Пермский филиал федерального государственного автономного образовательного учреждения высшего образования

«Национальный исследовательский университет

«Высшая школа экономики» Пермь»

Факультет экономики, менеджмента и бизнес-информатики.

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №6

студента образовательной программы «Программная инженерия»

по направлению подготовки *09.03.04 Программная инженерия*

Чепокова Елизара Сергеевича

Руководитель:

Кандидат технических наук

О.Л. Викентьева

Пермь, 2018 год

# **1.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.

|  |  |  |  |
| --- | --- | --- | --- |
| Вариант | **Тип массива** | Тип элементов | **Операция** |
| 23 | Двумерный | char | Удалить из массива все строки, в которых нет цифр. |
|  | **Задание 2** | | |
| 23 | Перевернуть все слова в предложении и отсортировать слова по убыванию. | | |

# **2.Анализ задачи:**

Основные функции

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Функция | Входные данные | Классы входных данных | Выходные данные | Классы выходных данных |
| 1 | Формирование двумерного массива | Количество строк в массиве (stringSize)  Количество столбцов в массиве (columnSize) | stringSize – целое число> 0  columnSize – целое число> 0 | Массив из целых чисел | Массив (dvumernii\_array) |
| Символ  Целое число <=0  Вещественное число | Сообщение об ошибке | Неверный ввод / Массив пуст |
| 1.1 | Формирование массива с помощью ДСЧ | Количество строк в массиве (stringSize)  Количество столбцов в массиве (columnSize)  Количество элементов  Нижний и верхний порог для ДСЧ (ArrayMin и ArrayMax) | stringSize – целое число> 0  columnSize – целое число> 0  целое число> 0  ArrayMin и ArrayMax – любое целое число | Массив из целых чисел | Массив (dvumernii\_array) |
| Символ  Вещественное число  ArrayMin и ArrayMax – вещественное число или символ | Сообщение об ошибке | Неверный ввод / Массив пуст |
| 1.2 | Формирование массива с клавиатуры | Количество строк в массиве (stringSize)  Количество столбцов в массиве (columnSize)  Число для каждого элемента | stringSize – целое число> 0  columnSize – целое число> 0 | Массив из целых чисел | Массив (dvumernii\_array) |
| Символ  Целое число <=0  Вещественное число | Сообщение об ошибке | Неверный ввод / Массив пуст |
| 2 | Ввод строки | Строка символов, разделенных пробелами и знаками препинания.  Предложения из символов, в конце у которых знак препинания. | Символ | Строка символов или строка предложений из символов | Массив |
| Целое число  Вещественное число | Сообщение об ошибке | Я не могу обработать это предложение, введите верно |

Дополнительные функции

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Функция | Входные данные | Классы входных данных | Выходные данные | Классы выходных данных |
| Ввод целого числа с клавиатуры | Proverkavvoda | Целое число | Число | Число |
| Символ  Целое число <=0  Вещественное число | Сообщение об ошибке | Неверный ввод |
| Ввод символа с клавиатуры | Trying | Символ или число = 1 символу | Символ или число | Символ или число |
| Символ или число > 1 символу | Сообщение об ошибке | Неверный ввод |
| Печать диалога для ввода команд пользователя | Proverkavvoda | Целое число | Диалоговая таблица | Меню |
| Символ  Целое число <=0  Вещественное число | Сообщение об ошибке | Нужно выбрать из списка! |

