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 №5

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:**

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 двумерный 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.)

|  |  |  |  |
| --- | --- | --- | --- |
| Вариант | Одномерный массив | Двумерный массив | Рваный массив |
| 23 | Удалить все элементы с четными индексами | Добавить К столбцов, начиная со столбца с номером N | Добавить строку с заданным номером |

# **2.Анализ задачи:**

Основные функции

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Функция | Входные данные | Классы входных данных | Выходные данные | Классы выходных данных |
| 1 | Формирование одномерного массива | Количество элементов в массиве (elementi) | Целое число> 0 | Массив из целых чисел | Массив (massiv) |
| Символ  Целое число <=0  Вещественное число | Сообщение об ошибке | Неверный ввод / Массив пуст |
| 1.1 | Формирование массива с помощью ДСЧ | Количество элементов в массиве (elementi)  Нижний и верхний порог для ДСЧ (ArrayMin и ArrayMax) | Целое число> 0 | Массив из чисел | Массив (massiv) |
| Символ  Целое число <=0  Вещественное число | Сообщение об ошибке | Неверный ввод / Массив пуст |
| 1.2 | Формирование массива с клавиатуры | Количество элементов в массиве (elementi)  Число для каждого элемента | Целое число> 0  Целые числа | Массив из чисел | Массив (massiv) |
| elementi (Символ  Целое число <=0  Вещественное число)  Целые числа | Сообщение об ошибке | Неверный ввод / Массив пуст |
| 1.3 | Удаление элементов с чётным индексом | Массив  Размер массива | Массив не пустой | Массив из чисел | Новый массив (massiv) |
| Массив пустой | Сообщение об ошибке | Неверный ввод / Массив пуст |
| 2 | 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 |
| 2.1 | 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 |
| 2.2 | 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 |
| 2.3 | Добавление столбцов в массив | Массив  Размер массива  Номер, с которого добавляем  Количество добавленных элементов | * Массив пустой, * номер, с которого добавляем =1, * сколько добавляем >0 | Измененный массив  Измененный размер массива | Новый массив (dvumernii\_array) |
| * Массив не пустой, * номер, с которого добавляем =>1 и <= Размер массива, * Количество добавленных элементов >0 | Измененный массив  Измененный размер массива | Новый массив (dvumernii\_array) |
| * номер, с которого добавляем <1, * номер, с которого добавляем – символ, вещественное число | Сообщение об ошибке | Неверный ввод |
| * Количество добавленных элементов >0 * Количество добавленных элементов <= 0 * Количество добавленных элементов – символ, вещественное число | Сообщение об ошибке | Неверный ввод |
| 2.4 | Добавление с помощью ДСЧ | Массив  Размер массива  Номер, с которого добавляем  Количество добавленных элементов | * Массив пустой, * номер, с которого добавляем =1, * сколько добавляем >0 | Измененный массив  Измененный размер массива | Новый массив (dvumernii\_array) |
| * Массив не пустой, * номер, с которого добавляем =>1 и <= Размер массива, * Количество добавленных элементов >0 | Измененный массив  Измененный размер массива | Новый массив (dvumernii\_array) |
| * номер, с которого добавляем <1, * номер, с которого добавляем – символ, вещественное число | Сообщение об ошибке | Неверный ввод |
| * Количество добавленных элементов >0 * Количество добавленных элементов <= 0 * Количество добавленных элементов – символ, вещественное число | Сообщение об ошибке | Неверный ввод |
| 2.5 | Добавление с клавиатуры | Массив  Размер массива  Номер, с которого добавляем  Количество добавленных элементов  Число для каждого элемента | * Массив пустой, * номер, с которого добавляем =1, * сколько добавляем >0 * Добавляемые элементы – любое целое число | Измененный массив  Измененный размер массива | Новый массив (dvumernii\_array) |
| * Массив не пустой, * номер, с которого добавляем =>1 и <= Размер массива, * Количество добавленных элементов >0 * Добавляемые элементы – любое целое число | Измененный массив  Измененный размер массива | Новый массив (dvumernii\_array) |
| * номер, с которого добавляем <1, * номер, с которого добавляем – символ, вещественное число * Добавляемые элементы – вещественное число или символ | Сообщение об ошибке | Неверный ввод |
| * Количество добавленных элементов >0 * Количество добавленных элементов <= 0 * Количество добавленных элементов – символ, вещественное число * Добавляемые элементы – вещественное число или символ | Сообщение об ошибке | Неверный ввод |
| 3 | Формирование рваного массива | Количество строк в массиве (NumberOfStrings)  Количество элементов для каждой строки | NumberOfStrings – целое число> 0  Элементы в строке– целое число> 0 | Массив из целых чисел | Массив (dvumernii\_array) |
| NumberOfStrings – целое число> 0  Элементы в строке– целое число=> 0 | Массив из целых чисел с пустой строкой | Массив (dvumernii\_array) |
| Символ  Целое число <0  Вещественное число | Сообщение об ошибке | Неверный ввод / Массив пуст |
| 3.1 | Формирование массива с помощью ДСЧ | Количество строк в массиве (NumberOfStrings)  Количество элементов для каждой строки  Нижний и верхний порог для ДСЧ (ArrayMin и ArrayMax) | NumberOfStrings – целое число> 0  Элементы в строке– целое число> 0  ArrayMin и ArrayMax – любое целое число | Массив из целых чисел | Массив (rvanii\_array) |
| NumberOfStrings – целое число> 0  Элементы в строке– целое число=> 0  ArrayMin и ArrayMax – любое целое число | Массив из целых чисел | Массив (rvanii\_array) |
| Символ  Вещественное число  ArrayMin и ArrayMax – вещественное число или символ | Сообщение об ошибке | Неверный ввод / Массив пуст |
| 3.2 | Формирование массива с клавиатуры | Количество строк в массиве (NumberOfStrings)  Количество элементов для каждой строки  Число для каждого элемента | NumberOfStrings – целое число> 0  Элементы в строке– целое число> 0 | Массив из целых чисел | Массив (rvanii\_array) |
| NumberOfStrings – целое число> 0  Элементы в строке– целое число=> 0 | Массив из целых чисел | Массив (rvanii\_array) |
| Символ  Целое число <=0  Вещественное число | Сообщение об ошибке | Неверный ввод / Массив пуст |
| 3.3 | Добавление строк в массив | Массив  Размер массива  Номер, с которого добавляем  Количество добавленных элементов | * Массив пустой, * номер, с которого добавляем =1, * сколько добавляем >0 | Измененный массив  Измененный размер массива | Новый массив (rvanii\_array) |
| * Массив не пустой, * номер, с которого добавляем =>1 и <= Размер массива, * Количество добавленных элементов >0 | Измененный массив  Измененный размер массива | Новый массив (rvanii\_array) |
| * номер, с которого добавляем <1, * номер, с которого добавляем – символ, вещественное число | Сообщение об ошибке | Неверный ввод |
| * Количество добавленных элементов >0 * Количество добавленных элементов <= 0 * Количество добавленных элементов – символ, вещественное число | Сообщение об ошибке | Неверный ввод |
| 3.4 | Добавление с помощью ДСЧ | Массив  Размер массива  Номер, с которого добавляем  Количество добавленных элементов | * Массив пустой, * номер, с которого добавляем =1, * сколько добавляем >0 | Измененный массив  Измененный размер массива | Новый массив (rvanii\_array) |
| * Массив не пустой, * номер, с которого добавляем =>1 и <= Размер массива, * Количество добавленных элементов >0 | Измененный массив  Измененный размер массива | Новый массив (rvanii\_array) |
| * номер, с которого добавляем <1, * номер, с которого добавляем – символ, вещественное число | Сообщение об ошибке | Неверный ввод |
| * Количество добавленных элементов >0 * Количество добавленных элементов <= 0 * Количество добавленных элементов – символ, вещественное число | Сообщение об ошибке | Неверный ввод |
| 3.5 | Добавление с клавиатуры | Массив  Размер массива  Номер, с которого добавляем  Количество добавленных элементов  Число для каждого элемента | * Массив пустой, * номер, с которого добавляем =1, * сколько добавляем >0 * Добавляемые элементы – любое целое число | Измененный массив  Измененный размер массива | Новый массив (rvanii\_array) |
| * Массив не пустой, * номер, с которого добавляем =>1 и <= Размер массива, * Количество добавленных элементов >0 * Добавляемые элементы – любое целое число | Измененный массив  Измененный размер массива | Новый массив (rvanii\_array) |
| * номер, с которого добавляем <1, * номер, с которого добавляем – символ, вещественное число * Добавляемые элементы – вещественное число или символ | Сообщение об ошибке | Неверный ввод |
| * Количество добавленных элементов >0 * Количество добавленных элементов <= 0 * Количество добавленных элементов – символ, вещественное число * Добавляемые элементы – вещественное число или символ | Сообщение об ошибке | Неверный ввод |

