|  |
| --- |
| Пермский филиал федерального государственного автономного образовательного учреждения высшего образования  «Национальный исследовательский университет  «Высшая школа экономики»  *Факультет социально-экономических и компьютерных наук* |
|  |
| Гуцол Степан Дмитриевич  **ЛАБОРАТОРНАЯ РАБОТА №6**  по направлению подготовки *38.03.05 Бизнес-информатика*  образовательная программа «Разработка информационных систем для бизнеса»   |  |  | | --- | --- | |  | Руководитель  Преподаватель кафедры ИТБ  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Найданов И.В. |   Пермь, 2022 |

**Оглавление**

[**Задача №1** 3](#_Toc122231678)

[**1. Постановка задачи** 3](#_Toc122231679)

[**2. Анализ** 3](#_Toc122231680)

[**3. Программа (листинг)** 4](#_Toc122231681)

[**4. Тестирование** 5](#_Toc122231682)

[**Задача №2** 6](#_Toc122231683)

[**1. Постановка задачи** 6](#_Toc122231684)

[**2. Анализ** 6](#_Toc122231685)

[**3. Программа (листинг)** 8](#_Toc122231686)

[**4. Тестирование** 9](#_Toc122231687)

[**Задача №3** 10](#_Toc122231688)

[**1. Постановка задачи** 10](#_Toc122231689)

[**2. Анализ** 10](#_Toc122231690)

[**3. Программа (листинг)** 11](#_Toc122231691)

[**4. Тестирование** 12](#_Toc122231692)

[**Задача №4** 13](#_Toc122231693)

[**1. Постановка задачи** 13](#_Toc122231694)

[**2. Анализ** 13](#_Toc122231695)

[**3. Программа (листинг)** 14](#_Toc122231696)

[**4. Тестирование** 15](#_Toc122231697)

[**Задача №5** 16](#_Toc122231698)

[**1. Постановка задачи** 16](#_Toc122231699)

[**2. Анализ** 16](#_Toc122231700)

[**3. Программа (листинг)** 17](#_Toc122231701)

[**4. Тестирование** 18](#_Toc122231702)

[**Задача №6** 19](#_Toc122231703)

[**1. Постановка задачи** 19](#_Toc122231704)

[**2. Анализ** 19](#_Toc122231705)

[**3. Программа (листинг)** 20](#_Toc122231706)

[**4. Тестирование** 21](#_Toc122231707)

[ПРИЛОЖЕНИЕ А **Программа** 22](#_Toc122231708)

**Задача №1**

**1. Постановка задачи**

Сформировать двумерный массив и заполнить его случайными символами типа char.

**2. Анализ**

Исходные данные:

* lenH – целое число, большее нуля (количество строчек),
* lenW – целое число, большее нуля(количество столбцов),

Выходные данные:

twoDimArr – двумерный массив, с lenH количеством строчек и lenW количеством столбцов с символами в диапазоне от ‘0’ до ‘z’.

Таблица 1 – Анализ классов исходных данных

| **Задача** | **Исходные данные** | **Класс входных данных** | **Класс выходных данных** |
| --- | --- | --- | --- |
| Сформировать двумерный массив и заполнить его случайными символами типа char. | lenH | lenH – целое число, большее нуля | создание двумерного массива(twoDimArr), если lenW тоже подходит |
| * lenH – строки (не числа) * lenH – отрицательные числа * lenH – нуль * lenH – вещественные числа | сообщение об ошибке |
| lenW | lenW – целое число, большее нуля | создание двумерного массива(twoDimArr), если lenH тоже подходит |
| * lenW – строки (не числа) * lenW – отрицательные числа * lenW – нуль * lenW – вещественные числа | сообщение об ошибке |

**3. Программа (листинг)**

На выбор пользователю предоставляется возможность воспользоваться разными способами создания двумерного массива, в этом случае пользователь выбирает создание двумерного массива с помощью датчика случайных символов в диапазоне от ‘0’ до ‘z’. Весь код программы представлен в приложении. См. Приложение А.