# **3.Алгоритм программы:**

## **Функции**

Основные функции

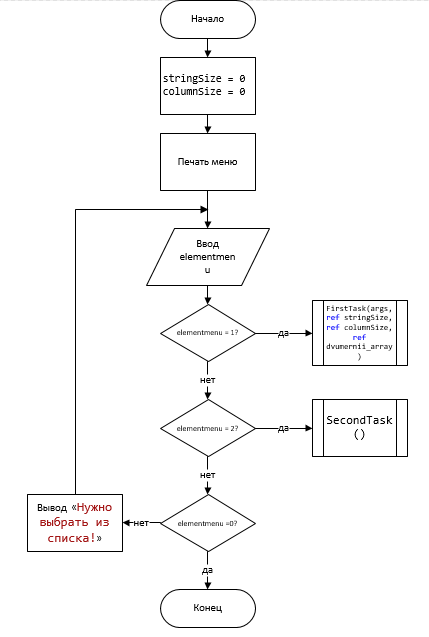
|  |  |  |  |
| --- | --- | --- | --- |
| № | Функции | Входные данные | Выходные данные |
| 1 | FirstTask (формирование двумерного массива) | int stringSize (количество строк)  int columnSize (количество столбцов) | char[,] dvumernii\_array (двумерный массив) |
| 1.1 | RanomDvumerniiArray (формирование двумерного массива с помощью ДСЧ) | int stringSize (количество строк)  int columnSize (количество столбцов) |
| 1.2 | VvodDvumerniiArray (формирование двумерного массива с клавиатуры) | int stringSize (количество строк)  int columnSize (количество столбцов) |
| 2 | SecondTask (полное выполнение 2 задания) | string str (ввод предложения) | changedString |

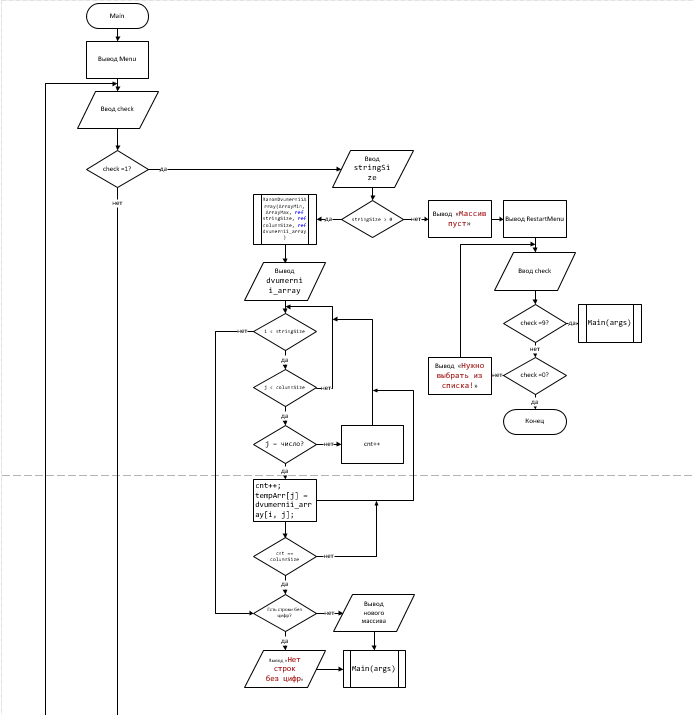
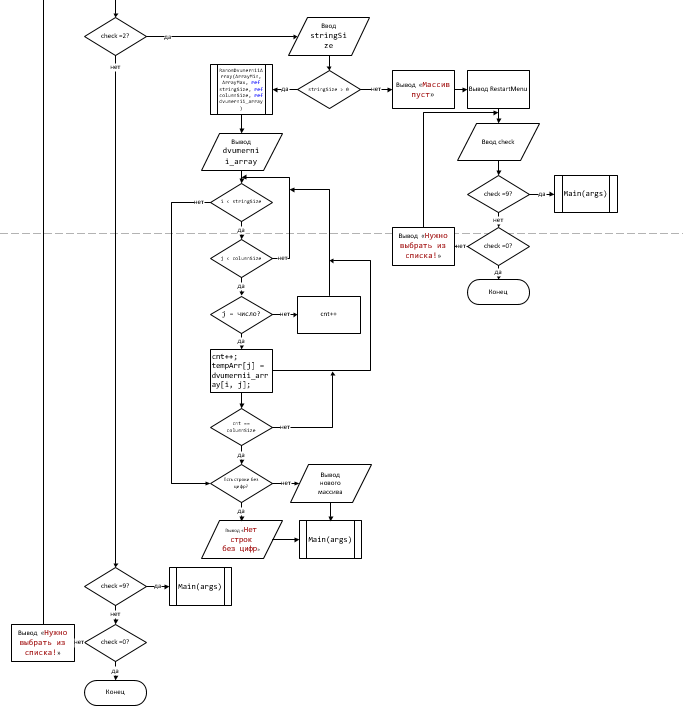
Вспомогательные функции

