|  |
| --- |
| Пермский филиал федерального государственного автономного образовательного учреждения высшего образования  «Национальный исследовательский университет  «Высшая школа экономики»  *Факультет экономики, менеджмента и бизнес-информатики* |
|  |
| Полежаев Яков Вячеславович  **ЛАБОРАТОРНАЯ РАБОТА №5**  по направлению подготовки *38.03.05 Бизнес-информатика*  образовательная программа «Бизнес-информатика»   |  |  | | --- | --- | |  | Руководитель  Преподаватель кафедры ИТБ  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Марквирер В.Д. |   Пермь, 2022 |

**Задача №1**

**1. Постановка задачи**

1. Сформировать динамический одномерный массив, заполнить его случайными числами и вывести на печать.
2. Выполнить указанное в варианте задание и вывести полученный массив на печать.
3. Сформировать динамический двумерный массив, заполнить его случайными числами и вывести на печать.
4. Выполнить указанное в варианте задание и вывести полученный массив на печать.
5. Сформировать динамический рваный массив, заполнить его случайными числами и вывести на печать.
6. Выполнить указанное в варианте задание и вывести полученный массив на печать.
7. При реализации функций необходимо продемонстрировать использование параметров разных типов и различные способы организации функций (параметры по умолчанию, перегрузку функций, и .т.д.)

|  |  |  |  |
| --- | --- | --- | --- |
| № варианта | Одномерный массив | Двумерный массив | Рваный массив |
| 1 | Удалить первый четный элемент | Добавить строку с заданным номером | Удалить самую длинную строку |

**2. Анализ**

Исходные данные:

* NumOfStrings – натуральное положительное число(int).
* NumsOfColons– натуральное положительное число(int).
* Nums[указанный элемент] - натуральное число (int)

Выходные данные:

* nums – массив (int)

***Табл. 1. Анализ классов исходных данных.***

|  |  |  |
| --- | --- | --- |
| **Исходные данные** | **Класс входных данных** | **Выходные данные** |
| NumOfStrings | Не целое или натуральное число | Сообщение об ошибке ввода, повторный ввод |
| NumOfStrings | Целое натуральное положительное число | N = {n} |
| NumOfStrings | Равен 0 | Массив пустой |
| NumOfColons | Целое натуральное положительное число | N = {n} |
| NumOfColons | Равен 0 | Массив пустой |
| NumOfColons | Не целое или натуральное число | Сообщение об ошибке ввода, повторный ввод |
| nums[i]/ nums[I,j]/ nums[i][j] | Целые числа | Вывод массива nums |
| Все нулевые | Вывод пустой строки |
| Не число или не целое число | Сообщение об ошибке ввода, повторный ввод |

**2.Краткое описание**

* Код делится на множество подпрограмм, среди них есть как простейшие(ввод числа с проверкой, перебор массивов и т.д.), так и сложные(создание любого массива, добавление строки, удаление строки)
* Для функции создания массива использовалась перегрузка.
* В Мейне располагается код для консоли, где уже используются все основные методы.
* Само управление консолью проходит через switch при помощи ввода цифр от 1 до 4 и ответов “да” “нет”.
* Основной текст (желтого цвета)\ номер элемента(зеленого)\данные на ввод и вывод(белого)
* При ошибке будет звуковой сигнал и красный текст об ошибке
* При завершении программы с вами прощаются котики

**3. Программа (листинг)**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Media;

namespace лаб\_5

{

internal class Program

{

public static void YellowColor(string text)

{

Console.ForegroundColor = ConsoleColor.DarkYellow;

Console.WriteLine(text);

Console.ForegroundColor = ConsoleColor.White;

}

public static int NumInput(out int a)//ввод числа типа инт

{

a = 0;

bool flag;

do

{

flag = int.TryParse(Console.ReadLine(), out a);

if (flag == false)

{

Console.ForegroundColor = ConsoleColor.Red;

SystemSounds.Beep.Play();

Console.WriteLine("!!!Введено не число или не целое число или отрицательное!!!");

Console.ForegroundColor = ConsoleColor.White;

flag = false;

}

} while (flag != true);

return a;

}

public static void ManualArrFilling(int[] mas)//ручное наполнение одномерного массива

{

int num;

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

{

mas[i] = NumInput(out num);