**4. Тестирование**

Таблица 2 – Тестирование методом чёрного ящика

| **№** | **Исходные данные** | **Ожидаемый результат** | **Полученный результат** |
| --- | --- | --- | --- |
| **1** | lenH = ‘0’  lenW = ‘’ | Сообщение об ошибке | Сообщение об ошибке |
| **2** | lenH = ‘-2’  lenW = ‘’ | Сообщение об ошибке | Сообщение об ошибке |
| **3** | lenH = ‘2.3’  lenW = ‘’ | Сообщение об ошибке | Сообщение об ошибке |
| **4** | lenH = ‘три’  lenW = ‘’ | Сообщение об ошибке | Сообщение об ошибке |
| **5** | lenH = ‘2’  lenW = ‘0’ | Сообщение об ошибке | Сообщение об ошибке |
| **6** | lenH = ‘2’  lenW = ‘-1’ | Сообщение об ошибке | Сообщение об ошибке |
| **7** | lenH = ‘2’  lenW = ‘три’ | Сообщение об ошибке | Сообщение об ошибке |
| **8** | lenH = ‘2’  lenW = ‘1.5’ | Сообщение об ошибке | Сообщение об ошибке |
| **9** | lenH = ‘2’  lenW = ‘3’ | двумерный массив(twoDimArr) со случайными символами в диапазоне от ‘0’ до ‘z’ | twoDimArr = ‘x i n  6 v >’ |

**Задача №2**

**1. Постановка задачи**

Сформировать двумерный массив и заполнить его элементами типа char с помощью ручного ввода.

**2. Анализ**

Исходные данные:

* lenH – целое число, большее нуля (количество строчек),
* lenW – целое число, большее нуля(количество столбцов),
* element – символ типа char,

Выходные данные:

twoDimArr – двумерный массив, с lenH количеством строчек и lenW количеством столбцов с элементами element.

Таблица 3 – Анализ классов исходных данных

| **Задача** | **Исходные данные** | **Класс входных данных** | **Класс выходных данных** |
| --- | --- | --- | --- |
| Сформировать двумерный массив и заполнить его элементами типа char с помощью ручного ввода | lenH | lenH – целое число, большее нуля | создание двумерного массива(twoDimArr), если lenW тоже подходит |
| * lenH – строки (не числа) * lenH – отрицательные числа * lenH – нуль   lenH – вещественные числа | сообщение об ошибке |
| lenW | lenW – целое число, большее нуля | создание двумерного массива(twoDimArr), если lenH тоже подходит |
| * lenW – строки (не числа) * lenW – отрицательные числа * lenW – нуль * lenW – вещественные числа | сообщение об ошибке |
| element | element – строка из одного символа | двумерный массив(twoDimArr) символов из заданного алфавита с lenH количеством строчек и lenW количеством столбцов |
| * element – строки из двух и более символов * element – пустой ввод | сообщение об ошибке ввода |

**3. Программа (листинг)**

На выбор пользователю предоставляется возможность воспользоваться разными способами создания двумерного массива, в этом случае пользователь выбирает создание двумерного массива с помощью ручного ввода. Весь код программы представлен в приложении. См. Приложение А.

**4. Тестирование**

Таблица 4 – Тестирование методом чёрного ящика

| **№** | **Исходные данные** | **Ожидаемый результат** | **Полученный результат** |
| --- | --- | --- | --- |
| **1** | lenH = ‘0’  lenW = ‘’  element = ‘’ | Сообщение об ошибке | Сообщение об ошибке |
| **2** | lenH = ‘-2’  lenW = ‘’  element = ‘’ | Сообщение об ошибке | Сообщение об ошибке |
| **3** | lenH = ‘2.3’  lenW = ‘’  element = ‘’ | Сообщение об ошибке | Сообщение об ошибке |
| **4** | lenH = ‘три’  lenW = ‘’  element = ‘’ | Сообщение об ошибке | Сообщение об ошибке |
| **5** | lenH = ‘2’  lenW = ‘0’  element = ‘’ | Сообщение об ошибке | Сообщение об ошибке |
| **6** | lenH = ‘2’  lenW = ‘-1’  element = ‘’ | Сообщение об ошибке | Сообщение об ошибке |
| **7** | lenH = ‘2’  lenW = ‘три’  element = ‘’ | Сообщение об ошибке | Сообщение об ошибке |
| **8** | lenH = ‘2’  lenW = ‘1.5’  element = ‘’ | Сообщение об ошибке | Сообщение об ошибке |
| **9** | lenH = ‘1’  lenW = ‘2’  element = ‘a ba’ | Сообщение об ошибке | Сообщение об ошибке |
| **10** | lenH = ‘1’  lenW = ‘2’  element = ‘a ‘ ‘ ’ | Сообщение об ошибке | Сообщение об ошибке |
| **11** | lenH = ‘2’  lenW = ‘3’  element = ‘a b c  а б в’ | twoDimArr = ‘a b c  а б в’ | twoDimArr = ‘a b c  а б в’ |