|  |  |  |  |
| --- | --- | --- | --- |
| № | Функции | Входные данные | Выходные данные |
| 1 | PrintErrorMenu() | int check  int elementmenu | Меню с перезапуском |
| 2 | ProverkaVvoda() | int number | Число или ошибка |
| 3 | RestartMenu | int check | Меню с перезапуском |
| 4 | Trying() | string str | char[,] dvumernii\_array  или ошибка |
| 5 | GetRandomChar() | var index  string symbols | char[,] dvumernii\_array |

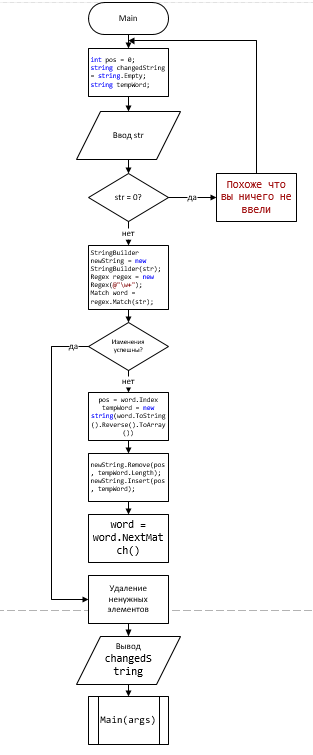
## **Блок-схема**

Блок-схема Main



Укрупнённая блок-схема FirstTask

Укрупнённая блок-схема SecondTask



# **4.Текст программы:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Text.RegularExpressions;

using System.Threading.Tasks;

namespace LAB\_6\_CH

{

class Text\_Dialog

{

//---Все менюшки----------------------------------------------

public static void PrintErrorMenu()

{

Console.ForegroundColor = ConsoleColor.Red;

Console.WriteLine("Нужно выбрать из списка!");

Console.ResetColor();

}

//------------------------------------------------------------

}

class Search\_for\_Bugs

{

//---Проверка ввода-------------------------------------------

public static int ProverkaVvoda() //Проверка ввода в массив

//Тут больше и нечего добавлять

//Просто небольшая функция на проверку

{

int number;

bool res;

do

{

res = int.TryParse(Console.ReadLine(), out number);

if (res == false)

{

Console.WriteLine("Некорректный ввод");

}

} while (!res);

return number;

}

public static char Trying()

{

Console.WriteLine("Введите букву или цифру");

while (true)

{

string str = Console.ReadLine();

if (str.Length == 1)

return str[0];

else

Console.WriteLine("Неверный ввод\nПопробуйте снова");

}

}

//------------------------------------------------------------

}

class Program

{

//---Первое задание------------------------------------------------------------------------------------------

#region

static string symbols = "1234567890abcdefghijklmnopqrstuvwxyz1234567890ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890";

static Random r = new Random();

static char GetRandomChar()

{

var index = r.Next(symbols.Length);

return symbols[index];

}

static void RestartMenu(string[] args)

{

Console.Clear();

Console.WriteLine("\n---!!!!!------Массив пуст-----!!!!!!-----");

Console.WriteLine("\n------------------------------");

Console.WriteLine("| Выберите действие: |");

Console.WriteLine("| 9) Вернуться в начало |");

Console.WriteLine("| 0) Выход из консоли |");

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

Console.Write("Действие: ");

int check = Search\_for\_Bugs.ProverkaVvoda();

switch (check)