}

}

public static void ManualArrFilling(int[,] mas, int strings, int colons)//ручное наполнение двумерного массива

{

int num;

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

{

Console.ForegroundColor = ConsoleColor.Green;

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

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

{

Console.ForegroundColor = ConsoleColor.Green;

Console.WriteLine($"{j + 1}элемент");

Console.ForegroundColor = ConsoleColor.White;

mas[i, j] = NumInput(out num);

}

}

}

public static void ManualArrFilling(int[] mas, int strings)//ручное наполнение рваного массива

{

int num;

Console.WriteLine($"!{strings}строка!");

for (int j = 0; j < mas.Length; j++)

{

Console.ForegroundColor = ConsoleColor.Green;

Console.WriteLine($"{j + 1}элемент");

Console.ForegroundColor = ConsoleColor.White;

mas[j] = NumInput(out num);

}

}

public static void AutoArrFilling(int[] mas)// автоматическое наполнение массива

{

int a, b;

Random rnd = new Random();

YellowColor("хотите устновить диапозон? напишите да или нет");

do

{

string quest = Console.ReadLine();

if (quest == "да")

{

do

{

YellowColor("Вводите числа. Ваш диапозон:");

Console.ForegroundColor = ConsoleColor.DarkYellow;

Console.Write("от ");

Console.ForegroundColor= ConsoleColor.White;

NumInput(out a);

Console.ForegroundColor = ConsoleColor.DarkYellow;

Console.Write("до ");

Console.ForegroundColor = ConsoleColor.White;

NumInput(out b);

if (a > b)

{

Console.ForegroundColor = ConsoleColor.Red;

SystemSounds.Beep.Play();

Console.WriteLine("!!!Левая граница не может быть больше правой!!!");

Console.ForegroundColor = ConsoleColor.White;

}

} while (a > b);

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

{

mas[i] = rnd.Next(a, b);

}

break;

}

else if (quest == "нет")

{

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

{

mas[i] = rnd.Next();

}

break;

}

else

{

Console.ForegroundColor = ConsoleColor.Red;

SystemSounds.Beep.Play();

Console.WriteLine("!!!ошибка ввода!!!");

Console.ForegroundColor = ConsoleColor.White;

}

} while (true);

}

public static void AutoArrFilling(int[,] mas, int strings, int colons)// автоматическое наполнение двумерного массива

{

int a, b;

Random rnd = new Random();

YellowColor("хотите устновить диапозон? напишите да или нет");

do

{

string quest = Console.ReadLine();

if (quest == "да")

{

do

{

YellowColor("Вводите числа. Ваш диапозон:");

Console.ForegroundColor = ConsoleColor.DarkYellow;

Console.Write("от ");

Console.ForegroundColor = ConsoleColor.White;

NumInput(out a);

Console.ForegroundColor = ConsoleColor.DarkYellow;

Console.Write("до ");

Console.ForegroundColor = ConsoleColor.White;

NumInput(out b);

if (a > b)

{

Console.ForegroundColor = ConsoleColor.Red;

SystemSounds.Beep.Play();

Console.WriteLine("!!!Левая граница не может быть больше правой!!!");

Console.ForegroundColor = ConsoleColor.White;

}

} while (a > b);

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

{

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

{

mas[i, j] = rnd.Next(a, b);

}

}

break;

}

else if (quest == "нет")

{

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

{

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

{

mas[i, j] = rnd.Next();

}

}

break;

}

else

{

Console.ForegroundColor = ConsoleColor.Red;

SystemSounds.Beep.Play();

Console.WriteLine("!!!ошибка ввода!!!");

Console.ForegroundColor = ConsoleColor.White;

}

} while (true);

}

public static void AutoArrFilling(int[] mas, int strings)// автоматическое наполнение рваного массива

{

int a, b;

Random rnd = new Random();

YellowColor("хотите устновить диапозон? напишите да или нет");

do

{

string quest = Console.ReadLine();

if (quest == "да")

{

do

{

YellowColor("Вводите числа. Ваш диапозон:");

Console.ForegroundColor = ConsoleColor.DarkYellow;

Console.Write("от ");

Console.ForegroundColor = ConsoleColor.White;

NumInput(out a);

Console.ForegroundColor = ConsoleColor.DarkYellow;

