|  |
| --- |
| Пермский филиал федерального государственного автономного образовательного учреждения высшего образования  «Национальный исследовательский университет  «Высшая школа экономики»  *Факультет экономики, менеджмента и бизнес-информатики* |
|  |
| Меньшиков Олег Владимирович  *Лабораторная 5*  по направлению подготовки *09.03.04 Программная инженерия*  образовательная программа «Программная инженерия»   |  |  | | --- | --- | |  | Доцент к[афедры информационных технологий в бизнесе](https://perm.hse.ru/bi/infobis/), кандидат технических наук  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  О.Л. Викентьева | |

**Оглавление**

[Постановка задач 3](#_Toc52393374)

[Задача 1 3](#_Toc52393375)

[Задача 2 3](#_Toc52393376)

[Задача 3 3](#_Toc52393377)

[Анализ 3](#_Toc52393378)

[Задача 1 3](#_Toc52393379)

[Задача 2 3](#_Toc52393380)

[Задача 3 4](#_Toc52393381)

[Блок-схемы 5](#_Toc52393382)

[Задача 1 5](#_Toc52393383)

[Задача 2 7](#_Toc52393384)

[Задача 3 8](#_Toc52393385)

[Текст программы 9](#_Toc52393386)

[Задача 1 9](#_Toc52393387)

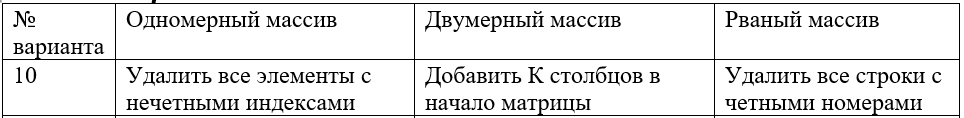
[Задача 2 10](#_Toc52393388)

[Задача 3 11](#_Toc52393389)

**Вариант 10**

# **Анализ**

Постановка задачи:



Введем переменные:

1. i – выбор в меню
2. d – выбор в подменю
3. mas[]– одномерный массив
4. mas1[,]– двумерный массив
5. mas2[][] – рваный массив
6. k– сколько столбцов необходимо добавить
7. size, strings, column – размерности массива

|  |  |  |
| --- | --- | --- |
| Класс входных данных | Входные данные | Ожидаемый результат |
| i – целое число от 1 до 4 | i=3 | Продолжение корректной работы программы |
| i – нецелое число | i=7,6 | «Значение введено неверно, необходимо ввести целое число! Введите заново:» |
| i – строка | i = ‘fghgh’ | «Значение введено неверно, необходимо ввести целое число! Введите заново:» |
| i – целое число, не равное [1;4] | i= 10 | "В меню нет данного варианта!" |
| d – целое число от 1 до 6 | d=3 | Продолжение корректной работы программы |
| d – нецелое число | d=7,6 | «Значение введено неверно, необходимо ввести целое число! Введите заново:» |
| d – строка | d = ‘fghgh’ | «Значение введено неверно, необходимо ввести целое число! Введите заново:» |
| d – целое число, не равное [1;4] | d = 10 | "В меню нет данного варианта!" |
| k – вещественное число | k = 13,4 | «Значение введено неверно, необходимо ввести целое число! Введите заново:» |
| k - строка | k = ‘fghgh’ | «Значение введено неверно, необходимо ввести целое число! Введите заново:» |
| k – отрицательное число | k = -5 | «Число должно положительным. Введите заново:» |
| k - неотрицательное и неположительное число | k =0 | Массив остается прежним |
| k - целое положительное число | k =8 | Добавление 8 столбцов |
| size, strings, column – вещественное число | size= 13,4 | «Значение введено неверно, необходимо ввести целое число! Введите заново:» |
| size, strings, column - строка | size = ‘fghgh’ | «Значение введено неверно, необходимо ввести целое число! Введите заново:» |
| size, strings, column – отрицательное число | size = -5 | «Количество элементов в массиве должно положительным. Введите заново:» |
| size, strings, column - неотрицательное и неположительное число | n=0 | «Количество элементов в массиве должно положительным. Введите заново:» |
| size, strings, column - целое положительное число | size=8 | Выделение памяти в куче для массива с количеством элементов равным size |