{

case 9:

Main(args);

break;

case 0:

Environment.Exit(0); //Выход из консоли

break;

default:

Text\_Dialog.PrintErrorMenu();

break;

}

}

static void FirstTask(string[] args, ref int stringSize, ref int columnSize, ref char[,] dvumernii\_array)

{

do

{

Console.Clear();

dvumernii\_array = null;

Console.WriteLine("\n------------------------------");

Console.WriteLine("| Выберите вид массива: |");

Console.WriteLine("| 1) Рандомный массив |");

Console.WriteLine("| 2) Массив с вводом |");

Console.WriteLine("| 9) Вернуться в начало |");

Console.WriteLine("| 0) Выход из консоли |");

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

Console.Write("Действие: ");

int check = Search\_for\_Bugs.ProverkaVvoda();

switch (check)

{

case 1:

Console.Clear();

Console.WriteLine("----------Формирование массива----------------");

Console.Write("Ввдеите количество строк: ");

stringSize = Search\_for\_Bugs.ProverkaVvoda();

if (stringSize > 0)

{

RanomDvumerniiArray(args, ref stringSize, ref columnSize, ref dvumernii\_array);

}

else

{

RestartMenu(args);

}

break;

case 2:

Console.Clear();

Console.WriteLine("----------Формирование массива----------------");

Console.Write("Ввдеите количество строк: ");

stringSize = Search\_for\_Bugs.ProverkaVvoda();

if (stringSize > 0)

{

VvodDvumerniiArray(args, ref stringSize, ref columnSize, ref dvumernii\_array);

}

else

{

RestartMenu(args);

}

break;

case 9:

Main(args);

break;

case 0:

Environment.Exit(0); //Выход из консоли

break;

default:

Text\_Dialog.PrintErrorMenu();

continue;

}

} while (!true);

}

static void RanomDvumerniiArray(string[] args, ref int stringSize, ref int columnSize, ref char[,] dvumernii\_array)

{

Console.Write("\nВведите количество столбцов: ");

columnSize = Search\_for\_Bugs.ProverkaVvoda();

dvumernii\_array = new char[stringSize, columnSize];

Random rand = new Random();

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

{

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

{

dvumernii\_array[i, j] = GetRandomChar();

}

}

Dictionary<int, char[]> tempDict = new Dictionary<int, char[]>();

char[] tempArr = new char[columnSize];

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

{

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

{

tempArr[j] = dvumernii\_array[i, j];

}

if (!tempArr.All(char.IsLetter))

{

tempDict.Add(i, tempArr);

tempArr = new char[columnSize];

}

}

Console.WriteLine("------------------Массив-----------------\n");

for (int i = 0; i < dvumernii\_array.GetLength(0); i++)

{

for (int j = 0; j < dvumernii\_array.GetLength(1); j++)

{

Console.Write("{0}\t", dvumernii\_array[i, j]);

}

Console.WriteLine();

}

Console.WriteLine("\n-----------------------------------------");

if (tempDict.Count > 0)

{

Console.WriteLine("---------------Новый массив--------------\n");

if (dvumernii\_array.GetLength(0) == tempArr.Length)

{

Console.WriteLine(" Нет строк без цифр ");

}

else

{

foreach (var item in tempDict)

{

foreach (var ar in item.Value)

{

Console.Write("{0}\t", ar);

}

Console.WriteLine();

}

}

}

}

static void VvodDvumerniiArray(string[] args, ref int stringSize, ref int columnSize, ref char[,] dvumernii\_array)

{

Console.Write("\nВведите количество столбцов: ");

columnSize = Search\_for\_Bugs.ProverkaVvoda();

dvumernii\_array = new char[stringSize, columnSize];

Random rand = new Random();

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

{

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

{

dvumernii\_array[i, j] = Search\_for\_Bugs.Trying();

}

}

Dictionary<int, char[]> tempDict = new Dictionary<int, char[]>();

char[] tempArr = new char[columnSize];