**Задача №3**

**1. Постановка задачи**

Распечатать двумерный динамический массив

**2. Анализ**

Исходные данные:

twoDimArr – двумерный массив, состоящий из символов типа char,

Выходные данные:

symbol – элемент массива.

Таблица 5 – Анализ классов исходных данных

| **Задача** | **Исходные данные** | **Класс входных данных** | **Класс выходных данных** |
| --- | --- | --- | --- |
| Распечатать двумерный динамический массив | twoDimArr | twoDimArr – массив с положительном количеством элементов | symbol – каждый элемент массива |

**3. Программа (листинг)**

Эта функция позволяет распечатать двумерный массив. Весь код программы представлен в приложении. См. Приложение А.

**4. Тестирование**

Таблица 6 – Тестирование методом чёрного ящика

| **№** | **Исходные данные** | **Ожидаемый результат** | **Полученный результат** |
| --- | --- | --- | --- |
| **1** | twoDimArr = ‘1  2’ | symbol = ‘1  2’ | symbol = ‘1  2’ |
| **2** | twoDimArr = ‘a b  c d’ | symbol = ‘a b  c d’ | symbol = ‘a b  c d’ |
| **3** | twoDimArr = ‘а б’ | symbol = ‘а б’ | symbol = ‘а б’ |

**Задача №4**

**1. Постановка задачи**

Удалить из массива все строки, в которых нет цифр.

**2. Анализ**

Исходные данные:

twoDimArr – двумерный массив, состоящий из символов типа char.

Выходные данные:

twoDimArr –изменённый двумерный массив, с которым были произведены операции.

Таблица 7 – Анализ классов исходных данных

| **Задача** | **Исходные данные** | **Класс входных данных** | **Класс выходных данных** |
| --- | --- | --- | --- |
| Удалить из массива все строки, в которых нет цифр. | twoDimArr | twoDimArr – двумерный массив с положительным количеством элементов | изменённый двумерный массив (twoDimArr) |

**3. Программа (листинг)**

Данная функция удаляет из двумерного массива все строчки, в которых нет цифр. Весь код программы представлен в приложении. См. Приложение А.

**4. Тестирование**

Таблица 8 – Тестирование методом чёрного ящика

| **№** | **Исходные данные** | **Ожидаемый результат** | **Полученный результат** |
| --- | --- | --- | --- |
| **1** | twoDimArr = ‘2 a  3 б’ | twoDimArr = ‘2 a  3 б’ | twoDimArr = ‘2 a  3 б’ |
| **2** | twoDimArr = ‘2 a  B б’ | twoDimArr = ‘2 a’ | twoDimArr = ‘2 a’ |
| **3** | twoDimArr = ‘p a  b б’ | twoDimArr = ‘’ | twoDimArr = ‘’ |

**Задача №5**

**1. Постановка задачи**

Распечатать изменённый двумерный динамический массив

**2. Анализ**

Исходные данные:

twoDimArr – двумерный массив, состоящий из символов типа char,

Выходные данные:

symbol – элемент массива.

Таблица 9 – Анализ классов исходных данных

| **Задача** | **Исходные данные** | **Класс входных данных** | **Класс выходных данных** |
| --- | --- | --- | --- |
| Распечатать двумерный динамический массив | twoDimArr | twoDimArr – массив с положительном количеством элементов | symbol – каждый элемент массива |
| twoDimArr – пустой массив | сообщение о пустом массиве |

**3. Программа (листинг)**

Эта функция позволяет распечатать изменённый двумерный массив. Весь код программы представлен в приложении. См. Приложение А.