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 |
| 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**

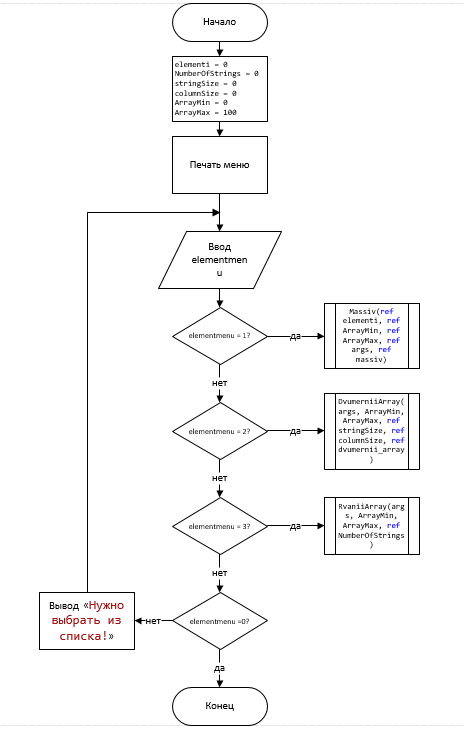
Основные функции

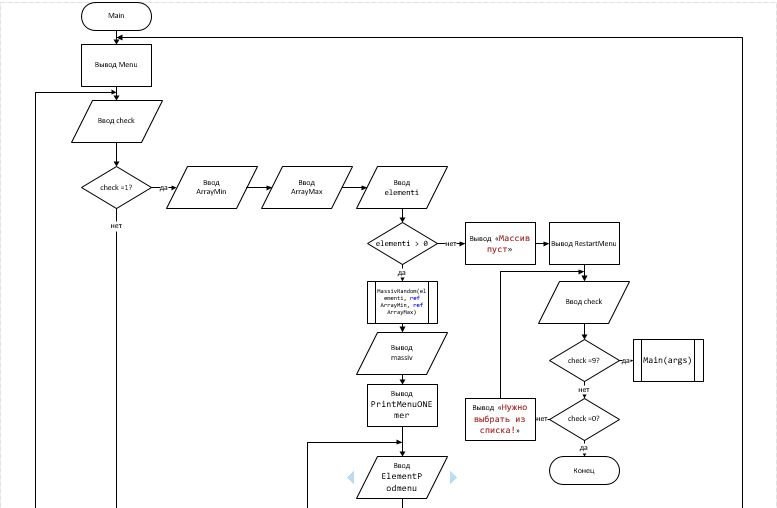
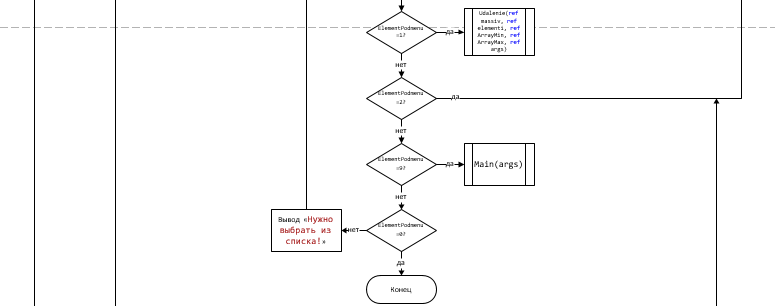
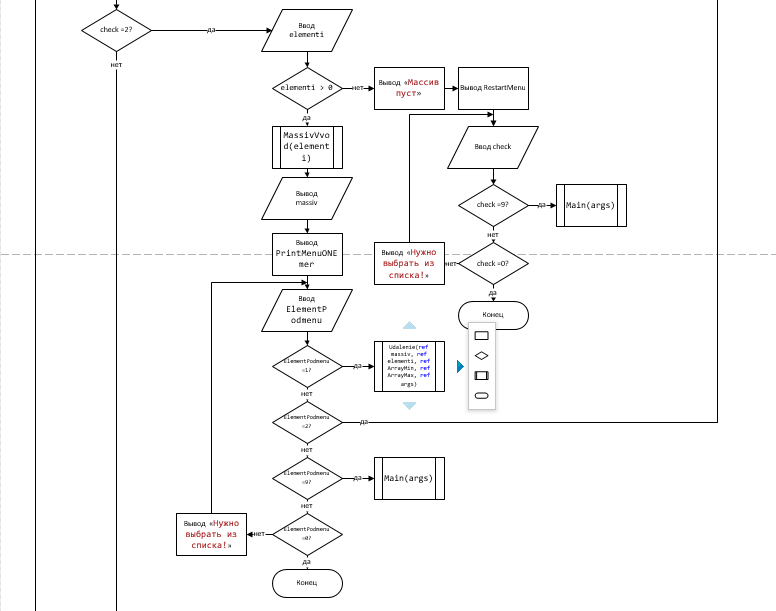
|  |  |  |  |
| --- | --- | --- | --- |
| № | Функции | Входные данные | Выходные данные |
| 1 | Massiv (формирование одномерного массива) | int elementi (количество элементов в массиве) | int[] massiv (массив) |
| 1.1 | MassivRandom (формирование одномерного массива с помощью ДСЧ) | int elementi (количество элементов в массиве)  int ArrayMin (нижняя граница для ДСЧ)  int ArrayMax (верхняя граница для ДСЧ) |
| 1.2 | MassivVvod (формирование одномерного массива с клавиатуры) | int elementi (количество элементов в массиве) |
| 1.3 | Udalenie (удаление элементов с чётным индексом) | int[] massiv (массив)  int elementi (количество элементов в массиве)  int ArrayMin (нижняя граница для ДСЧ)  int ArrayMax (верхняя граница для ДСЧ) | int[] massiv (массив)  int elementi |
| 2 | DvumerniiArray (формирование двумерного массива) | int stringSize (количество строк)  int columnSize (количество столбцов) | int[,] dvumernii\_array (двумерный массив) |
| 2.1 | RanomDvumerniiArray формирование двумерного массива с помощью ДСЧ) | int stringSize (количество строк)  int columnSize (количество столбцов)  int ArrayMin (нижняя граница для ДСЧ)  int ArrayMax (верхняя граница для ДСЧ) |
| 2.2 | VvodDvumerniiArray (формирование двумерного массива с клавиатуры) | int stringSize (количество строк)  int columnSize (количество столбцов) |
| 2.3 | RandomVvod (Добавление столбцов с помощью ДСЧ) | int[,] dvumernii\_array (двумерный массив)  int ArrayMin (нижняя граница для ДСЧ)  int ArrayMax (верхняя граница для ДСЧ)  int N (количество добавляемых столбцов)  int K (столбец с которого добавляем) | int[,] dvumernii\_array (двумерный массив)  int columnSize |
| 2.4 | RuchnoiVvod (Добавление столбцов с клавиатуры) | int[,] dvumernii\_array (двумерный массив)  int N (количество добавляемых столбцов)  int K (столбец с которого добавляем) |
| 3 | RvaniiArray(формирование рваного массива) | int NumberOfStrings (Количество строк) | int[][] rvanii\_array (Рваный массив) |
| 3.1 | RvaniiArrayRandom формирование рваного массива с помощью ДСЧ) | int NumberOfStrings (Количество строк)int ArrayMin (нижняя граница для ДСЧ)  int ArrayMax (верхняя граница для ДСЧ) |
| 3.2 | RvaniiArrayVvod (формирование рваного массива с клавиатуры) | int NumberOfStrings (Количество строк) |
| 3.3 | RandomDobavlenie (Добавление строк с помощью ДСЧ) | int[][] rvanii\_array (Рваный массив)  int ArrayMin (нижняя граница для ДСЧ)  int ArrayMax (верхняя граница для ДСЧ)  int plus\_string (добавляемая строка) | int[][] rvanii\_array (Рваный массив)  int NumberOfStrings |
| 3.4 | VvodDobavlenie (Добавление строк с клавиатуры) | int[][] rvanii\_array (Рваный массив)  int plus\_string (добавляемая строка) |