# **Проектирование**

|  |  |  |  |
| --- | --- | --- | --- |
| **Название функции** | **Назначение** | **Входные параметры** | **Выходные параметры** |
| ParsingElement | Функция конвертирует строку в тип integer |  | Число типа integer(d) |
| ParsingPositive | Функция конвертирует строку в тип integer, если число неотрицательно |  | Неотрицательное число типа integer(d) |
| CreateRandom | Функция создает одномерный массив с помощью датчика случайных чисел |  | Одномерный массив(mas) |
| CreateRandomDvumer | Функция создает двумерный массив с помощью датчика случайных чисел |  | Двумерный массив(mas) |
| CreateRandomRagged | Функция создает рваный массив с помощью датчика случайных чисел |  | Рваный массив(mas) |
| CreateArr | Функция создает одномерный массив, элементы которого вводятся пользователем |  | Одномерный массив(mas) |
| CreateArrDvumer | Функция создает двумерный массив, элементы которого вводятся пользователем |  | Двумерный массив(mas) |
| CreateArrRagged | Функция создает рваный массив, элементы которого вводятся пользователем |  | Рваный массив(mas) |
| OutputArray | Функция выводит на консоль все элементы одномерного массива | Одномерный массив(mas) |  |
| OutputArray | Функция выводит на консоль все элементы двумерного массива | Двумерный массив(mas) |  |
| OutputArray | Функция выводит на консоль все элементы рваного массива | Рваный массив(mas) |  |
| DeleteOddIndex | Функция удаляет элементы с нечетными индексами из массива, путем глубокого клонирования | Одномерный массив(mas) | Одномерный массив(arr) |
| AddColomns | Функция добавляет столбцы, элементы которых вводятся пользователем, в массив путем глубокого клонирования | Двумерный массив(mas) | Двумерный массив(arr) |
| AddColomnsRandom | Функция добавляет столбцы, элементы которых генерируются датчиком случайных чисел, в массив путем глубокого клонирования | Двумерный массив(mas) | Двумерный массив(arr) |
| DeleteEvenStrings | Функция удаляет четные строки рваного массива | Рваный массив(mas) | Рваный массив(arr) |

# **Листинг**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace лабораторная5

{

class Program

{

//Парсирование

static int ParsingElement()

{

bool ok;

int d;

do

{

string a = Console.ReadLine();

ok = int.TryParse(a, out d);

if (!ok) Console.WriteLine("Необходимо ввести ЦЕЛОЕ число! Введите снова: ");

}

while (!ok);

return d;

}

static int ParsingPositive()

{

bool ok;

int d;

do

{

string a = Console.ReadLine();

ok = int.TryParse(a, out d);

if (!ok) Console.WriteLine("Необходимо ввести ЦЕЛОЕ число! Введите снова: ");

else if (d < 0)

{

Console.WriteLine("Необходимо ввести ПОЛОЖИТЕЛЬНОЕ число! Введите снова: ");

ok = false;

}

}

while (!ok);

return d;

}

//Создание массивов рандомом

static int[] CreateRandom()

{

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

int size = ParsingPositive();

int[] mas;

if (size == 0) mas = null;

else

{

mas = new int[size];

Random rnd = new Random();

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

{

mas[i] = rnd.Next(100);

}

}

return mas;

}

static int[,] CreateRandomDvumer()

{

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

int strings = ParsingPositive();

Console.WriteLine("Введите количество столбцов: ");

int colomns = ParsingPositive();

int[,] mas;

if (strings \* colomns == 0) mas = null;

else {

mas = new int[strings, colomns];

Random rand = new Random();

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

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

{

mas[i, j] = rand.Next(100);

}

}

return mas;

}

static int[][] CreateRandomRagged()