**4. Тестирование**

Таблица 10 – Тестирование методом чёрного ящика

| **№** | **Исходные данные** | **Ожидаемый результат** | **Полученный результат** |
| --- | --- | --- | --- |
| **1** | twoDimArr = ‘1  2’ | symbol = ‘1  2’ | symbol = ‘1  2’ |
| **2** | twoDimArr = ‘a b  c d’ | symbol = ‘a b  c d’ | symbol = ‘a b  c d’ |
| **3** | twoDimArr = ‘а б’ | symbol = ‘а б’ | symbol = ‘а б’ |
| **4** | twoDimArr = ‘’ | сообщение о пустом массиве | сообщение о пустом массиве |

**Задача №6**

**1. Постановка задачи**

Ввести строку символов с помощью клавиатуры, состоящая из символов, включённые в заданный алфавит (латиница и кириллица, а также знаки препинания и пробел).

**2. Анализ**

Исходные данные:

str – строчка, состоящая из элементов заданного символа.

Таблица 11 – Анализ классов исходных данных

| **Задача** | **Исходные данные** | **Класс входных данных** | **Класс выходных данных** |
| --- | --- | --- | --- |
| Ввести строку символов с помощью клавиатуры, состоящая из символов, включённые в заданный алфавит (латиница и кириллица, а также знаки препинания и пробел). | str | str – строчка, состоящая из символов алфавита | str – строчка, с которой буду производиться действия |
| str – строчка, включающая в себя символы, не входящие в алфавит | сообщение об ошибке |

**3. Программа (листинг)**

Данная функция позволяет пользователю ввести строку с помощью клавиатуры в заданном алфавите. Весь код программы представлен в приложении. См. Приложение А.

**4. Тестирование**

Таблица 12 – Тестирование методом чёрного ящика

| **№** | **Исходные данные** | **Ожидаемый результат** | **Полученный результат** |
| --- | --- | --- | --- |
| **1** | str = ‘Привет, как дела?1’ | сообщение об ошибке | сообщение об ошибке |
| **2** | str = ‘Привет, как дела?’ | str = ‘Привет, как дела?’ | str = ‘Привет, как дела?’ |
| **3** | str = ‘Привет\ как дела?’ | сообщение об ошибке | сообщение об ошибке |

# ПРИЛОЖЕНИЕ А **Программа**

using System;

using System.Collections.Immutable;

using System.ComponentModel.Design;

using System.Diagnostics.Metrics;

using System.Linq.Expressions;

using System.Transactions;

namespace Labo6

{

internal class Program

{

static void Main(string[] args) // Основная функция

{

char[] numbers = new char[] { '0', '1', '2', '3', '4', '5', '6', '7', '8', '9' }; // Массив с цифрами

char[] alphabet = new char[] { 'q', 'w', 'e', 'r', 't', 'y', 'u', 'i', 'o', 'p', 'a', 's', 'd', 'f', 'g', 'h', 'j', 'k', 'l', 'z', 'x', 'c', 'v', 'b', 'n', 'm', 'й', 'ц', 'у', 'к', 'е', 'н', 'г', 'ш', 'щ', 'з', 'х', 'ъ', 'ф', 'ы', 'в', 'а', 'п', 'р', 'о', 'л', 'д', 'ж', 'э', 'я', 'ч', 'с', 'м', 'и', 'т', 'ь', 'б', 'ю', 'ё', 'Q', 'W', 'E', 'R', 'T', 'Y', 'U', 'I', 'O', 'P', 'A', 'S', 'D', 'F', 'G', 'H', 'J', 'K', 'L', 'Z', 'X', 'C', 'V', 'B', 'N', 'M', 'Й', 'Ц', 'У', 'К', 'Е', 'Н', 'Г', 'Ш', 'Щ', 'З', 'Х', 'Ъ', 'Ф', 'Ы', 'В', 'А', 'П', 'Р', 'О', 'Л', 'Д', 'Ж', 'Э', 'Я', 'Ч', 'С', 'М', 'И', 'Т', 'Ь', 'Б', 'Ю', 'Ё', ' ', '.', ',', '"', ':', ';', '!', '?', '-' }; // Массив разрешённых символов для 2-ой задачи

bool exit = false; //Логическая переменная выхода из программы

do

{

int enter = 0; // Переменная ввода данных

FirstMenu(ref enter); // Функция появления первого меню с последующим выбора пункта

CheckExitMain(enter, ref exit); // Функция проверка выхода из программы

switch (enter)

{

case 1: // Первая задача

bool exitSecond = false; // Переменная выхода из второго меню

do

{

SecondMenuOne(ref enter); // Функция , вызвывающая появления второго меню с последующим вводом