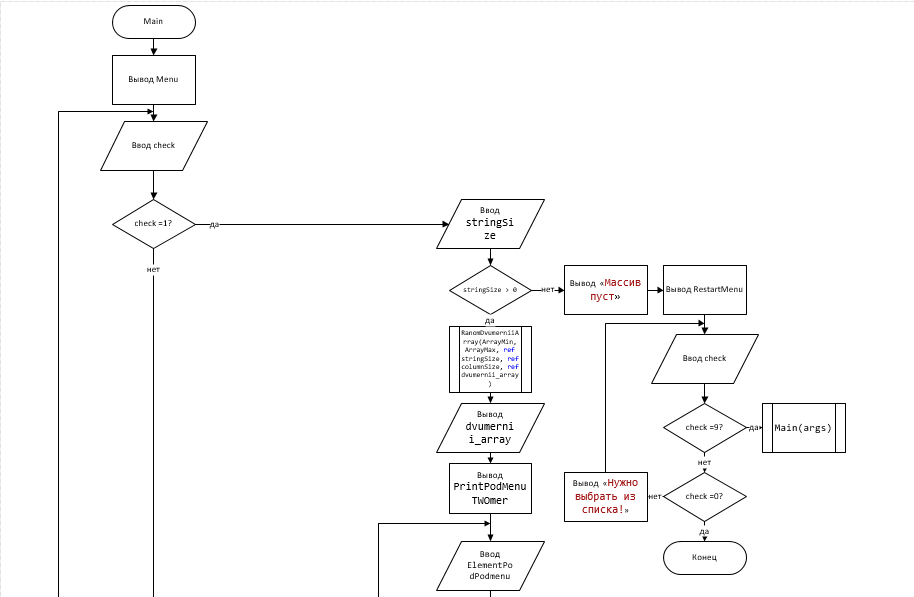
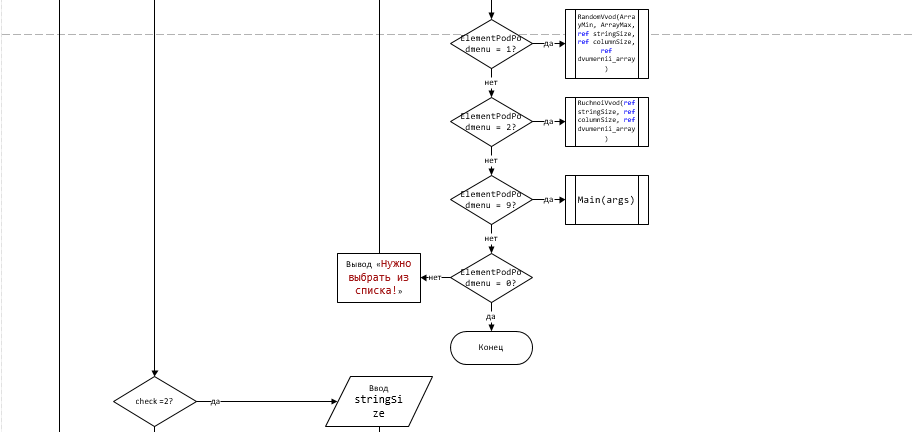
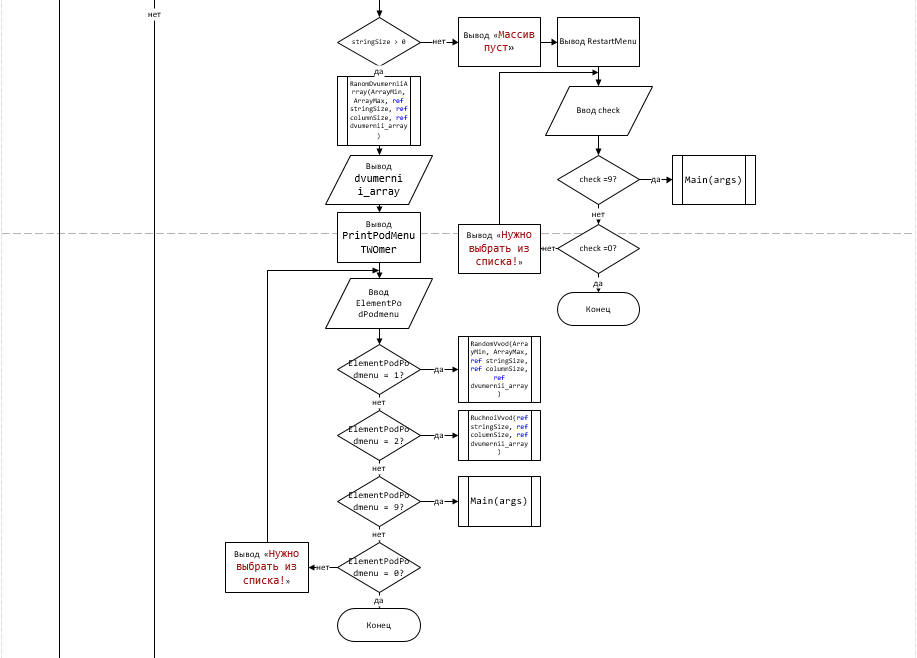
Вспомогательные функции

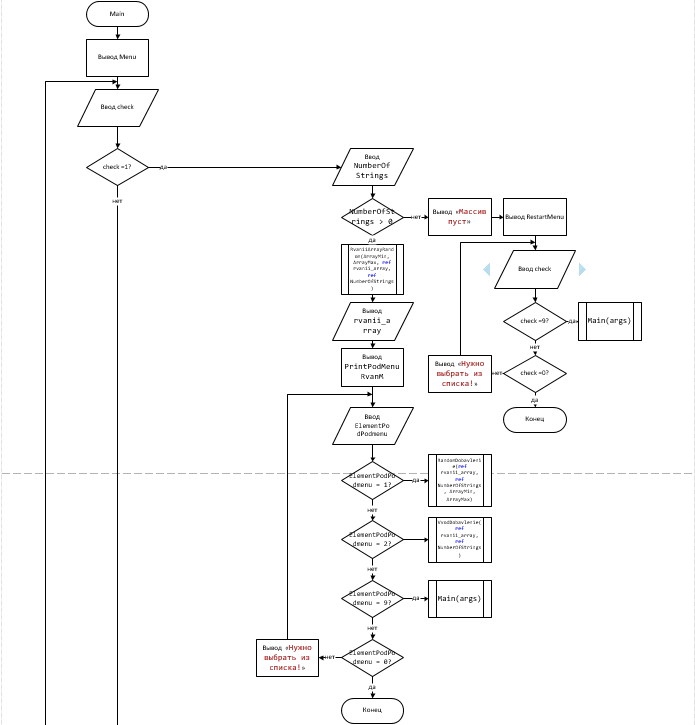
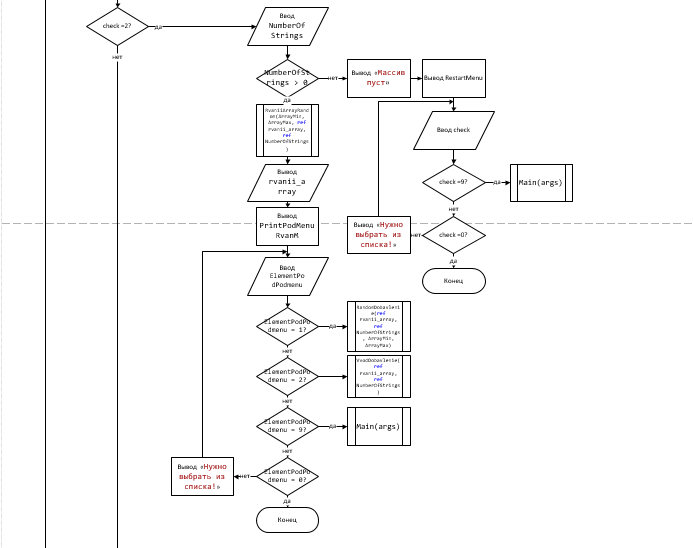
|  |  |  |  |
| --- | --- | --- | --- |
| № | Функции | Входные данные | Выходные данные |
| 1 | PrintMenuONEmer() | int check | Выбранный элемент |
| 2 | PrintMenuTWOmer() | int check | Выбранный элемент |
| 3 | PrintPodMenuTWOmer() | int check | Выбранный элемент |
| 4 | PrintMenuRvanM() | int check | Выбранный элемент |
| 5 | PrintPodMenuRvanM() | int check | Выбранный элемент |
| 6 | PrintErrorMenu() | int check | Выбранный элемент |
| 7 | ProverkaMassiva | int ArrayMin | ArrayMin или сообщение об ошибке |
| 8 | ProverkaVvoda() | int number | Число или или сообщение об ошибке |
| 9 | InputNumber | int number | Число или или сообщение об ошибке |
| 10 | RestartMenu | int check | Меню с перезапуском |