Console.Write("до ");

Console.ForegroundColor = ConsoleColor.White;

NumInput(out b);

if (a > b)

{

Console.ForegroundColor = ConsoleColor.Red;

SystemSounds.Beep.Play();

Console.WriteLine("!!!Левая граница не может быть больше правой!!!");

Console.ForegroundColor = ConsoleColor.White;

}

} while (a > b);

for (int j = 0; j < mas.Length; j++)

{

mas[j] = rnd.Next(a, b);

}

break;

}

else if (quest == "нет")

{

for (int j = 0; j < mas.Length; j++)

{

mas[j] = rnd.Next();

}

break;

}

else

{

Console.ForegroundColor = ConsoleColor.Red;

SystemSounds.Beep.Play();

Console.WriteLine("!!!ошибка ввода!!!");

Console.ForegroundColor = ConsoleColor.White;

}

} while (true);

}

public static void ArrIteration( int[] mas)//перебор одномерного массива

{

if(mas.Length == 0)

{

Console.WriteLine("UsualArr = пустой");

}

else

{

Console.Write("UsualArr = ");

foreach (int i in mas)

{

Console.Write($"{i}, ");

}

Console.WriteLine();

}

}

public static void ArrIteration(int[,] mas, int strings, int colons)//перебор двумерного массива

{

if (strings == 0|| colons == 0)

{

Console.WriteLine("DoubleArr = пустой");

}

else

{

Console.WriteLine("DoubleArr = ");

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

{

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

{

if (mas[i, j] >= 10 || mas[i, j] < 0)

{

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

}

else

{

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

}

}

Console.WriteLine();

}

}

}

public static void ArrIteration( int[][] mas, int strings)//перебор рваного массива

{

if (mas.Length == 0)

{

Console.WriteLine("Nums = пустой");

}

else

{

Console.WriteLine("DoubleArr = ");

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

{

if(mas[i].Length == 0)

{

Console.WriteLine("строка пустая");

}

else

{

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

{

if (mas[i][j] >= 10 || mas[i][j] <= 0)

{

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

}

else

{

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

}

}

Console.WriteLine();

}

}

}

}

public static void CreateArr(out int[] mas)//создание одномерного массива

{

int elements = 0;

YellowColor("Введите колличество элементов");

do

{

NumInput(out elements);

if (elements < 0)

{

Console.ForegroundColor = ConsoleColor.Red;

SystemSounds.Beep.Play();

Console.WriteLine("!!!вы ввели отрицательное число!!!");

Console.ForegroundColor = ConsoleColor.White;

}

} while (elements < 0);

mas = new int[elements];

if (elements != 0)

{

YellowColor("хотите ввести элементы вручную?");

do

{

string quest = Console.ReadLine();

if (quest == "да")

{

YellowColor("вводите числа по порядку");

ManualArrFilling(mas);

break;

}

else if (quest == "нет")

{

AutoArrFilling(mas);

break;

}

else

{

Console.ForegroundColor = ConsoleColor.Red;

SystemSounds.Beep.Play();

Console.WriteLine("!!!ошибка ввода!!!");

Console.ForegroundColor = ConsoleColor.White;

}

} while (true);

}

}

public static void CreateArr(out int[,] mas, out int strings, out int colons)//создание двумерного массива

{

YellowColor("укажите колличество строк");

do

{

NumInput(out strings);

if (strings < 0)

{

Console.ForegroundColor = ConsoleColor.Red;

SystemSounds.Beep.Play();

Console.WriteLine("!!!вы ввели отрицательное число!!!");

Console.ForegroundColor = ConsoleColor.White;

}

} while (strings < 0);

YellowColor("укажите колличество столбцов");

do

{

NumInput(out colons);

if (colons < 0)

{

Console.ForegroundColor = ConsoleColor.Red;

SystemSounds.Beep.Play();

Console.WriteLine("!!!вы ввели отрицательное число!!!");

Console.ForegroundColor = ConsoleColor.White;

}

} while (colons < 0);

mas = new int[strings, colons];

if (strings != 0 && colons != 0)

{

YellowColor("хотите ввести элементы вручную?");

do

{

string quest = Console.ReadLine();

if (quest == "да")

{

ManualArrFilling(mas, strings, colons);

break;