CheckExit(enter, ref exitSecond);

switch (enter)

{

case 1: //ДСС

bool exitThirdMenu = false;

Console.Clear();

Console.WriteLine("Введите целое число - количество строчек двумерного массива:");

int lenH = EnterNumberMoreZero();

Console.WriteLine("Введите целое число - количество столбцов двумерного массива:");

int lenW = EnterNumberMoreZero();

char[,] twoDimArr = new char[lenH, lenW];

CreateRandomArray(twoDimArr);

Console.WriteLine("Ваш массив:");

PrintArray(twoDimArr);

int counterOfStrings = 0;

ActionWithArr(ref twoDimArr, numbers, ref counterOfStrings);

Console.Write(@"

Ваш изменённый массив:");

if (counterOfStrings == 0)

{

Console.Write(@"

Пустой");

}

else

{

Console.WriteLine();

}

PrintArray(twoDimArr);

ExitThird(enter, ref exitThirdMenu);

break;

case 2: // Ручной ввод

exitThirdMenu = false;

Console.Clear();

Console.WriteLine("Введите целое число - количество строчек двумерного массива:");

lenH = EnterNumberMoreZero();

Console.WriteLine("Введите целое число - количество столбцов двумерного массива:");

lenW = EnterNumberMoreZero();

twoDimArr = new char[lenH, lenW];

CreateArray(twoDimArr);

Console.WriteLine("Ваш массив:");

PrintArray(twoDimArr);

counterOfStrings = 0;

ActionWithArr(ref twoDimArr, numbers, ref counterOfStrings);

Console.Write(@"

Ваш изменённый массив:");

if (counterOfStrings == 0)

{

Console.Write(@"

Пустой");

}

else

{

Console.WriteLine();

}

PrintArray(twoDimArr);

ExitThird(enter, ref exitThirdMenu);

break;

}

} while (!exitSecond);

break;

case 2: // Вторая задача

exitSecond = false;

Console.Clear();

do

{

CheckExit(enter, ref exitSecond);

string str = "";

SecondMenuTwo(ref str, alphabet);

Console.WriteLine(@$"

Ваша строка:

{str}

Ваша изменённая строка:

Ошибка! Кажется, у разработчика проблемы с распределением времени! :(

");

Console.WriteLine(@"

[0] - Вернуться назад");

do

{

enter = EnterNumber();

if (enter == 0)

{

exitSecond = true;

}

} while (enter != 0);

} while (!exitSecond);

break;

}

} while (!exit);

Console.Clear();

}

/// <summary>

/// Функция действий с двумерным массивом в первой задаче

/// </summary>

/// <param name="twoDimArr"> Двумерный массив</param>

/// <param name="numbers"> Массив со всеми возомжными цифрами для проверки</param>

/// <param name="counterOfStrings"> Счётчик строчек</param>

static void ActionWithArr(ref char[,] twoDimArr, char[] numbers, ref int counterOfStrings)

{

char[,] tmpArr = new char[twoDimArr.GetLength(0), twoDimArr.GetLength(1)];

int numberOfString = 0;

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

{

bool found = false;

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

{

int counter = 0;

Seeker(ref counter, ref found, twoDimArr, i, j, numbers);

}

if (found)

{

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

{

tmpArr[numberOfString, j] = twoDimArr[i, j];

}

numberOfString++;

counterOfStrings++;

}

}

twoDimArr = tmpArr;

}

/// <summary>

///

/// </summary>

/// <param name="counter"> Счётчик количества элементов</param>

/// <param name="found">Логическая функция нахождения или ненахождения цифр</param>

/// <param name="twoDimArr"> Двумерный массив</param>

/// <param name="i"> Индекс строчки в массиве</param>

/// <param name="j"> Индекс столбца в массиве</param>

/// <param name="numbers"> Массив с цифрами для проверки</param>

static void Seeker(ref int counter, ref bool found, char[,] twoDimArr, int i, int j, char[] numbers)

{

foreach (char element in numbers)

{

if (twoDimArr[i, j] == element)

{

counter++;

}

}

if (counter > 0)

{

found = true;

}

}

/// <summary>

/// Функция подсчёта элементов в строке, находящихся в алфавите(alphabet)