## **Блок-схема**

Блок-схема Main

Укрупнённая блок-схема Massiv

Укрупнённая блок-схема DvumerniiArray

Укрупнённая блок-схема RvaniiArray

# **4.Текст программы:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace LAB\_5\_CH

{

class Text\_Dialog

{

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

public static void PrintMenuONEmer()

{

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

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

Console.WriteLine("| 1) Удалить элементы с чётными индексами |");

Console.WriteLine("| 2) Построить новый одномерный массив |");

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

Console.WriteLine("| 0) Выход |");

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

Console.Write("Действие: ");

}

public static void PrintMenuTWOmer()

{

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

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

Console.WriteLine("| 1) Добавить K столбцов начиная со столбца под номером N |");

Console.WriteLine("| 2) Построить новый двумерный массив |");

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

Console.WriteLine("| 0) Выход |");

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

Console.Write("Действие: ");

}

public static void PrintPodMenuTWOmer()

{

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

Console.WriteLine("| Как вы хотите добавлять? |");

Console.WriteLine("| 1) Рандомно |");

Console.WriteLine("| 2) Вручную |");

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

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

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

Console.Write("Действие: ");

}

public static void PrintMenuRvanM()

{

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

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

Console.WriteLine("| 1) Добавить строку с заданным номером |");

Console.WriteLine("| 2) Построить новый рваный массив |");

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

Console.WriteLine("| 0) Выход |");

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

Console.Write("Действие: ");

}

public static void PrintPodMenuRvanM()

{

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

Console.WriteLine("| Как вы хотите добавлять? |");

Console.WriteLine("| 1) Рандомно |");

Console.WriteLine("| 2) Вручную |");

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

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

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

Console.Write("Действие: ");

}

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 ProverkaMassiva(string title, int ArrayMin, int right) //Проверка на ввод массива и его элементов

//Тут больше и нечего добавлять

//Просто небольшая функция на проверку

{

bool ok = false;

int number = ArrayMin;

do

{

Console.WriteLine(title);

try

{

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

number = Convert.ToInt32(buf);

if (number >= ArrayMin && number < right)

ok = true;

else

{

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

ok = false;

}

}

catch (FormatException)

{

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

ok = false;

}

catch (OverflowException)

{

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

ok = false;

}

} while (!ok);

return number;

}

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 int InputNumber(string Text, int minSize, int maxSize)

{

int number = 0;

bool ok = false;

do

{

try

{

Console.WriteLine(Text);

number = Convert.ToInt32(Console.ReadLine());

if (number >= minSize && number < maxSize) ok = true;

else ok = false;

}

catch (FormatException)

{

Console.WriteLine("Ошибка при вводе числа");

ok = false;

}

catch (OverflowException)

{

Console.WriteLine("Ошибка при вводе числа");

ok = false;

}

} while (!ok);

return number;

}

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

}

class Program

{

const int MinSize = 1;

const int MaxSize = 100;

const int MinNumber = -100;

const int MaxNumber = 100;

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;

}

}

//---ОДНОМЕРНЫЙ МАССИВ----------------------------------------------------------------------------------------------------

#region

static void Massiv(ref int elementi, ref int ArrayMin, ref int ArrayMax, ref string[] args, ref int[] massiv) //Меню с массивами

{

do

{

Console.Clear();

massiv = 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.WriteLine("----------Формирование массива----------------");

Console.Write("Введите нижнюю границу массива:");

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

do

{

Console.Write("Введите верхнюю границу массива:");

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

if (ArrayMax < ArrayMin)

{

Console.WriteLine("Верхняя граница не может быть меньше нижней!!!");

}

} while (ArrayMax < ArrayMin);

Console.Write("Введите количество элементов в массиве:");

elementi = Search\_for\_Bugs.ProverkaMassiva("", MinSize, MaxSize);

massiv = MassivRandom(elementi, ref ArrayMin, ref ArrayMax);

if (elementi > 0)

{

Console.Clear();

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

foreach (var i in massiv) //Построение нового массива

{

Console.Write(" " + i);

}

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

do

{

Text\_Dialog.PrintMenuONEmer();

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

switch (ElementPodmenu) //Выбор действия из меню

{

case 1:

Udalenie(ref massiv, ref elementi, ref ArrayMin, ref ArrayMax, ref args);

break;

case 2:

Massiv(ref elementi, ref ArrayMin, ref ArrayMax, ref args, ref massiv);

break;

case 9:

Main(args);

break;

case 0:

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

break;

default:

Text\_Dialog.PrintErrorMenu();

continue;

}

} while (true);

}

else

{

RestartMenu(args);

}

break;

case 2:

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

Console.Write("Введите количество элементов в массиве:");

elementi = Search\_for\_Bugs.ProverkaMassiva("", MinSize, MaxSize);

if (elementi > 0)

{

massiv = MassivVvod(elementi);

Console.Clear();

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

foreach (var i in massiv) //Построение нового массива

{

Console.Write(" " + i);

}

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

do

{

Text\_Dialog.PrintMenuONEmer();

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

switch (ElementPodmenu) //Выбор действия из меню

{

case 1:

Udalenie(ref massiv, ref elementi, ref ArrayMin, ref ArrayMax, ref args);

break;

case 2:

Massiv(ref elementi, ref ArrayMin, ref ArrayMax, ref args, ref massiv);

break;

case 9:

Main(args);

break;

case 0:

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

break;

default:

Text\_Dialog.PrintErrorMenu();

continue;

}

} while (true);

}

else

{

RestartMenu(args);

}

break;

case 9:

Massiv(ref elementi, ref ArrayMin, ref ArrayMax, ref args, ref massiv);

break;

case 0:

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

break;

default:

Text\_Dialog.PrintErrorMenu();

continue;

}

} while (!true);

}

static int[] MassivRandom(int elementi, ref int ArrayMin, ref int ArrayMax)

{

Random rnd = new Random();

int[] massiv = new int[elementi];

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

{

massiv[i] = rnd.Next(ArrayMin, ArrayMax);

}

return massiv;

}

static int[] MassivVvod(int elementi)

{

int[] massiv = new int[elementi];

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

{

massiv[i] = Search\_for\_Bugs.ProverkaMassiva("Введите элемент массива из диапазона от -100 до 100", MinNumber, MaxNumber);

}

return massiv;

}

static void Udalenie(ref int[] massiv, ref int elementi, ref int ArrayMin, ref int ArrayMax, ref string[] args)

{

Console.Clear();

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

foreach (var i in massiv) //Построение нового массива

{

Console.Write(" " + i);

}

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

int[] NewMassiv = new int[massiv.Length / 2];

if (massiv.Length %2 == 0)

{

NewMassiv = new int[massiv.Length / 2];

}

else

{

NewMassiv = new int[(massiv.Length / 2)+1];

}

int v = 0;

if (elementi > 1)

{

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

if (i % 2 == 0)

{

NewMassiv[v] = massiv[i];

v++;

}

}

else

{

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

}

foreach (var i in NewMassiv) //Построение нового массива

{

Console.Write(" " + i);

}

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

do

{

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

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

Console.WriteLine("| 1) Построить новый одномерный массив |");

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

Console.WriteLine("| 0) Выход |");

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

Console.Write("Действие: ");

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

switch (ElementPodmenu) //Выбор действия из меню

{

case 1:

Massiv(ref elementi, ref ArrayMin, ref ArrayMax, ref args, ref massiv);

break;

case 9:

Main(args);

break;

case 0:

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

break;

default:

Text\_Dialog.PrintErrorMenu();

continue;

}

} while (true);

}

#endregion

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

//---ДВУМЕРНЫЙ МАССИВ-----------------------------------------------------------------------------------------------------

