Министерство образования и науки Российской Федерации Федеральное‌ ‌государственное‌ ‌бюджетное‌ ‌образовательное‌ ‌учреждение‌ высшего‌ ‌образования‌

**«Пермский национальный исследовательский политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**ОТЧЁТ**

**по лабораторной работе №6**

Дисциплина: «Объектно-ориентированное программирование»

Тема: Класс Array. Строки. Класс String

Вариант 11

Выполнил работу

студент группы РИС-22-1б

Вышенская Екатерина

Проверила

Доцент кафедры ИТАС

Викентьева О.Л.

Пермь, 2023

**Постановка задачи**

**Постановка задачи 1.**

1. Создать динамический массив (одномерный, двумерный, рваный) из элементов заданного типа. При заполнении массива использовать 2 способа (ручной и с помощью ДСЧ).
2. Массив вывести на печать.
3. Выполнить операции с массивом, указанные в варианте, используя, по возможности, методы класса Array.
4. Результаты обработки вывести на печать.

**Постановка задачи 2.**

1. Ввести строку символов с клавиатуры. Строка состоит из слов, разделенных пробелами (пробелов может быть несколько) и знаками препинания (, ;:). В строке может быть несколько предложений, в конце каждого предложения стоит знак препинания (.!?).
2. Выполнить обработку строки в соответствии с вариантом.
3. Результаты обработки вывести на печать.

|  |  |  |  |
| --- | --- | --- | --- |
| 11 | Двумерный | double | Удалить все строки, в которых есть число, совпадающее с максимальным элементом. |

|  |  |
| --- | --- |
| 11 | Удалить первое и последнее предложение в строке. |

**Анализ задачи**

**Задача 1**

Основные этапы:

* Создание массива заданного типа (одномерного, двумерного, рваного)
* Заполнение массива элементами двумя способами: вручную и случайным образом
* Вывод массива на печать
* Выполнение заданных операций над элементами массива с использованием методов класса Array
* Вывод результатов на печать

**Задача 2**

Основные этапы:

* Ввод строки с клавиатуры
* Разбиение строки на отдельные слова по заданным разделителям (пробел, знаки препинания)
* Выполнение заданных операций над словами (в зависимости от условия)
* Объединение полученных слов в результирующую строку
* Вывод результата на печать