}

else if (quest == "нет")

{

AutoArrFilling(mas, strings, colons);

break;

}

else

{

Console.ForegroundColor = ConsoleColor.Red;

SystemSounds.Beep.Play();

Console.WriteLine("!!!ошибка ввода!!!");

Console.ForegroundColor = ConsoleColor.White;

}

} while (true);

}

}

public static void CreateArr(out int[][] mas, out int strings)//создание рваного массива

{

YellowColor("укажите колличество строк");

do

{

NumInput(out strings);

if (strings < 0)

{

Console.ForegroundColor = ConsoleColor.Red;

SystemSounds.Beep.Play();

Console.WriteLine("!!!вы ввели отрицательное число!!!");

Console.ForegroundColor = ConsoleColor.White;

}

} while (strings < 0);

mas = new int[strings][];

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

{

YellowColor("Введите кол-во элементов");

Console.ForegroundColor = ConsoleColor.Green;

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

YellowColor("строки");

int columns = 0;

do

{

NumInput(out columns);

if (columns < 0)

{

Console.ForegroundColor = ConsoleColor.Red;

SystemSounds.Beep.Play();

Console.WriteLine("!!!вы ввели отрицательное число!!!");

Console.ForegroundColor = ConsoleColor.White;

}

} while (columns < 0);

mas[i] = new int[columns];

if (columns != 0)

{

YellowColor("хотите ввести элементы вручную?");

do

{

string quest = Console.ReadLine();

if (quest == "да")

{

ManualArrFilling(mas[i], i);

break;

}

else if (quest == "нет")

{

AutoArrFilling(mas[i], strings);

break;

}

else

{

Console.ForegroundColor = ConsoleColor.Red;

SystemSounds.Beep.Play();

Console.WriteLine("!!!ошибка ввода!!!");

Console.ForegroundColor = ConsoleColor.White;

}

} while (true);

}

}

}

public static void FirstEvenElement(ref int[] nums)//удалить первый четный

{

if (nums.Length == 0)

{

Console.ForegroundColor = ConsoleColor.Red;

Console.WriteLine("невозможно удалить четный элемент в пустом массиве");

Console.ForegroundColor = ConsoleColor.White;

}

else

{

int[] ArrNoItem = new int[nums.Length - 1];

int element = -1;

for (int item = 0; item < nums.Length; item++)

{

if (nums[item] % 2 == 0)

{

element = item;

Console.ForegroundColor = ConsoleColor.DarkYellow;

Console.Write("удалим первый четный элемент ");

Console.ForegroundColor = ConsoleColor.White;

Console.Write($"{nums[item]}");

Console.ForegroundColor = ConsoleColor.DarkYellow;

YellowColor(" на ");

Console.ForegroundColor = ConsoleColor.Green;

Console.Write($"{item + 1}");

YellowColor(" позиции ");

break;

}

}

if (element == -1)

{

Console.ForegroundColor = ConsoleColor.Red;

Console.WriteLine("в массиве нет четных элементов");

Console.ForegroundColor = ConsoleColor.White;

}

else

{

for (int item = 0, item2 = 0; item < ArrNoItem.Length && item2 < nums.Length;)

{

if (item2 != element)

{

ArrNoItem[item] = nums[item2];

item++;

item2++;

}

else

{

item2++;

}

}

nums = ArrNoItem;

}

ArrIteration(nums);

}

}

public static void AddString(ref int[,] mas2,ref int strings, ref int colons)//добавление строки к двумерному массиву

{

int addstring;

YellowColor("добавим к массиву одну строку. Под каким номером?");

do

{

NumInput(out addstring);

if (addstring < 0)

{

Console.ForegroundColor = ConsoleColor.Red;

SystemSounds.Beep.Play();

Console.WriteLine("!!!вы ввели отрицательное число!!!");

Console.ForegroundColor = ConsoleColor.White;

}

else if(addstring > strings+1)

{

Console.ForegroundColor = ConsoleColor.Red;

SystemSounds.Beep.Play();

Console.WriteLine("!!!вы ввели склишком большое число!!!");

Console.ForegroundColor = ConsoleColor.White;

}

else if (addstring == 0)

{

Console.ForegroundColor = ConsoleColor.Red;

SystemSounds.Beep.Play();