#region

static int[,] DvumerniiArray(string[] args, int ArrayMin, int ArrayMax, ref int stringSize, ref int columnSize, ref int[,] 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();

RanomDvumerniiArray(ArrayMin, ArrayMax, ref stringSize, ref columnSize, ref dvumernii\_array);

if (stringSize > 0)

{

do

{

Text\_Dialog.PrintMenuTWOmer();

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

switch (ElementPodmenu) //Выбор действия из меню

{

case 1:

Text\_Dialog.PrintPodMenuTWOmer();

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

switch(ElementPodPodmenu)

{

case 1:

Console.Clear();

RandomVvod(ArrayMin, ArrayMax, ref stringSize, ref columnSize, ref dvumernii\_array);

break;

case 2:

Console.Clear();

RuchnoiVvod(ref stringSize, ref columnSize, ref dvumernii\_array);

break;

case 9:

Main(args);

break;

case 0:

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

break;

default:

Text\_Dialog.PrintErrorMenu();

continue;

}

break;

case 2:

DvumerniiArray(args, ArrayMin, ArrayMax, ref stringSize, ref columnSize, ref dvumernii\_array);

break;

case 9:

Main(args);

Console.Clear();

break;

case 0:

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

break;

default:

Text\_Dialog.PrintErrorMenu();

continue;

}

} while (true);

}

else

{

RestartMenu(args);

}

break;

case 2:

Console.Clear();

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

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

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

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

if (stringSize > 0)

{

do

{

Text\_Dialog.PrintMenuTWOmer();

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

switch (ElementPodmenu) //Выбор действия из меню

{

case 1:

Text\_Dialog.PrintPodMenuTWOmer();

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

switch (ElementPodPodmenu)

{

case 1:

Console.Clear();

RandomVvod(ArrayMin, ArrayMax, ref stringSize, ref columnSize, ref dvumernii\_array);

break;

case 2:

Console.Clear();

RuchnoiVvod(ref stringSize, ref columnSize, ref dvumernii\_array);

break;

case 9:

Main(args);

break;

case 0:

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

break;

default:

Text\_Dialog.PrintErrorMenu();

continue;

}

break;

case 2:

DvumerniiArray(args, ArrayMin, ArrayMax, ref stringSize, ref columnSize, ref dvumernii\_array);

break;

case 9:

Main(args);

break;

case 0:

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

break;

default:

Text\_Dialog.PrintErrorMenu();

continue;

}

} while (true);

}

else

{

RestartMenu(args);

}

break;

case 9:

Main(args);

break;

case 0:

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

break;

default:

Text\_Dialog.PrintErrorMenu();

continue;

}

return dvumernii\_array;

} while (true);

}

static int[,] RanomDvumerniiArray(int ArrayMin, int ArrayMax, ref int stringSize, ref int columnSize, ref int[,] dvumernii\_array)

{

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

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

Console.Write("\nВведите нижнюю границу массива:");

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

do

{

Console.Write("Введите верхнюю границу массива:");

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

if (ArrayMax < ArrayMin)

{

Console.WriteLine("Верхняя граница не может быть меньше нижней!!!");

}

} while (ArrayMax < ArrayMin);

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

Random rand = new Random();

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

{

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

{

dvumernii\_array[i, j] = rand.Next(ArrayMin, ArrayMax);

}

}

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(dvumernii\_array[i, j] + "\t ");

}

Console.WriteLine();

}

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

return dvumernii\_array;

}

static int[,] VvodDvumerniiArray(ref int stringSize, ref int columnSize, ref int[,] dvumernii\_array)

{

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

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

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

dvumernii\_array = new int[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.InputNumber("Введите элемент матрицы", MinNumber, MaxNumber);

}

}

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(dvumernii\_array[i, j] + "\t ");

}

Console.WriteLine();

}

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

return dvumernii\_array;

}

static int[,] RandomVvod( int ArrayMin, int ArrayMax, ref int stringSize, ref int columnSize, ref int[,] dvumernii\_array)

{

{

Console.Clear(); //Очищение консоли

int N = 1, //номер с которого добавляем

K = 1; //Количество добавляемых столбцов

bool ok\_K = true, //bool на проверку

ok\_N = true;

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(dvumernii\_array[i, j] + "\t ");

}

Console.WriteLine();

}

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

do

{

ok\_N = true;

N = Search\_for\_Bugs.InputNumber("Введите номер столбца (N), с которого нужно добавлять:", MinSize, MaxSize); //проверка на правильность

if (N <= 0 || N > columnSize + 1)

{

ok\_N = false;

Console.WriteLine("Не правильно задан номер, с которого нужно добавлять!");

}

} while (!ok\_N);

do

{

ok\_K = true;

K = Search\_for\_Bugs.InputNumber("Введите количество столбцов (K) для добавления:", MinSize, MaxSize); //проверка на правильность

if (columnSize + K > MaxSize)

{

ok\_K = false;

Console.WriteLine("Не правильно задано количество столбцов!");

}

} while (!ok\_K);

Console.Write("\nВведите нижнюю границу массива:");

ArrayMin = Search\_for\_Bugs.ProverkaVvoda(); //проверка на правильность

do

{

Console.Write("Введите верхнюю границу массива:");

ArrayMax = Search\_for\_Bugs.ProverkaVvoda(); //проверка на правильность

if (ArrayMax < ArrayMin)

{

Console.WriteLine("Верхняя граница не может быть меньше нижней!!!");

}

} while (ArrayMax < ArrayMin);

int[,] temp = new int[stringSize, K];

Random rand = new Random();

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

{

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

{

temp[i, j] = rand.Next(ArrayMin, ArrayMax);

}

}

columnSize += K; // необходимо при следующем добавлении новых столбцов

int[,] New\_Array = new int[stringSize, columnSize]; //создание нового двумерного массива

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

{

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

{

//Напоминалка

//при смене N на (N-1) столбцы будут добавляться начиная с 0

New\_Array[i, Index] = dvumernii\_array[i, j];

if (N > dvumernii\_array.GetLength(0))

{

if (j >= (N - 2))

{

for (int y = 0; y < temp.GetLength(1); y++)

{

Index++;

New\_Array[i, Index] = temp[i, y];

}

}

}

else

{

if (j == (N - 1))

{

for (int y = 0; y < temp.GetLength(1); y++)

{

New\_Array[i, Index] = temp[i, y];

Index++;

}

}

New\_Array[i, Index] = dvumernii\_array[i, j];

}

}

}

dvumernii\_array = New\_Array;

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

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

{

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

{

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

}

Console.WriteLine();

}

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

}

return dvumernii\_array;

}

static int[,] RuchnoiVvod( ref int stringSize, ref int columnSize, ref int[,] dvumernii\_array)

{

{

Console.Clear(); //Очищение консоли

int N = 1,

K = 1;

bool ok\_K = true,

ok\_N = true;

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(dvumernii\_array[i, j] + "\t ");

}

Console.WriteLine();

}

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

do

{

ok\_N = true;

N = Search\_for\_Bugs.InputNumber("Введите номер столбца (N), с которого нужно добавлять:", MinSize, MaxSize);

if (N <= 0 || N > columnSize + 1)

{

ok\_N = false;

Console.WriteLine("Не правильно задан номер, с которого нужно добавлять!");

}

} while (!ok\_N);

do

{

ok\_K = true;

K = Search\_for\_Bugs.InputNumber("Введите количество столбцов (K) для добавления:", MinSize, MaxSize);

if (columnSize + K > MaxSize)

{

ok\_K = false;

Console.WriteLine("Не правильно задано количество столбцов!");

}

} while (!ok\_K);

int[,] temp = new int[stringSize, K];

Random rand = new Random();

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

{

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

{

temp[i, j] = Search\_for\_Bugs.InputNumber("Введите элемент матрицы", MinNumber, MaxNumber);

}

}

columnSize += K; // необходимо при следующем добавлении новых столбцов

int[,] New\_Array = new int[stringSize, columnSize];

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

{

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

{

//Напоминалка

//при смене N на (N-1) столбцы будут добавляться начиная с 0

New\_Array[i, Index] = dvumernii\_array[i, j];

if (N > dvumernii\_array.GetLength(0))

{

if (j >= (N - 2))

{

for (int y = 0; y < temp.GetLength(1); y++)

{

Index++;