**Работа программы**

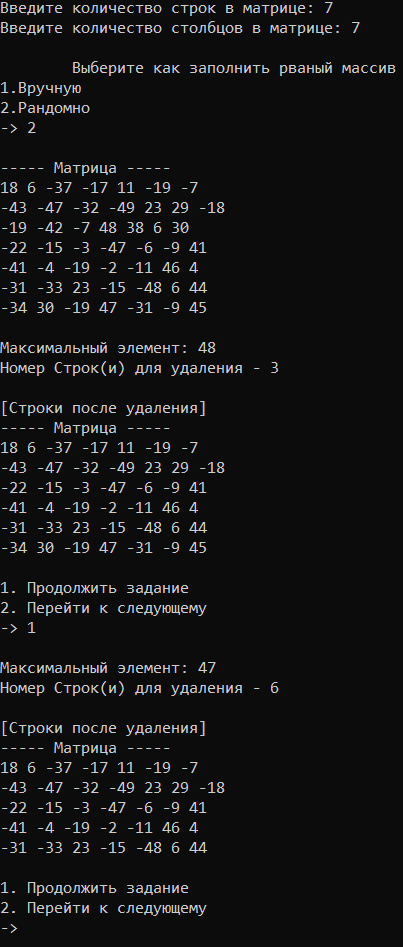


Рисунок 1 – Результат работы первой части задания

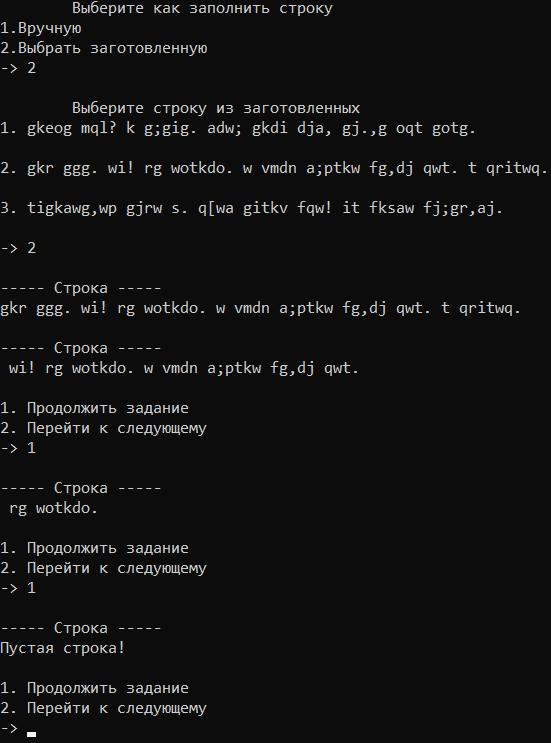


Рисунок 2 – Выбор заготовленной строки

**Тестирование программы**

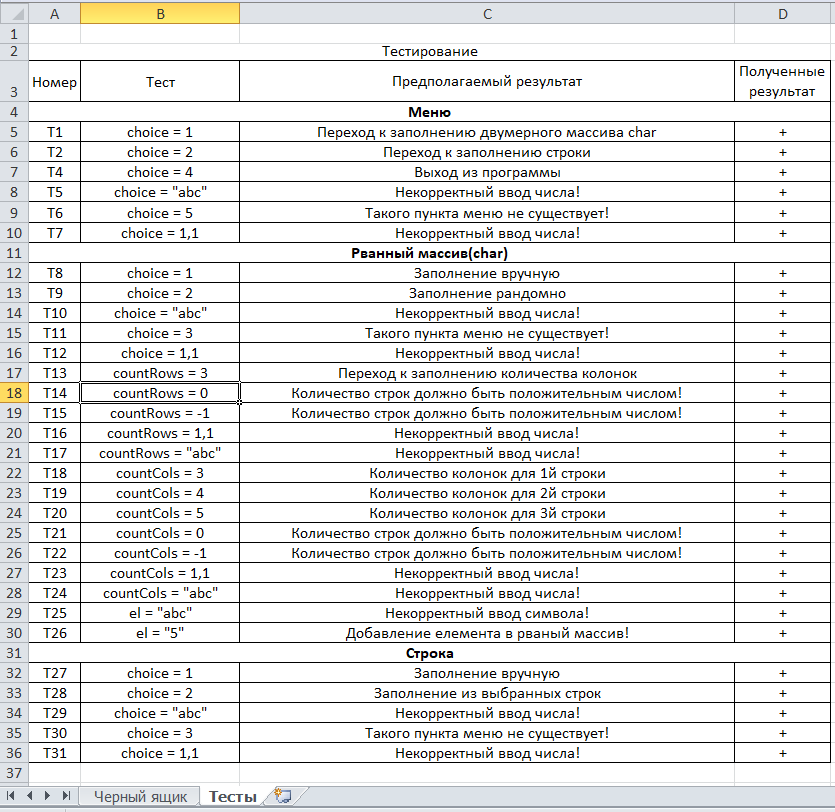


Таблица 1 – Тестирование программы

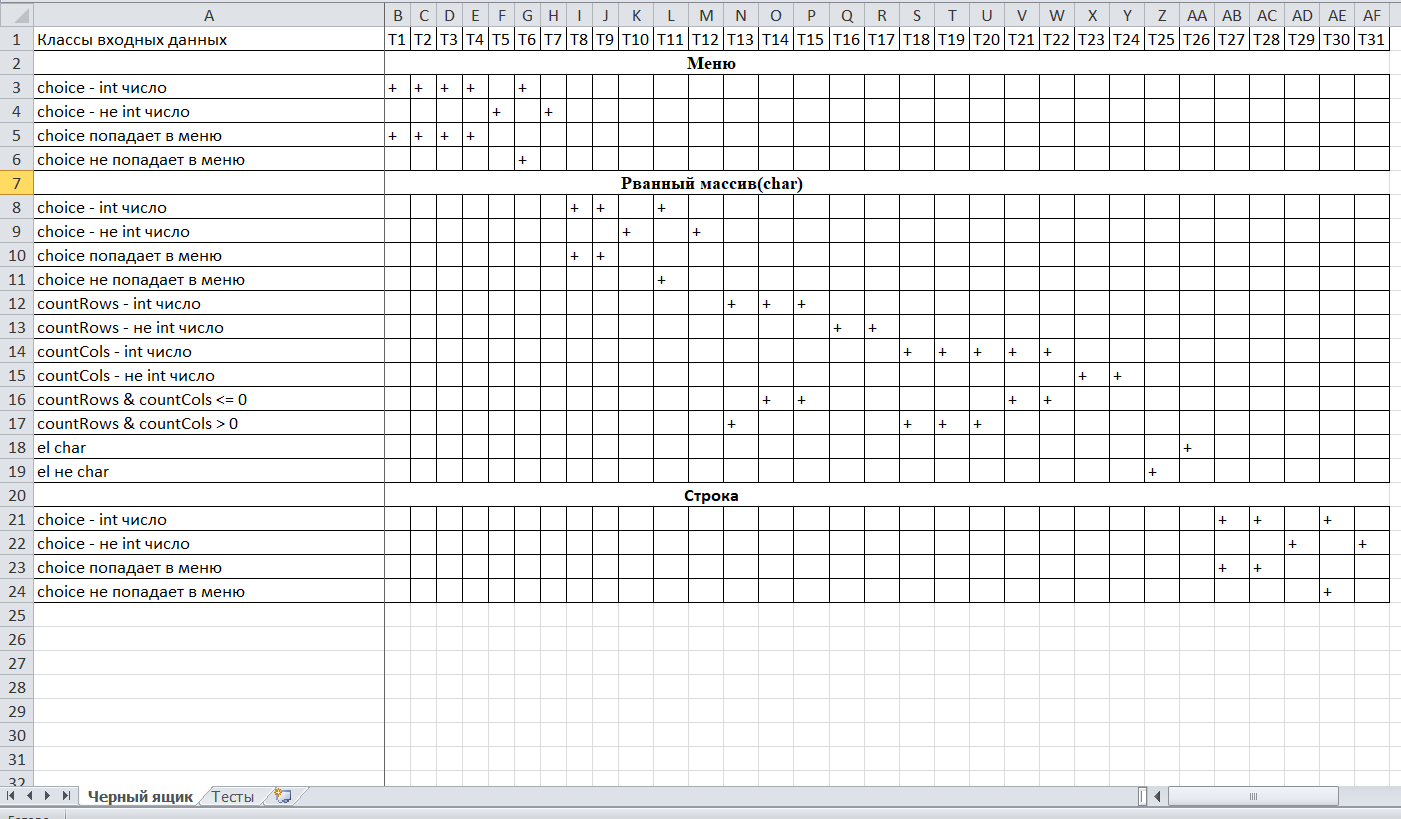


Таблица 2 – Тестирование программы черный ящик.

**Код программы**

**Файл – Str.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Reflection;

using System.Text;

using System.Text.RegularExpressions;

using System.Threading.Tasks;

namespace Laba\_6

{

internal class Str

{

string? str;

public Str()

{

str = null;

}

public Str(string str)

{

this.str = str;

}

public void ManualFill()

{

Console.WriteLine("\tВведите строку");

str = Console.ReadLine();

Console.WriteLine();

}

public void PreparedLines()

{

string str1 = "gkeog mql? k g;gig. adw; gkdi dja, gj.,g oqt gotg.";

string str2 = "gkr ggg. wi! rg wotkdo. w vmdn a;ptkw fg,dj qwt. t qritwq.";

string str3 = "tigkawg,wp gjrw s. q[wa gitkv fqw! it fksaw fj;gr,aj.";

string msg = $"\tВыберите строку из заготовленных\n1. {str1}\n\n2. {str2}\n\n3. {str3}\n\n-> ", errorMsg = "\nТакого пункта меню не существует!\n";

bool notExit = true;

do

{

int choice = CheckAndInput.InputIntNumber(msg);

if (choice == 1)

{

Console.WriteLine();

str = str1;

notExit = false;

}

else if (choice == 2)

{

Console.WriteLine();

str = str2;

notExit = false;

}

else if (choice == 3)

{

Console.WriteLine();

str = str3;

notExit = false;

}

else

Console.WriteLine(errorMsg);

} while (notExit);

}

public void ShowStr()