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

{

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

{

tempArr[j] = dvumernii\_array[i, j];

}

if (!tempArr.All(char.IsLetter))

{

tempDict.Add(i, tempArr);

tempArr = new char[columnSize];

}

}

Console.WriteLine("------------------Массив-----------------\n");

for (int i = 0; i < dvumernii\_array.GetLength(0); i++)

{

for (int j = 0; j < dvumernii\_array.GetLength(1); j++)

{

Console.Write("{0}\t", dvumernii\_array[i, j]);

}

Console.WriteLine();

}

Console.WriteLine("\n-----------------------------------------");

if (tempDict.Count > 0)

{

Console.WriteLine("---------------Новый массив--------------\n");

if (dvumernii\_array.GetLength(0) == tempArr.Length)

{

Console.WriteLine(" Нет строк без цифр ");

}

else

{

foreach (var item in tempDict)

{

foreach (var ar in item.Value)

{

Console.Write("{0}\t", ar);

}

Console.WriteLine();

}

}

}

}

#endregion

//-----------------------------------------------------------------------------------------------------------

//---Второе задание------------------------------------------------------------------------------------------

#region

static void SecondTask()

{

Console.Clear();

Console.Write("Введите сроку: ");

string str = Console.ReadLine();

Console.WriteLine(ChangeStrings(str));

}

static string ChangeStrings(string str)

{

int pos = 0; // Позиция начала слова

string changedString = string.Empty; // Возвращаемая строка

string tempWord; // Массив символов

StringBuilder newString = new StringBuilder(str);

Regex regex = new Regex(@"\w+");

Match word = regex.Match(str);

while (word.Success)

{

pos = word.Index;

tempWord = new string(word.ToString().Reverse().ToArray());

newString.Remove(pos, tempWord.Length);

newString.Insert(pos, tempWord);

word = word.NextMatch();

}

changedString = string.Join(" ", newString.ToString().Split(' ', ',', ';', ';', '.', '!', '?').OrderBy(x => x.Length)).TrimStart();

return changedString;

}

#endregion

//-----------------------------------------------------------------------------------------------------------

static void Main(string[] args)

{

Console.Clear(); //Очищение консоли

int stringSize = 0,

columnSize = 0;

char[,] dvumernii\_array = null;

do

{

Console.WriteLine("\n-----------------------------------------");

Console.WriteLine("| Выберите следующее действие: |");

Console.WriteLine("| 1) 1 задание |");

Console.WriteLine("| 2) 2 задание |");

Console.WriteLine("| 0) Выход |");

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

Console.Write("Действие: ");

int elementmenu = Search\_for\_Bugs.ProverkaVvoda();

switch (elementmenu) //Выбор действия из меню

{

case 1:

FirstTask(args, ref stringSize, ref columnSize, ref dvumernii\_array);

break;

case 2:

SecondTask();

break;

case 0:

Environment.Exit(0); //Выход из консоли

break;

default:

Text\_Dialog.PrintErrorMenu();

continue;

}

} while (true);

}

}

}

# **5.Тесты:**

Формирование двумерного массива (с помощью ДСЧ):