New\_Array[i, Index] = temp[i, y];

}

}

}

else

{

if (j == (N - 1))

{

for (int y = 0; y < temp.GetLength(1); y++)

{

New\_Array[i, Index] = temp[i, y];

Index++;

}

}

New\_Array[i, Index] = dvumernii\_array[i, j];

}

}

}

dvumernii\_array = New\_Array;

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

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

{

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

{

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

}

Console.WriteLine();

}

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

}

return dvumernii\_array;

}

#endregion

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

//---РВАНЫЙ МАССИВ--------------------------------------------------------------------------------------------------------

#region

static int[][] RvaniiArray(string[] args, int ArrayMin, int ArrayMax, ref int NumberOfStrings)

{

do

{

Console.Clear();

int[][] rvanii\_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();

RvaniiArrayRandom(ArrayMin, ArrayMax, ref rvanii\_array, ref NumberOfStrings);

if (NumberOfStrings > 0)

{

do

{

Text\_Dialog.PrintMenuRvanM();

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

switch (ElementPodmenu) //Выбор действия из меню

{

case 1:

Text\_Dialog.PrintPodMenuRvanM();

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

switch(ElementPodPodmenu)

{

case 1:

Console.Clear();

RandomDobavlenie(ref rvanii\_array, ref NumberOfStrings, ArrayMin, ArrayMax);

break;

case 2:

Console.Clear();

VvodDobavlenie(ref rvanii\_array, ref NumberOfStrings);

break;

case 9:

Main(args);

break;

case 0:

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

break;

default:

Text\_Dialog.PrintErrorMenu();

continue;

}

break;

case 2:

RvaniiArray(args, ArrayMin, ArrayMax, ref NumberOfStrings);

break;

case 9:

Main(args);

Console.Clear();

break;

case 0:

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

break;

default:

Text\_Dialog.PrintErrorMenu();

continue;

}

} while (true);

}

else

{

RestartMenu(args);

}

break;

case 2:

Console.Clear();

RvaniiArrayVvod(ref rvanii\_array, ref NumberOfStrings);

if (NumberOfStrings > 0)

{

do

{

Text\_Dialog.PrintMenuRvanM();

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

switch (ElementPodmenu) //Выбор действия из меню

{

case 1:

Text\_Dialog.PrintPodMenuRvanM();

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

switch (ElementPodPodmenu)

{

case 1:

Console.Clear();

RandomDobavlenie(ref rvanii\_array, ref NumberOfStrings, ArrayMin, ArrayMax);

break;

case 2:

Console.Clear();

VvodDobavlenie(ref rvanii\_array, ref NumberOfStrings);

break;

case 9:

Main(args);

break;

case 0:

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

break;

default:

Text\_Dialog.PrintErrorMenu();

continue;

}

break;

case 2:

RvaniiArray(args, ArrayMin, ArrayMax, ref NumberOfStrings);

break;

case 9:

Main(args);

break;

case 0:

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

break;

default:

Text\_Dialog.PrintErrorMenu();

continue;

}

} while (true);

}

else

{

RestartMenu(args);

}

break;

case 9:

Main(args);

break;

default:

Text\_Dialog.PrintErrorMenu();

continue;

}

return rvanii\_array;

} while (true);

}

static int[][] RvaniiArrayRandom(int ArrayMin, int ArrayMax, ref int[][] rvanii\_array, ref int NumberOfStrings)

{

int ColumnSize,

Roll = 1;

NumberOfStrings = 1;

Random rnd = new Random();

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

do

{

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

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

if (NumberOfStrings <= 0)

{

Console.WriteLine("Количество строк не может быть меньше или равно 0\n");

}

}

while (NumberOfStrings <= 0);

Console.Write("Введите нижнюю границу массива:");

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

do

{

Console.Write("Введите верхнюю границу массива:");

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

if (ArrayMax < ArrayMin)

{

Console.WriteLine("Верхняя граница не может быть меньше нижней!!!");

}

} while (ArrayMax < ArrayMin);

rvanii\_array = new int[NumberOfStrings][];

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

{

Console.Write("Введите количество Элементов в " + Roll + " строке: ");

do

{

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

if(ColumnSize <= 0)

{

Console.WriteLine("Я не хочу чтобы Массив пустовал!!!");

}

} while (ColumnSize <= 0);

rvanii\_array[i] = new int[ColumnSize];

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

{

rvanii\_array[i][j] = rnd.Next(ArrayMin, ArrayMax);

}

Roll++;

}

Console.Clear();

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

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

{

for (int j = 0; j < rvanii\_array[i].Length; j++)

Console.Write(rvanii\_array[i][j].ToString() + " ");

Console.WriteLine();

}

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

return rvanii\_array;

}

static int[][] RvaniiArrayVvod(ref int[][] rvanii\_array, ref int NumberOfStrings)

{

int ColumnSize,

Roll = 1;

NumberOfStrings = 1;

Random rnd = new Random();

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

do

{

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

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

if (NumberOfStrings <= 0)

{

Console.WriteLine("Количество строк не может быть меньше или равно 0\n");

}

}

while (NumberOfStrings <= 0);

rvanii\_array = new int[NumberOfStrings][];

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

{

Console.Write("Введите количество Элементов в " + Roll + " строке: ");

do

{

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

if (ColumnSize <= 0)

{

Console.WriteLine("Я не хочу чтобы Массив пустовал!!!");

}

} while (ColumnSize <= 0);

rvanii\_array[i] = new int[ColumnSize];

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

{

rvanii\_array[i][j] = Search\_for\_Bugs.InputNumber("Введите элемент:", MinNumber, MaxNumber);

}

Roll++;

}

Console.Clear();

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

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

{

for (int j = 0; j < rvanii\_array[i].Length; j++)

Console.Write(rvanii\_array[i][j].ToString() + " ");

Console.WriteLine();

}

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

return rvanii\_array;

}

static int[][] RandomDobavlenie(ref int[][] rvanii\_array, ref int NumberOfStrings, int ArrayMin, int ArrayMax)

{

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

for (int i = 0; i < rvanii\_array.Length; i++)

{

for (int j = 0; j < rvanii\_array[i].Length; j++)

Console.Write(rvanii\_array[i][j].ToString() + " ");

Console.WriteLine();

if (rvanii\_array[i].Length == 0)

{

Console.ForegroundColor = ConsoleColor.Cyan;

Console.Write("пустая строка"); ;

Console.ResetColor();

}

}

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

Random rnd = new Random();

int ColumnSize,

plus\_string;

Console.Write("Введите номер строки: ");

do

{

plus\_string = Search\_for\_Bugs.ProverkaVvoda();

if ((plus\_string > rvanii\_array.Length + 1) || (plus\_string <= 0))

{

Console.WriteLine("Строка не должна выходить за пределы массива!!!");

}

}

while ((plus\_string > rvanii\_array.Length + 1) || (plus\_string <= 0));

plus\_string--;

Console.Write("Введите нижнюю границу массива:");

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

do

{

Console.Write("Введите верхнюю границу массива:");

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

if (ArrayMax < ArrayMin)

{

Console.WriteLine("Верхняя граница не может быть меньше нижней!!!");

}

} while (ArrayMax < ArrayMin);

int[][] new\_dobavlenie = new int[rvanii\_array.Length + 1][];

for (int i = 0; i < new\_dobavlenie.Length; i++)

{

if (i < plus\_string)

{

new\_dobavlenie[i] = new int[rvanii\_array[i].Length];

for (int j = 0; j < rvanii\_array[i].Length; j++)

new\_dobavlenie[i][j] = rvanii\_array[i][j];

}

if ((i >= plus\_string) && (i != rvanii\_array.Length))

{

new\_dobavlenie[i + 1] = new int[rvanii\_array[i].Length];

for (int j = 0; j < rvanii\_array[i].Length; j++)

new\_dobavlenie[i + 1][j] = rvanii\_array[i][j];

}

if (i == plus\_string)

{

Console.Write("Введите количество Элементов в задаваемой строке: ");

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

new\_dobavlenie[i] = new int[ColumnSize];

Random randm = new Random();

for (int j = 0; j < new\_dobavlenie[i].Length; j++)

{

new\_dobavlenie[i][j] = randm.Next(ArrayMin, ArrayMax);

}

}

}

rvanii\_array = new int[rvanii\_array.Length + 1][];

for (int i = 0; i < rvanii\_array.Length; i++)