{

Console.WriteLine("----- Строка -----");

if (str == null || str.Length == 0)

{

Console.WriteLine("Пустая строка!\n");

return;

}

Console.WriteLine(str + '\n');

}

public void DeleteSentences()

{

if (str == null || str.Length == 0)

{

Console.WriteLine("Нет предложений для удаления!");

return;

}

string[] sentences = SplitIntoSentences();

string tempStr = "";

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

{

if (i != 0 && i != sentences.Length - 2)

tempStr += sentences[i];

}

str = tempStr;

}

public string[] SplitIntoSentences()

{

string pattern = @"(?<=[.!?])";

string[] sentences = Regex.Split(str, pattern);

return sentences;

}

public void ContinuePartTwo()

{

string msg = "1. Продолжить задание\n2. Перейти к следующему\n-> ", errorMsg = "\nТакого пункта меню не существует!\n";

bool notExit = true;

do

{

int choice = CheckAndInput.InputIntNumber(msg);

if (choice == 1)

{

Console.WriteLine();

DeleteSentences();

ShowStr();

ContinuePartTwo();

notExit = false;

}

else if (choice == 2)

return;

else

Console.WriteLine(errorMsg);

} while (notExit);

}

}

}

**Файл – Matrix.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Text.RegularExpressions;

using System.Threading.Tasks;

namespace Laba\_6

{

internal class Matrix

{

static Random random = new Random();

double[,] matrix;

public Matrix()

{

int countRows = 0, countCols = 0;

string msg = "";

string errorMsg = "";

msg = "Введите количество строк в матрице: ";

errorMsg = "\nКоличество строк должно быть положительным числом!\n";

countRows = CheckAndInput.InputPositiveNumber(msg, errorMsg);

errorMsg = "\nКоличество столбцов должно быть положительным числом!\n";

msg = "Введите количество столбцов в матрице: ";

countCols = CheckAndInput.InputPositiveNumber(msg, errorMsg);

matrix = new double[countRows, countCols];

Console.WriteLine();

}

public void RandomFill()

{

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

{

for (int j = 0; j < matrix.GetLength(1); j++)

matrix[i, j] = random.Next(-50, 50);

}

}

public void ManualFill()

{

string? msg;

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

{

for (int j = 0; j < matrix.GetLength(1); j++)

{

msg = $"Введите элемент для {i + 1}-й строки {j + 1}-го столбца: ";

matrix[i, j] = CheckAndInput.InputIntNumber(msg);

}

}

Console.WriteLine();

}

public void ShowMatrix()

{

if (matrix.Length == 0)

{

Console.WriteLine("Матрица: <пустая>");

return;

}

Console.WriteLine("----- Матрица -----");

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

{

for (int j = 0; j < matrix.GetLength(1); j++)

Console.Write($"{matrix[i, j]} ");

Console.WriteLine();

}

}

public void DeleteRows()

{

if (matrix.Length == 0)

{

Console.WriteLine("\nНет строк для удаления!\n");

return;

}

double maxElement = matrix.Cast<double>().Max();

Console.WriteLine($"\nМаксимальный элемент: {maxElement}");

List<int> indexRowsToDelete = new List<int>();

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

{

for (int j = 0; j < matrix.GetLength(1); j++)

{

//преобразование коллекции в одномерный массив

if (matrix[i, j] == maxElement)

{

indexRowsToDelete.Add(i);

break;

}

}

}

if (indexRowsToDelete.Count == 0)

{

Console.WriteLine("Не нашлось строк для удаления!\n");

return;

}

else

{

Console.Write("Номер Строк(и) для удаления - ");

for (int i = 0; i < indexRowsToDelete.Count; i++)

{

if (i != indexRowsToDelete.Count - 1)

Console.Write($"{indexRowsToDelete[i] + 1},");

else

Console.Write($"{indexRowsToDelete[i] + 1}");

}

}

Console.WriteLine("\n\n[Строки после удаления]");

double[,] temp = new double[matrix.GetLength(0) - indexRowsToDelete.Count, matrix.GetLength(1)];

for (int i = 0, z = 0; i < matrix.GetLength(0); i++)

{

if (!indexRowsToDelete.Contains(i))

{

for (int j = 0; j < matrix.GetLength(1); j++)

temp[z, j] = matrix[i, j];

z++;