Console.WriteLine("!!!порядок строк начинается от 1!!!");

Console.ForegroundColor = ConsoleColor.White;

}

} while (addstring < 0 || addstring > strings + 1 || addstring ==0);

addstring--;

strings = mas2.GetLength(0);

colons = mas2.GetLength(1);

if (colons == 0)

{

YellowColor("Массив был пустой. Введите количество элементов строки");

NumInput(out colons);

}

int[,] ArrBigger = new int[strings + 1, colons];

strings++;

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

{

if (i == addstring)

{

YellowColor("хотите ввести элементы вручную?");

do

{

string quest = Console.ReadLine();

if (quest == "да")

{

YellowColor("вводите числа по порядку");

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

{

NumInput(out ArrBigger[i, j]);

}

break;

}

else if (quest == "нет")

{

int a, b;

Random rnd = new Random();

YellowColor("хотите устновить диапозон? напишите да или нет");

do

{

quest = Console.ReadLine();

if (quest == "да")

{

do

{

YellowColor("Вводите числа. Ваш диапозон:");

Console.ForegroundColor = ConsoleColor.DarkYellow;

Console.Write("от ");

Console.ForegroundColor = ConsoleColor.White;

NumInput(out a);

Console.ForegroundColor = ConsoleColor.DarkYellow;

Console.Write("до ");

Console.ForegroundColor = ConsoleColor.White;

NumInput(out b);

if (a > b)

{

Console.ForegroundColor = ConsoleColor.Red;

SystemSounds.Beep.Play();

Console.WriteLine("!!!Левая граница не может быть больше правой!!!");

Console.ForegroundColor = ConsoleColor.White;

}

} while (a > b);

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

{

ArrBigger[i, j] = rnd.Next(a, b);

}

break;

}

else if (quest == "нет")

{

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

{

ArrBigger[i, j] = rnd.Next();

}

break;

}

else

{

Console.ForegroundColor = ConsoleColor.Red;

SystemSounds.Beep.Play();

Console.WriteLine("!!!ошибка ввода!!!");

Console.ForegroundColor = ConsoleColor.White;

}

} while (true);

break;

}

else

{

Console.ForegroundColor = ConsoleColor.Red;

SystemSounds.Beep.Play();

Console.WriteLine("!!!ошибка ввода!!!");

Console.ForegroundColor = ConsoleColor.White;

}

} while (true);

i++;

}

if(i < strings && strings - 1 !=0)

{

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

{

ArrBigger[i, j] = mas2[n, j];

}

}

}

mas2 = ArrBigger;

}

public static void DeleteTheLonsgest(ref int [][] nums3,ref int strings)//удаление самой длинной строки

{

if (nums3.Length != 0)

{

int longstring = 0;

int DelString = 0;

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

{

if (nums3[i].Length > longstring)

{

longstring = nums3[i].Length;

DelString = i;

}

}

int[][] ArrWithoutString = new int[strings - 1][];

for (int i = 0, b=0;i < nums3.Length-1 && b < nums3.Length;)

{

if (b != DelString)

{

ArrWithoutString[i] = new int[nums3[b].Length];

for (int j = 0;j < nums3[b].Length;j++)

{

ArrWithoutString[i][j] = nums3[b][j];

}

i++;

b++;

}

else

{

b++;

}

}

Console.WriteLine($" удалим {DelString+1} строку");

foreach(int num in nums3[DelString])

{ Console.Write($"{num} "); }

nums3 = ArrWithoutString;

strings--;

}

else

{

Console.ForegroundColor = ConsoleColor.Red;

Console.WriteLine("В пустом массиве нечего удалять");

Console.ForegroundColor = ConsoleColor.White;

}

}

static void Main(string[] args)

{

string MainMenu = "1. работа с одномерным массивом" +

"\n2. работа с двумерным массивом" +

"\n3. работа с рваным массивом" +

"\n4. выход";

string UsualMassiveMenu = "1. создать новый массив" +

"\n2. перебор массива" +

"\n3. удалить первый четный элемент" +

"\n4. назад";

string DoubleMassiveMenu = "1. создать новый массив" +

"\n2. перебор массива" +

"\n3. добавить строку" +

"\n4. назад";

string SplitMassiveMenu = "1. создать новый массив" +

"\n2. перебор массива" +

"\n3. удалить самую длинную строку" +

"\n4. назад";

int num;

do

{

YellowColor("нажмите цифру от 1 до 4");