{

rvanii\_array[i] = new int[new\_dobavlenie[i].Length];

for (int j = 0; j < rvanii\_array[i].Length; j++)

rvanii\_array[i][j] = new\_dobavlenie[i][j];

}

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

for (int i = 0; i < rvanii\_array.Length; i++)

{

for (int j = 0; j < rvanii\_array[i].Length; j++)

{

Console.Write(rvanii\_array[i][j].ToString() + " ");

}

Console.WriteLine();

if (rvanii\_array[i].Length == 0)

{

Console.ForegroundColor = ConsoleColor.Cyan;

Console.Write("пустая строка"); ;

Console.ResetColor();

}

}

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

rvanii\_array = new\_dobavlenie;

return rvanii\_array;

}

static int[][] VvodDobavlenie(ref int[][] rvanii\_array, ref int NumberOfStrings)

{

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

for (int i = 0; i < rvanii\_array.Length; i++)

{

for (int j = 0; j < rvanii\_array[i].Length; j++)

Console.Write(rvanii\_array[i][j].ToString() + " ");

Console.WriteLine();

if (rvanii\_array[i].Length == 0)

{

Console.ForegroundColor = ConsoleColor.Cyan;

Console.Write("пустая строка"); ;

Console.ResetColor();

}

}

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

Random rnd = new Random();

int ColumnSize,

plus\_string;

Console.Write("Введите номер строки: ");

do

{

plus\_string = Search\_for\_Bugs.ProverkaVvoda();

if ((plus\_string > rvanii\_array.Length + 1) || (plus\_string <= 0))

{

Console.WriteLine("Строка не должна выходить за пределы массива!!!");

}

}

while ((plus\_string > rvanii\_array.Length + 1) || (plus\_string <= 0));

plus\_string--;

int[][] new\_dobavlenie = new int[rvanii\_array.Length + 1][];

for (int i = 0; i < new\_dobavlenie.Length; i++)

{

if (i < plus\_string)

{

new\_dobavlenie[i] = new int[rvanii\_array[i].Length];

for (int j = 0; j < rvanii\_array[i].Length; j++)

new\_dobavlenie[i][j] = rvanii\_array[i][j];

}

if ((i >= plus\_string) && (i != rvanii\_array.Length))

{

new\_dobavlenie[i + 1] = new int[rvanii\_array[i].Length];

for (int j = 0; j < rvanii\_array[i].Length; j++)

new\_dobavlenie[i + 1][j] = rvanii\_array[i][j];

}

if (i == plus\_string)

{

Console.Write("Введите количество Элементов в задаваемой строке: ");

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

new\_dobavlenie[i] = new int[ColumnSize];

for (int j = 0; j < new\_dobavlenie[i].Length; j++)

{

bool okey = false;

do

{

Console.WriteLine("Введите {0} элемент", j + 1);

try

{

new\_dobavlenie[i][j] = Search\_for\_Bugs.ProverkaVvoda();

okey = true;

}

catch

{

okey = false;

Console.WriteLine("Неверный ввод");

}

} while (!okey);

}

}

}

rvanii\_array = new int[rvanii\_array.Length + 1][];

for (int i = 0; i < rvanii\_array.Length; i++)

{

rvanii\_array[i] = new int[new\_dobavlenie[i].Length];

for (int j = 0; j < rvanii\_array[i].Length; j++)

rvanii\_array[i][j] = new\_dobavlenie[i][j];

}

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

for (int i = 0; i < rvanii\_array.Length; i++)

{

for (int j = 0; j < rvanii\_array[i].Length; j++)

Console.Write(rvanii\_array[i][j].ToString() + " ");

Console.WriteLine();

if (rvanii\_array[i].Length == 0)

{

Console.ForegroundColor = ConsoleColor.Cyan;

Console.Write("пустая строка"); ;

Console.ResetColor();

}

}

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

rvanii\_array = new\_dobavlenie;

return rvanii\_array;

}

#endregion

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

static void Main(string[] args)

{

Console.Clear(); //Очищение консоли

int elementi = 0,

NumberOfStrings = 0,

stringSize = 0,

columnSize = 0,

ArrayMin = 0,

ArrayMax = 100;

int[] massiv = null;

int[,] dvumernii\_array = null;

do

{

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

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

Console.WriteLine("| 1) Построить одномерный массив |");

Console.WriteLine("| 2) Построить двумерный массив |");

Console.WriteLine("| 3) Построить рваный массив |");

Console.WriteLine("| 0) Выход |");

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

Console.Write("Действие: ");

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

switch (elementmenu) //Выбор действия из меню

{

case 1:

{

Massiv(ref elementi, ref ArrayMin, ref ArrayMax, ref args, ref massiv);

}

break;

case 2:

{

DvumerniiArray(args, ArrayMin, ArrayMax, ref stringSize, ref columnSize, ref dvumernii\_array);

}

break;

case 3:

{

RvaniiArray(args, ArrayMin, ArrayMax, ref NumberOfStrings);

}

break;

case 0:

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

break;

default:

Text\_Dialog.PrintErrorMenu();

continue;

}

} while (true);

}

}

}

# **5.Тесты:**

Формирование одномерного массива (с помощью ДСЧ):

|  |  |  |  |
| --- | --- | --- | --- |
| **Тест** | **Входные данные** | **Ожидаемый результат** | **Реальный результат** |
| 1 | 5 | 5 чисел | -90 0 27 54 -67 |
| 2 | 0 | Сообщение об ошибке | Неверный ввод |
| 3 | -3 | Сообщение об ошибке | Неверный ввод |
| 4 | 1,34 | Сообщение об ошибке | Неверный ввод |
| 5 | q | Сообщение об ошибке | Неверный ввод |
| 6 | 1 | 1 число | 10 |
| 7 | 1 | 1 | 1 |
| 8 | 1111111111111 | Сообщение об ошибке | Неверный ввод |
| 9 | 100 | Сообщение об ошибке | Неверный ввод |
| 10 | -100 | -100 | -100 |
| 11 | 0 | 0 | 0 |

Формирование одномерного массива (с клавиатуры):

|  |  |  |  |
| --- | --- | --- | --- |
| **Тест** | **Входные данные** | **Ожидаемый результат** | **Реальный результат** |
| 1 | -90 0 27 54 -67 | -90 0 27 54 -67 | -90 0 27 54 -67 |
| 2 | 0 | 0 | 0 |
| 3 | -3 | Сообщение об ошибке | Неверный ввод |
| 4 | 1,34 | Сообщение об ошибке | Неверный ввод |
| 5 | q | Сообщение об ошибке | Неверный ввод |
| 6 | 10 | 10 | 10 |
| 7 | 1 | 1 | 1 |
| 8 | 1111111111111 | Сообщение об ошибке | Неверный ввод |
| 9 | 100 | Сообщение об ошибке | Неверный ввод |
| 10 | -100 | -100 | -100 |

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 | 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 | 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 |
| 11 | 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 |

Добавление в двумерный массив (с помощью ДСЧ):

|  |  |  |  |
| --- | --- | --- | --- |
| **Тест** | **Входные данные** | **Ожидаемый результат** | **Реальный результат** |
| 1 | 1 1 1 1 1 1  1 1 1 1 1 1  1 1 1 1 1 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 | 1 1 1 1 1 1  1 1 1 1 1 1  1 1 1 1 1 1  0 строка | Сообщение об ошибке | Неверный ввод |
| 3 | 1 1 1 1 1 1  1 1 1 1 1 1  1 1 1 1 1 1  -3 строка | Сообщение об ошибке | Неверный ввод |
| 4 | 1 1 1 1 1 1  1 1 1 1 1 1  1 1 1 1 1 1  1,34 строка | Сообщение об ошибке | Неверный ввод |
| 5 | 1 1 1 1 1 1  1 1 1 1 1 1  1 1 1 1 1 1  Q строка | Сообщение об ошибке | Неверный ввод |
| 6 | 1 1 1 1 1 1  1 1 1 1 1 1  1 1 1 1 1 1  1 строка  нижняя граница -100,  верхняя 100 | 5 цифр  1 1 1 1 1 1  1 1 1 1 1 1  1 1 1 1 1 1 | -49 -8 43 22 4  1 1 1 1 1 1  1 1 1 1 1 1  1 1 1 1 1 1 |
| 7 | 1 1 1 1 1 1  1 1 1 1 1 1  1 1 1 1 1 1  1 строка  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 |
| 8 | 1 1 1 1 1 1  1 1 1 1 1 1  1 1 1 1 1 1  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  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 |
| 9 | 1 1 1 1 1 1  1 1 1 1 1 1  1 1 1 1 1 1  1 строка 1 столбец  нижняя граница 1,  верхняя -5 | Сообщение об ошибке | Нижняя граница не может быть больше верхней |