}

}

matrix = temp;

}

public void ContinuePartOne()

{

string msg = "\n1. Продолжить задание\n2. Перейти к следующему\n-> ", errorMsg = "\nТакого пункта меню не существует!\n";

bool notExit = true;

do

{

int choice = CheckAndInput.InputIntNumber(msg);

if (choice == 1)

{

DeleteRows();

ShowMatrix();

ContinuePartOne();

notExit = false;

}

else if (choice == 2)

return;

else

Console.WriteLine(errorMsg);

} while (notExit);

}

}

}

**Файл – CheckAndInput.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Laba\_6

{

internal static class CheckAndInput

{

static public int InputIntNumber(string msg, string errorMsg = "\nНекорректный ввод числа!\n")

{

string? strEl;

int el = 0;

do

{

Console.Write(msg);

strEl = Console.ReadLine();

if (!int.TryParse(strEl, out el))

Console.WriteLine(errorMsg);

} while (!int.TryParse(strEl, out el));

return el;

}

static public int InputPositiveNumber(string msg, string errorMsg)

{

int el = 0;

do

{

el = CheckAndInput.InputIntNumber(msg);

if (el < 1)

Console.WriteLine(errorMsg);

} while (el < 1);

return el;

}

static public int InputWithinBoundariesArray(int[] array, string msg, string errorMsg) //ввод в границах массива

{

int el = 0;

do

{

el = CheckAndInput.InputIntNumber(msg);

if (el < 1 || el > array.Length + 1)

Console.WriteLine(errorMsg);

} while (el < 1 || el > array.Length + 1);

return el;

}

}

}

**Файл – Menu.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Laba\_6

{

internal static class Menu

{

static public void CallMenu()

{

bool isNonExit = true;

int choice = 0;

string msg = "", errorMsg = "\nТакого пункта меню не существует!\n";

do

{

Console.Clear();

Console.WriteLine("----- Добро пожаловать в МЕНЮ программы! -----\n");

msg = "1. Удалить все строки, в которых есть число, совпадающее с максимальным элементом.\n" +

"2. Удалить первое и последнее предложение в строке.\n" +

"3. Завершить работу программы.\n-> ";

choice = CheckAndInput.InputIntNumber(msg);

switch (choice)

{

case 1:

Console.Clear(); Matrix matrix = new Matrix(); choiceFill(matrix); matrix.ShowMatrix(); matrix.DeleteRows(); matrix.ShowMatrix(); matrix.ContinuePartOne(); Console.WriteLine(); break;

case 2:

Console.Clear(); Str str = new Str(); choiceFill(str); str.ShowStr(); str.DeleteSentences(); str.ShowStr(); str.ContinuePartTwo(); break;

case 3:

isNonExit = false; break;

default:

Console.WriteLine(errorMsg); break;

}

if (isNonExit)

{

Console.Write("Нажмите любую клавишу для продолжения. . .");

Console.ReadKey();

}

} while (isNonExit);

}

static public void choiceFill(Matrix matrix)

{

string msg = "\tВыберите как заполнить рваный массив\n1.Вручную\n2.Рандомно\n-> ", errorMsg = "\nТакого пункта меню не существует!\n";

bool notExit = true;

do

{

int choice = CheckAndInput.InputIntNumber(msg);

if (choice == 1)

{

Console.WriteLine();

matrix.ManualFill();

notExit = false;

}

else if (choice == 2)

{

Console.WriteLine();

matrix.RandomFill();

notExit = false;

}

else

Console.WriteLine(errorMsg);

} while (notExit);

}

static public void choiceFill(Str str)

{

string msg = "\tВыберите как заполнить строку\n1.Вручную\n2.Выбрать заготовленную\n-> ", errorMsg = "\nТакого пункта меню не существует!\n";

bool notExit = true;

do

{

int choice = CheckAndInput.InputIntNumber(msg);

if (choice == 1)

{

Console.WriteLine();

str.ManualFill();

notExit = false;

}

else if (choice == 2)

{

Console.WriteLine();

str.PreparedLines();

notExit = false;

}

else

Console.WriteLine(errorMsg);

} while (notExit);

}

}

}

**Файл – Program.cs**

Using Laba\_6;

Menu.CallMenu();