{

Console.WriteLine("Введите количество строк в рваном массиве: ");

int strings = ParsingPositive();

int[][] mas;

if (strings == 0) mas = null;

else

{

mas = new int[strings][];

Random random = new Random();

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

{

Console.WriteLine($"Введите количество чисел в {i + 1} строке: ");

int colomn = ParsingPositive();

if (colomn == 0) mas[i] = null;

else

{

mas[i] = new int[colomn];

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

{

mas[i][j] = random.Next(100);

}

}

}

Console.WriteLine();

}

return mas;

}

//Создание массива вручную

static int[] CreateArr()

{

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

int size = ParsingPositive();

int[] mas;

if (size == 0) mas = null;

else

{

mas = new int[size];

Console.WriteLine("Введите элеметны массива:");

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

mas[i] = int.Parse(Console.ReadLine());

}

return mas;

}

static int[,] CreateArrDvumer()

{

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

int strings = ParsingPositive();

Console.WriteLine("Введите количество столбцов: ");

int colomns = ParsingPositive();

int[,] mas;

if (strings \* colomns == 0) mas = null;

else

{

mas = new int[strings, colomns];

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

{

Console.WriteLine($"Введите числа из {i + 1} строки: ");

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

{

mas[i, j] = ParsingElement();

}

Console.WriteLine();

}

}

return mas;

}

static int[][] CreateArrRagged()

{

Console.WriteLine("Введите количество строк в рваном массиве: ");

int strings = ParsingPositive();

int[][] mas;

if (strings == 0) mas = null;

else

{

mas = new int[strings][];

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

{

Console.WriteLine($"Введите количество чисел в {i + 1} строке: ");

int colomn = ParsingPositive();

if (colomn == 0)

mas[i] = null;

else

{

Console.WriteLine("Введите числа: ");

mas[i] = new int[colomn];

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

{

mas[i][j] = ParsingElement();

}

}

}

Console.WriteLine();

}

return mas;

}

//Вывод массива на экран

static void OutputArray(int[] mas)

{

Console.WriteLine("Текущий массив: ");

if (mas == null) Console.WriteLine("Массив пуст. Убедитесь, что вы заполнили массив.");

else foreach (int x in mas) Console.Write(x + " ");

Console.WriteLine();

}

static void OutputArray(int[,] mas)

{

Console.WriteLine("Текущий массив: ");

if (mas == null) Console.WriteLine("Массив пуст. Убедитесь, что вы заполнили массив.");

else

{

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

{

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

{

Console.Write(mas[i, j] + " ");

}

Console.WriteLine();

}

}

Console.WriteLine();

}

static void OutputArray(int[][] mas)

{

Console.WriteLine("Текущий массив: ");

if (mas == null) Console.WriteLine("Массив пуст. Убедитесь, что вы заполнили массив.");

else

{

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

{

if (mas[i] == null) Console.WriteLine("(Пустой подмассив)");

else

{

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

Console.Write(mas[i][j] + " ");

Console.WriteLine();

}

}

}

Console.WriteLine();

}

//Действия с массивами

static int[] DeleteOddIndex(int[] mas)

{

int[] arr = null;

if (mas != null)

if (mas.Length != 1)

{

arr = new int[mas.Length / 2];

for (int i = 0, j = 1; j < mas.Length; i++, j = j + 2)

{

arr[i] = mas[j];

}

}

return arr;

}

static int[,] AddColumnsRandom(int[,] mas)

{

int[,] arr;

if (mas == null)

{

arr = null;

}

else

{

Console.WriteLine("Введите сколько столбцов необходимо добавить: ");

int k = ParsingPositive();

Random rand = new Random();

arr = new int[mas.GetLength(0), mas.GetLength(1) + k];

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

{

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

{

arr[i, j] = rand.Next(100);

}

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

{

arr[i, j + k] = mas[i, j];

}

}

}

return arr;

}

static int[,] AddColumns(int[,] mas)

{

int[,] arr;

if (mas == null)