YellowColor(MainMenu);

NumInput(out num);

switch (num)

{

case 1://работа с одномерным массивом

{

Console.Clear();

int[] nums = {};

do

{

ArrIteration(nums);

YellowColor("нажмите цифру от 1 до 4");

YellowColor(UsualMassiveMenu);

NumInput(out num);

switch (num)

{

case 1:

{

Console.Clear();

CreateArr(out nums);

Console.ReadKey();

Console.Clear();

break;

}

case 2:

{

Console.Clear();

ArrIteration(nums);

Console.ReadKey();

Console.Clear();

break;

}

case 3:

{

Console.Clear();

FirstEvenElement(ref nums);

Console.ReadKey();

Console.Clear();

break;

}

case 4:

{

Console.Clear();

break;

}

default:

{

Console.Clear();

Console.ForegroundColor = ConsoleColor.Red;

SystemSounds.Beep.Play();

Console.WriteLine("!!!ошибка ввода!!!");

Console.ForegroundColor = ConsoleColor.White;

Console.ReadKey();

Console.Clear();

break;

}

}

} while (num != 4);

num = 0;

break;

}

case 2://работа с двумерным массивом

{

Console.Clear();

int[,] nums = {};

int NumOfStrings = 0;

int NumOfColons = 0;

do

{

ArrIteration(nums, NumOfStrings, NumOfColons);

YellowColor("нажмите цифру от 1 до 4");

YellowColor(DoubleMassiveMenu);

NumInput(out num);

switch (num)

{

case 1:

{

Console.Clear();

CreateArr(out nums, out NumOfStrings, out NumOfColons);

Console.ReadKey();

Console.Clear();

break;

}

case 2:

{

Console.Clear();

ArrIteration(nums, NumOfStrings, NumOfColons);

Console.ReadKey();

Console.Clear();

break;

}

case 3:

{

Console.Clear();

AddString(ref nums, ref NumOfStrings, ref NumOfColons);

Console.ReadKey();

Console.Clear();

break;

}

case 4:

{

Console.Clear();

break;

}

default:

{

Console.Clear();

Console.ForegroundColor = ConsoleColor.Red;

SystemSounds.Beep.Play();

Console.WriteLine("!!!ошибка ввода!!!");

Console.ForegroundColor = ConsoleColor.White;

Console.ReadKey();

Console.Clear();

break;

}

}

}while (num != 4);

num = 0;

break;

}

case 3://работа с рваным массивом

{

Console.Clear();

int[][] nums = { };

int NumOfStrings = 0;

do

{

if (nums.Length != 0)

{

ArrIteration(nums, NumOfStrings);

}

else

{

Console.WriteLine("Nums = пустой");

}

YellowColor("нажмите цифру от 1 до 4");

YellowColor(SplitMassiveMenu);

NumInput(out num);

switch (num)

{

case 1:

{

Console.Clear();

CreateArr(out nums, out NumOfStrings);

Console.ReadKey();

Console.Clear();

break;

}

case 2:

{

Console.Clear();

if(nums.Length != 0)

{

ArrIteration(nums, NumOfStrings);

}

else

{

Console.WriteLine("nums = пустой");

}

Console.ReadKey();

Console.Clear();

break;

}

case 3:

{

Console.Clear();

DeleteTheLonsgest(ref nums, ref NumOfStrings);

Console.ReadKey();

Console.Clear();

break;

}

case 4:

{

Console.Clear();

break;

}

default:

{

Console.Clear();

Console.ForegroundColor = ConsoleColor.Red;

SystemSounds.Beep.Play();

Console.WriteLine("!!!ошибка ввода!!!");

Console.ForegroundColor = ConsoleColor.White;

Console.ReadKey();

Console.Clear();

break;

}

}

}while(num != 4);

num = 0;

break;

}

case 4:

{

Console.