Добавление в двумерный массив (с клавиатуры):

|  |  |  |  |
| --- | --- | --- | --- |
| **Тест** | **Входные данные** | **Ожидаемый результат** | **Реальный результат** |
| 1 | 1 1 1 1 1 1  1 1 1 1 1 1  1 1 1 1 1 1  2 строка  30 0 5 28 -37 | 1 1 1 1 1 1  23 43 -95 4 4  1 1 1 1 1 1  1 1 1 1 1 1 | 1 1 1 1 1 1  23 43 -95 4 4  1 1 1 1 1 1  1 1 1 1 1 1 |
| 2 | 1 1 1 1 1 1  1 1 1 1 1 1  1 1 1 1 1 1  0 строка | Сообщение об ошибке | Неверный ввод |
| 3 | 1 1 1 1 1 1  1 1 1 1 1 1  1 1 1 1 1 1  -3 строка | Сообщение об ошибке | Неверный ввод |
| 4 | 1 1 1 1 1 1  1 1 1 1 1 1  1 1 1 1 1 1  1,34 строка | Сообщение об ошибке | Неверный ввод |
| 5 | 1 1 1 1 1 1  1 1 1 1 1 1  1 1 1 1 1 1  Q строка | Сообщение об ошибке | Неверный ввод |
| 6 | 1 1 1 1 1 1  1 1 1 1 1 1  1 1 1 1 1 1  1 строка  -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 |

Формирование рваного массива (с помощью ДСЧ):

|  |  |  |  |
| --- | --- | --- | --- |
| **Тест** | **Входные данные** | **Ожидаемый результат** | **Реальный результат** |
| 1 | 1 строка  4 столбца  нижняя граница -100,  верхняя 100 | 4 числа | 0 23 52 -61 |
| 2 | 0 строк | Пустой массив | Пустой массив |
| 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 | 5 строк  4 столбца  3 столбца  8 столбцов  2 столбца  1 столбец  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 |
| 7 | 1 строка  1 столбец  нижняя граница -100,  верхняя 100 | 1 число | 49 |
| 8 | 1111111111111 | Сообщение об ошибке | Неверный ввод |
| 9 | 100 | Сообщение об ошибке | Неверный ввод |
| 10 | -100 | -100 | -100 |
| 11 | 0 | 0 | 0 |

Формирование рваного массива (с клавиатуры):

|  |  |  |  |
| --- | --- | --- | --- |
| **Тест** | **Входные данные** | **Ожидаемый результат** | **Реальный результат** |
| 1 | 5 строк  6 столбцов -1 2 3 4 5 6  1 столбец 4  2 столбца 5  8 столбцов 9 9 9 9 9 9 9 9 9  4 столбца 49 -24 -3 33 | 1 2 3 4 5 6  4  5  9 9 9 9 9 9 9 9 9  49 -24 -3 33 | 1 2 3 4 5 6  4  5  9 9 9 9 9 9 9 9 9  49 -24 -3 33 |
| 2 | 0 строк | Пустой массив | Пустой массив |
| 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  9 столбцов 9 9 9 9 9 9 9 9 9 | 9 9 9 9 9 9 9 9 9 | 9 9 9 9 9 9 9 9 9 |
| 7 | 1 строка  1 столбец 1 | 1 | 1 |
| 8 | 1111111111111 | Сообщение об ошибке | Неверный ввод |
| 9 | 100 | Сообщение об ошибке | Неверный ввод |

Добавление в рваный массив (с помощью ДСЧ):

|  |  |  |  |
| --- | --- | --- | --- |
| **Тест** | **Входные данные** | **Ожидаемый результат** | **Реальный результат** |
| 1 | -90 0 27 54 -67  23 43 95 4 4  2 строка 5 столбцов  нижняя граница -100,  верхняя 100 | -90 0 27 54 -67  1 49 32 33 4  23 43 95 4 4 | -90 0 27 54 -67  1 49 32 33 4  23 43 95 4 4 |
| 2 | -90 0 27 54 -67  23 43 95 4 4  0 строка 5 столбцов  нижняя граница -100,  верхняя 100 | 1 49 32 33 4  -90 0 27 54 -67  23 43 95 4 4 | 1 49 32 33 4  -90 0 27 54 -67  23 43 95 4 4 |
| 3 | -90 0 27 54 -67  23 43 95 4 4  -3 строка | Сообщение об ошибке | Неверный ввод |
| 4 | -90 0 27 54 -67  23 43 95 4 4  1,34 строка | Сообщение об ошибке | Неверный ввод |
| 5 | -90 0 27 54 -67  23 43 95 4 4  Q строка | Сообщение об ошибке | Неверный ввод |
| 6 | -90 0 27 54 -67  23 43 95 4 4  2 строка 0 столбцов  нижняя граница -100,  верхняя 100 | -90 0 27 54 -67  Пустая строка  23 43 95 4 4 | -90 0 27 54 -67  Пустая строка  23 43 95 4 4 |
| 7 | 0 строка 1 столбец  нижняя граница 1,  верхняя 1 | 1 | 1 |
| 8 | -90 0 27 54 -67  23 43 95 4 4  4 строка | Сообщение об ошибке | Неверный ввод |
| 9 | -90 0 27 54 -67  23 43 95 4 4  0 строка | Сообщение об ошибке | Неверный ввод |

Добавление в рваный массив (с клавиатуры):

|  |  |  |  |
| --- | --- | --- | --- |
| **Тест** | **Входные данные** | **Ожидаемый результат** | **Реальный результат** |
| 1 | -90 0 27 54 -67  23 43 95 4 4  2 строка 5 столбцов  1 49 32 33 4 | -90 0 27 54 -67  1 49 32 33 4  23 43 95 4 4 | -90 0 27 54 -67  1 49 32 33 4  23 43 95 4 4 |
| 2 | -90 0 27 54 -67  23 43 95 4 4  1 строка 5 столбцов  1 49 32 33 4 | 1 49 32 33 4  -90 0 27 54 -67  23 43 95 4 4 | 1 49 32 33 4  -90 0 27 54 -67  23 43 95 4 4 |
| 3 | -90 0 27 54 -67  23 43 95 4 4  -3 строка | Сообщение об ошибке | Неверный ввод |
| 4 | -90 0 27 54 -67  23 43 95 4 4  1,34 строка | Сообщение об ошибке | Неверный ввод |
| 5 | -90 0 27 54 -67  23 43 95 4 4  Q строка | Сообщение об ошибке | Неверный ввод |
| 6 | -90 0 27 54 -67  23 43 95 4 4  2 строка 0 столбцов | -90 0 27 54 -67  Пустая строка  23 43 95 4 4 | -90 0 27 54 -67  Пустая строка  23 43 95 4 4 |
| 7 | 1 строка 1 столбец  нижняя граница 1,  верхняя 1 | 1 | 1 |
| 8 | -90 0 27 54 -67  23 43 95 4 4  0 строка | Сообщение об ошибке | Неверный ввод |
| 9 | -90 0 27 54 -67  23 43 95 4 4  4 строка | Сообщение об ошибке | Неверный ввод |

Печать массива:

|  |  |  |  |
| --- | --- | --- | --- |
| **Тест** | **Входные данные** | **Ожидаемый результат** | **Реальный результат** |
| 1 | пустой массив | Сообщение об ошибке | Массив пустой |
| 2 | -1 53 13 | -1 53 13 | -1 53 13 |

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 | 3 | Perestanovka(int[] massiv, int elementi) | Perestanovka(int[] massiv, int elementi) |
| 4 | 4 | Поиск | Поиск |
| 5 | 5 | static void Sortirovka(int[] massiv, int elementi) | static void Sortirovka(int[] massiv, int elementi) |
| 6 | 6 | Сообщение об ошибке | Неверный ввод |
| 7 | 9 | Main(string[] args) | Main(string[] args) |
| 8 | 0 | Выход из программы | Выход из программы |