/// </summary>

/// <param name="str">Вводимая строка</param>

/// <param name="counter">Переменная счётчика</param>

/// <param name="alphabet"> Массив, в котором находится весь допустимый алфавит</param>

static void Seeker(ref string str, ref int counter, char[] alphabet)

{

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

{

foreach (char element in alphabet)

{

if (str[i] == element)

{

counter++;

}

}

}

}

/// <summary>

/// Функция ввода и проверки на выход из третьего меню, чтобы вернуться в меню создания массива

/// </summary>

/// <param name="enter">Ввод ожидаемого нуля</param>

/// <param name="exitThirdMenu"> Логическая переменная выхода из меню</param>

static void ExitThird(int enter, ref bool exitThirdMenu)

{

Console.WriteLine("[0] - Вернуться назад");

do

{

Return(enter, ref exitThirdMenu);

} while (!exitThirdMenu);

}

/// <summary>

/// Функция возвращения в предыдущее меню

/// </summary>

/// <param name="enter">переменная ввода</param>

/// <param name="exitThirdMenu"> Логическая переменная, отвечающая за выход из меню</param>

static void Return(int enter, ref bool exitThirdMenu)

{

enter = EnterNumber();

if (enter == 0)

{

exitThirdMenu = true;

}

}

/// <summary>

/// Функция проверки ввода целого числа

/// </summary>

/// <returns></returns>

static int EnterNumber() // Ввод целого числа

{

bool ok;

int Num;

do

{

ok = int.TryParse(Console.ReadLine(), out Num);

} while (!ok);

return Num;

}

/// <summary>

/// функция заполнения двумерного массива элементами char с помощью ручного ввода

/// </summary>

/// <param name="twoDimArr">Двумерный массив</param>

static void CreateArray(char[,] twoDimArr) //Создание двумерного массива со своими элементами

{

Random Rand = new Random();

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

{

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

{

twoDimArr[i, j] = EnterCharArr();

}

}

}

/// <summary>

/// Функция ввода элемента массива с проверкой является ли он типом char

/// </summary>

/// <returns>Возвращает введённый с клавиатуры элемент, если он подходит</returns>

static char EnterCharArr() // Ввод элемента массива

{

bool ok;

char element;

do

{

Console.WriteLine("Введите пожалуйста элемент массива, он должен являться одним символом:");

ok = char.TryParse(Console.ReadLine(), out element);

if (!ok)

{

Console.ForegroundColor = ConsoleColor.Red;

Console.WriteLine(@"Вы ввели некорректный элемент массива!");

}

Console.ForegroundColor = ConsoleColor.White;

} while (!ok);

return element;

}

/// <summary>

/// Функция заполнения двумерного массива случайными элементами типа char

/// </summary>

/// <param name="twoDimArr">Двумерный массив</param>

static void CreateRandomArray(char[,] twoDimArr) //Создание случайного двумерного массива

{

Random Rand = new Random();

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

{

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

{

twoDimArr[i, j] = Convert.ToChar(Rand.Next('0', 'z' + 1));

}

}

}

/// <summary>

/// Функция ввода целого числа большего нуля, необходима для ввода количества строк и столбцов массива

/// </summary>

/// <returns>Возвращает целочисленное значение для количества строк или столбцов</returns>

static int EnterNumberMoreZero() // Ввод целого большего нуля числа

{

bool ok;

int Num;

do

{

Console.WriteLine("Введите пожалуйста целое число:");

ok = ((int.TryParse(Console.ReadLine(), out Num)) && (Num > 0));

if (!ok)

{

Console.ForegroundColor = ConsoleColor.Red;

Console.WriteLine("Вы ввели некорректное число!");

}

Console.ForegroundColor = ConsoleColor.White;

} while (!ok);

return Num;

}

/// <summary>

/// Функция печати массива

/// </summary>

/// <param name="twoDimArr">Двумерный массив</param>

static void PrintArray(char[,] twoDimArr) // Печать двумерного массива

{

int Counter = 0;

foreach (char symbol in twoDimArr)

{

Counter++;

if (Counter % twoDimArr.GetLength(1) == 0)

{

Console.WriteLine(symbol);

}

else

{

Console.Write(symbol + " ");