|  |  |  |  |
| --- | --- | --- | --- |
| **Тест** | **Входные данные** | **Ожидаемый результат** | **Реальный результат** |
| 1 | 2 строки 5 столбцов  нижняя граница -100,  верхняя 100 | 10 чисел от -100 до 100 | -90 0 27 54 -67  23 43 95 4 4 |
| 2 | 0 строк | Сообщение об ошибке | Неверный ввод |
| 3 | -3 строк | Сообщение об ошибке | Неверный ввод |
| 4 | 1,34 строк | Сообщение об ошибке | Неверный ввод |
| 5 | Q строк | Сообщение об ошибке | Неверный ввод |
| 6 | 1 строка 1 столбец  нижняя граница -100,  верхняя 100 | 1 число | 10 |
| 7 | 1 строка 1 столбец  нижняя граница 1,  верхняя 1 | 1 | 1 |
| 8 | 1111111111111 | Сообщение об ошибке | Неверный ввод |
| 9 | 100 | Сообщение об ошибке | Неверный ввод |
| 10 | 5 строк 5 столбцов  нижняя граница 1,  верхняя 1 | 1 1 1 1 1 1  1 1 1 1 1 1  1 1 1 1 1 1  1 1 1 1 1 1  1 1 1 1 1 1 | 1 1 1 1 1 1  1 1 1 1 1 1  1 1 1 1 1 1  1 1 1 1 1 1  1 1 1 1 1 1 |
| 11 | 1 строка 1 столбец  нижняя граница 1,  верхняя -5 | Сообщение об ошибке | Нижняя граница не может быть больше верхней |

Формирование двумерного массива (с клавиатуры):

|  |  |  |  |
| --- | --- | --- | --- |
| **Тест** | **Входные данные** | **Ожидаемый результат** | **Реальный результат** |
| 1 | 30 0 5 28 -37  23 43 -95 4 4 | 30 0 5 28 -37  23 43 -95 4 4 | 30 0 5 28 -37  23 43 -95 4 4 |
| 2 | 0 строк | Сообщение об ошибке | Массив пуст |
| 3 | -3 строк | Сообщение об ошибке | Неверный ввод |
| 4 | 1,34 строк | Сообщение об ошибке | Неверный ввод |
| 5 | Q строк | Сообщение об ошибке | Неверный ввод |
| 6 | 3 строки 5 столбцов  1 1 1 1 1 1  1 1 1 1 1 1  1 1 1 1 1 1 | 1 1 1 1 1 1  1 1 1 1 1 1  1 1 1 1 1 1 | 1 1 1 1 1 1  1 1 1 1 1 1  1 1 1 1 1 1 |
| 7 | 1 строка 1 столбец  1 | 1 | 1 |
| 8 | 1111111111111 строк | Сообщение об ошибке | Неверный ввод |
| 9 | 1 строка 1 столбец  -100 | -100 | -100 |
| 10 | 1 строка 1 столбец  0 | 0 | 0 |

Ввод слова

|  |  |  |  |
| --- | --- | --- | --- |
| **Тест** | **Входные данные** | **Ожидаемый результат** | **Реальный результат** |
| 1 | программирование. Я устал. Хочу оливье, отстаньте от меня! | Я то учоХ янем латсу еьвило етьнатсто еинавориммаргорп | Я то учоХ янем латсу еьвило етьнатсто еинавориммаргорп |
| 2 | 12345566 все хватит | есв титавх 66554321 | есв титавх 66554321 |

Печать массива:

|  |  |  |  |
| --- | --- | --- | --- |
| **Тест** | **Входные данные** | **Ожидаемый результат** | **Реальный результат** |
| 1 | пустой массив | Сообщение об ошибке | Массив пустой |
| 2 | 1 1 1 1 1 1  1 1 1 1 1 1  1 1 1 1 1 1  1 1 1 1 1 1  1 1 1 1 1 1 | 1 1 1 1 1 1  1 1 1 1 1 1  1 1 1 1 1 1  1 1 1 1 1 1  1 1 1 1 1 1 | 1 1 1 1 1 1  1 1 1 1 1 1  1 1 1 1 1 1  1 1 1 1 1 1  1 1 1 1 1 1 |

Выбор в меню:

|  |  |  |  |
| --- | --- | --- | --- |
| **Тест** | **Входные данные** | **Ожидаемый результат** | **Реальный результат** |
| 1 | 1 | Udalenie(ref int[] massiv, ref int elementi) | Udalenie(ref int[] massiv, ref int elementi) |
| 2 | 2 | Dobavlenie(ref int[] massiv, ref int elementi) | Dobavlenie(ref int[] massiv, ref int elementi) |
| 3 | 0 | Выход из программы | Выход из программы |