WriteLine(@"

/^--^\ /^--^\ /^--^\

\\_\_\_\_/ \\_\_\_\_/ \\_\_\_\_/

/ \ / \ / \

| | | | | |

\\_\_ \_\_/ \\_\_ \_\_/ \\_\_ \_\_/

|^|^|^|^|^|^|^|^|^|^|^|^\ \^|^|^|^/ /^|^|^|^|^\ \^|^|^|^|^|^|^|^|^|^|^|^|

| | | | | | | | | | | | |\ \| | |/ /| | | | | | \ \ | | | | | | | | | | |

########################/ /######\ \###########/ /#######################

| | | | | | | | | | | | \/| | | | \/| | | | | |\/ | | | | | | | | | | | |

|\_|\_|\_|\_|\_|\_|\_|\_|\_|\_|\_|\_|\_|\_|\_|\_|\_|\_|\_|\_|\_|\_|\_|\_|\_|\_|\_|\_|\_|\_|\_|\_|\_|\_|\_|\_| ");

Console.ReadKey();

break;

}

default:

{

Console.Clear();

Console.ForegroundColor = ConsoleColor.Red;

SystemSounds.Beep.Play();

Console.WriteLine("!!!ошибка ввода!!!");

Console.ForegroundColor = ConsoleColor.White;

Console.ReadKey();

Console.Clear();

break;

}

}

} while (num != 4);

num = 0;

}

}

}

**4**. **Тесты (с проверкой достаточности по критериям черного ящика) для каждой задачи.**

***Табл. 1. Тестирование методом черного ящика.* Для одномерного массива удаление первого четного элемента**

|  |  |  |  |
| --- | --- | --- | --- |
| **Номер** | **Исходные данные** | **Ожидаемый результат** | **Полученный результат** |
| **1** | Nums = {1,3,4} | Nums = {1,3} | Nums = {1,3} |
| **2** | Nums = {1} | Nums = {1} в массиве нет четных элементов | Nums = {1} в массиве нет четных элементов |
| **3** | Nums= {} | Nums= {} в пустом массиве нечего удалять | Nums= {} в пустом массиве нечего удалять |
| **4** | Nums = {2,4,6} провести операцию 3 раза | Nums= пустой массив | Nums= пустой массив |

***Табл. 2. Тестирование методом черного ящика.* Для двумерного массива добавление строки на выбранной позиции.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Номер** | **Исходные данные** | **Ожидаемый результат** | **Полученный результат** |
| **1** | Nums = {1,3,4}  {2,3,5}  Addstring{2,5,6}  Position = 1 | Nums = {2,5,6}  {1,3,4}  {2,3,5} | Nums = {2,5,6}  {1,3,4}  {2,3,5} |
| **2** | Nums = {}  Addstring{1,1,1}  Position = 2 | Ошибка (вы взяли слишком большую позицию) | Ошибка (вы взяли слишком большую позицию |
| **3** | Nums = {}  Addstring{1,1,1}  Position = 0 | Ошибка (порядок начинается с 1) | Ошибка (порядок начинается с 1) |
|  |  |  |  |
| **4** | Nums = {}  Addstring{1,1,1}  Position = 1 | Nums= {1,1,1} | Nums= {1,1,1} |

***Табл. 3. Тестирование методом черного ящика.* Для равного массива добавление удаление самой длинной строки.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Номер** | **Исходные данные** | **Ожидаемый результат** | **Полученный результат** |
| **1** | Nums = {1,3,4}  {2,3} | Nums = {2,3} | Nums = {2,3} |
| **2** | Nums = {1,3,4}  {пустая строка} | Nums = {пустая строка} | Nums = {пустая строка} |
| **3** | Nums= {} | В пустом массиве нечего удалять | В пустом массиве нечего удалять |
| **4** | Nums = {1,3,4}  {2,3}  Удаление 2 раза | Nums = пустой массив | Nums = пустой массив |

***Табл. 4. Тестирование методом черного ящика.* Для консоли*.***

|  |  |  |  |
| --- | --- | --- | --- |
| **Номер** | **Исходные данные** | **Ожидаемый результат** | **Полученный результат** |
| **1** | 1 | Создание массива | Создание массива |
| **2** | 2 | Перебор массива | Перебор массива |
| **3** | 3 | Выполнение функции над соответ. массивом | Выполнение функции над соответ. массивом |
| **4** | 4 | Назад\выход |  |
| **5** | Любая другая цифра или не цифра | Ошибка ввода | Ошибка ввода |