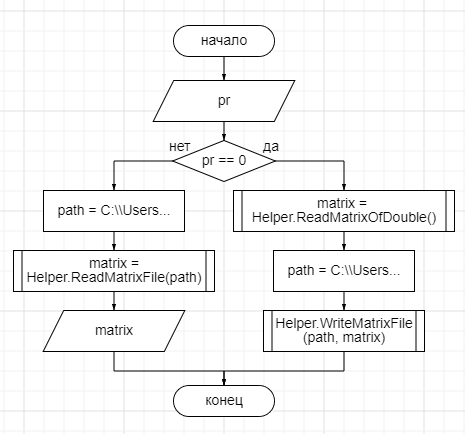
***Переменные:*** matrix – матрица; pr – переменная типа int - признак;

***Исходные данные:*** pr.

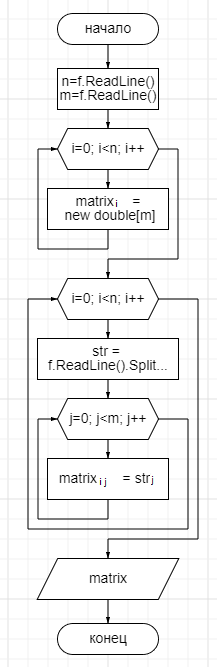
***Результат:***

Программа считывает массив из txt файла.

***Блок-схема:***



***Блок-схема ReadMatrixFile(path):***



***Код приложения:***

**Листинг 3.1 – Код консольного приложения Lab18\_03**

namespace PuzanovVE.OP.Lab18\_03

{

class Program

{

static void Main(string[] args)

{

//Стандартная шапка

int lab = 18; int app = 3; string info = "Вывод матрицы в txt файл";

Helper.Head(lab, app, info);

//основной код программы

Console.WriteLine("Что вы хотите сделать?\n0 - записать матрицу в файл\n1 - считать матрицу из файла");

int pr = int.Parse(Console.ReadLine());

if (pr == 0)

{

double[][] matrix = Helper.ReadMatrixOfDouble();

string path = "C:\\Users\\Clobr\\Desktop\\PuzanovVE\\matrix.txt";

Helper.WriteMatrixFile(path, matrix);

}

else

{

string path = "C:\\Users\\Clobr\\Desktop\\PuzanovVE\\matrix.txt";

double[][] matrix;

matrix = Helper.ReadMatrixFile(path);

Helper.WriteMatrix(matrix);

}

Console.ReadLine();

}

}

}

**Листинг 3.2 – ReadMatrixFile(string path)**

/// <params>

/// Ввод матрицы из txt файла

/// </params>

/// <param name="path">

/// Ссылка на файл

/// </param>

/// <returns>

/// Матрица, считанная из файла

/// </returns>

public static double[][] ReadMatrixFile(string path)

{

StreamReader f = new StreamReader(path);

int n = int.Parse(f.ReadLine());

int m = int.Parse(f.ReadLine());

double[][] matrix = new double[n][];

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

{

matrix[i] = new double[m];

}

string[] str = new string[m];

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

{

//StringSplitOptions.RemoveEmptyEntries запретит создание пустых строк.

str = f.ReadLine().Split(new char[] { ' ' }, StringSplitOptions.RemoveEmptyEntries);

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

{

matrix[i][j] = double.Parse(str[j]);

}

}

f.Close();

return matrix;

}

}

**Тестирование:**

Контрольный пример 1:

Исходные данные:

Массив записанный в txt файле.

Результат:

Вывод матрицы на консоль.

Результат работы программы для указанных исходных данных приведён на рисунках (3.1.1 - 3.1.2)

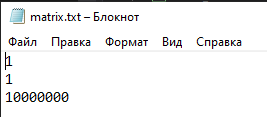


Рисунок 3.1.1 – txt файл перед выполнением программы Lab18\_03

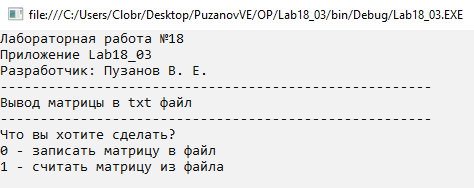


Рисунок 3.1.2 –Результат работы программы Lab18\_03

Контрольный пример 2:

Исходные данные:

Массив записанный в txt файле.

Результат:

Вывод матрицы на консоль.

Результат работы программы для указанных исходных данных приведён на рисунках (3.2.1 - 3.2.2)

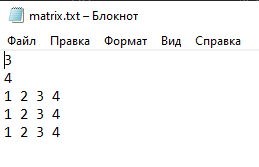


Рисунок 3.2.1 – txt файл перед выполнением программы Lab18\_03

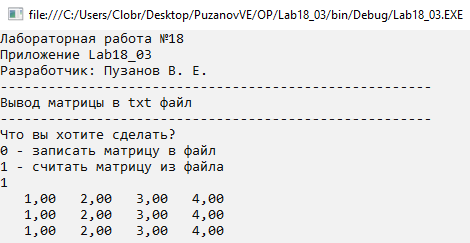


Рисунок 3.2.2 –Результат работы программы Lab18\_03

Контрольный пример 2:

Исходные данные:

Массив записанный в txt файле.

Результат:

Вывод матрицы на консоль.

Результат работы программы для указанных исходных данных приведён на рисунках (3.2.1 - 3.2.2)

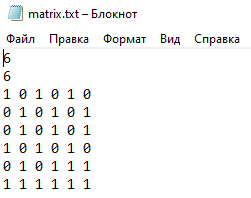


Рисунок 3.3.1 – txt файл перед выполнением программы Lab18\_03

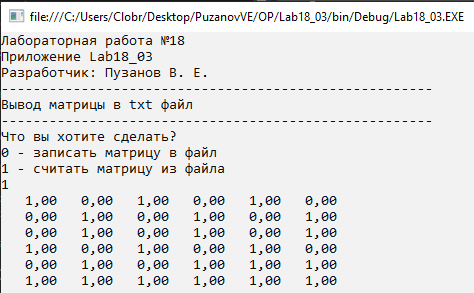


Рисунок 3.3.2 –Результат работы программы Lab18\_03

Выполнил студент Пузанов В. Е., ФИТУ 010304-КМСб-о22

Проверил ст. преподаватель каф. ПМ Черноиван Д.Н.