}

}

}

/// <summary>

/// Функция проверки выхода из меню

/// </summary>

/// <param name="enter">Перменная ,ожидающая ввода нуля</param>

/// <param name="exitSecond">Логическая функция, отвечающая за выход из меню</param>

static void CheckExit(int enter, ref bool exitSecond)

{

if (enter == 0)

{

exitSecond = true;

}

}

/// <summary>

/// Функция проверки выхода из программы

/// </summary>

/// <param name="enter">Переменная для ввода с клавиатуры, ожидающая нуля</param>

/// <param name="exit">Логическая переменная, отвечающая за выход из программы</param>

static void CheckExitMain(int enter, ref bool exit)

{

if (enter == 0)

{

exit = true;

}

}

/// <summary>

/// Фукнция печати приветственного сообщения с последующим вводом

/// </summary>

/// <param name="enter">Перменная ввода с клавиатуры для выбора пункта меню</param>

static void FirstMenu(ref int enter)

{

Console.Clear();

Console.WriteLine(@"Выберите задачу:

[1] - Создать двумерный массив и удалить из массива все строки, в которых нет цифр.

[2] - Ввести строку из символов ,и перевернуть все слова в предложении ,и отсортировать слова по убыванию в лексикографическом порядке.

[0] - Выйти из программы.

");

enter = EnterMenu();

}

/// <summary>

/// Функция печати приветственного сообщения для второго меню с последующим вводом

/// </summary>

/// <param name="enter">Переменная для ввода с клавиатуры следующего пункта меню</param>

static void SecondMenuOne(ref int enter)

{

Console.Clear();

Console.WriteLine(@"Каким образом создать массив?

[1] - С помощью Датчика Случайных Символов

[2] - Ручной ввод

[0] - Назад

");

enter = EnterMenu();

}

/// <summary>

/// Функция печати второго меню второй задачи с последующим вводом строки

/// </summary>

/// <param name="str">Переменная , хранящая строку</param>

/// <param name="alphabet">Массив с символами, которые допускается вводить вручную</param>

static void SecondMenuTwo(ref string str, char[] alphabet)

{

Console.WriteLine("Введите вашу строку, она должна содержать только символы русского и латинского алфавита, а также знаки препинания:");

EnterString(ref str, alphabet);

}

/// <summary>

/// Функция ввода строки

/// </summary>

/// <param name="str">Переменная, которая хранит строку</param>

/// <param name="alphabet">Массив-алфавит, в котором указаны допустимые символы</param>

static void EnterString(ref string str, char[] alphabet)

{

bool ok = false;

do

{

str = Console.ReadLine();

int counter = 0;

Seeker(ref str, ref counter, alphabet);

if ((str.Length == counter) && (str.Length > 0))

{

ok = true;

}

else

{

Console.ForegroundColor = ConsoleColor.Red;

Console.WriteLine("Вы ввели некорректную строку, строка может содержать только символы латиницы и кирилилцы, а также знаки препинания!");

}

Console.ForegroundColor = ConsoleColor.White;

} while (!ok);

}

/// <summary>

/// Функция ввода с клавиатуры для следующего выбора меню

/// </summary>

/// <returns>Возвращает целое число - пункта меню</returns>

static int EnterMenu() // Ввод целого числа

{

bool ok;

int Num;

do

{

Console.WriteLine("Выберите пункт меню:");

ok = int.TryParse(Console.ReadLine(), out Num);

if (!ok)

{

Console.ForegroundColor = ConsoleColor.Red;

Console.WriteLine("Вы ввели некорректные данные!");

}

Console.ForegroundColor = ConsoleColor.White;

} while (!ok);

return Num;

}

/\*static void ActionWithString(ref string str)

{

int counterOfSentences = 0;

foreach(char element in str)

{

if ((element == '.') || (element == '?') || (element == '!'))

{

counterOfSentences++;

}

}

char[] charArray = str.ToCharArray();

Console.WriteLine(charArray);

string newString = "";

do

{

char[] tmpArray = new char[charArray.Length];

int counterOfElements = 1;

bool found = false;

do

{

tmpArray[counterOfElements-1]=charArray[counterOfElements-1];

counterOfElements++;

found = tmpArray[counterOfElements - 1] == '!';

} while (!found);

Array.Reverse(tmpArray, 0, counterOfElements - 1);

newString = newString + tmpArray;

counterOfSentences--;

} while (counterOfSentences > 0);

str = newString;

}\*/

}

}