{

arr = null;

}

else

{

Console.WriteLine("Введите сколько столбцов необходимо добавить: ");

int k = ParsingPositive();

arr = new int[mas.GetLength(0), mas.GetLength(1) + k];

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

{

Console.WriteLine($"Введите добавленные числа в {i + 1} строку: ");

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

{

arr[i, j] = ParsingElement();

}

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

{

arr[i, j + k] = mas[i, j];

}

}

}

return arr;

}

static int[][] DeleteEvenStrings(int[][] mas)

{

int[][] arr;

if (mas == null)

{

arr = null;

}

else

{

int size = (int)Math.Round(mas.Length / 2.0, MidpointRounding.AwayFromZero);

arr = new int[size][];

for (int i = 0, j = 0; i < size; i++, j = j + 2)

{

arr[i] = mas[j];

}

}

return arr;

}

static void Main(string[] args)

{

bool ok, menu;

int d;

int[] mas = null;

int[,] mas1 = null;

int[][] mas2 = null;

do

{

ok = true;

Console.ForegroundColor = ConsoleColor.Green;

Console.WriteLine(" Главное меню\n1. Работа с одномерным массивом\n2. Работа с двумерным массивом\n3. Работа с рваным массивом\n4. Выход ");

Console.ResetColor();

Console.BackgroundColor = ConsoleColor.DarkRed;

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

Console.ResetColor();

int i = ParsingPositive();

Console.WriteLine();

//главное меню

switch (i)

{

case 1:

menu = true;

do

{

Console.ForegroundColor = ConsoleColor.Blue;

Console.WriteLine(" Одномерный массив\n1. Создать через класс Random\n2. Создать вручную\n3. Удалить элементы с нечетными индексами\n4. Вывести массив на экран\n" +

"5. Вернуться на главное меню\n6. Выход");

Console.ResetColor();

Console.BackgroundColor = ConsoleColor.DarkBlue;

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

Console.ResetColor();

d = ParsingPositive();

//подменю

switch (d)

{

case 1:

mas = CreateRandom();

OutputArray(mas);

break;

case 2:

mas = CreateArr();

OutputArray(mas);

break;

case 3:

mas = DeleteOddIndex(mas);

OutputArray(mas);

break;

case 4:

OutputArray(mas);

break;

case 5:

menu = false;

break;

case 6:

ok = false;

menu = false;

break;

default:

Console.WriteLine("Такого пункта нет в меню");

break;

}

}

while (menu);

Console.WriteLine();

break;

case 2:

menu = true;

do

{

Console.ForegroundColor = ConsoleColor.Blue;

Console.WriteLine(" Двумерный массив\n1. Создать через класс Random\n2. Создать вручную\n3. Добавить К столбиков\n4. Вывести массив на экран\n" +

"5. Вернуться на главное меню\n6. Выход");

Console.ResetColor();

Console.BackgroundColor = ConsoleColor.DarkBlue;

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

Console.ResetColor();

d = ParsingPositive();

//подменю

switch (d)

{

case 1:

mas1 = CreateRandomDvumer();

OutputArray(mas1);

break;

case 2:

mas1 = CreateArrDvumer();

OutputArray(mas1);

break;

case 3:

bool a = true;

do

{

Console.WriteLine("Как ввести числа в массив?\n1. Рандом\n2. Вручную");

int s = ParsingPositive();

switch (s)

{

case 1:

mas1 = AddColumnsRandom(mas1);

a = false;

break;

case 2:

mas1 = AddColumns(mas1);

a = false;

break;

default:

Console.WriteLine("Такого пункта нет в меню!");

break;

}

}

while (a);

OutputArray(mas1);

break;

case 4:

OutputArray(mas1);

break;

case 5:

menu = false;

break;

case 6:

ok = false;

menu = false;

break;

default:

Console.WriteLine("Такого пункта нет в меню");

break;

}

}

while (menu);

Console.WriteLine();

break;

case 3:

menu = true;

do

{

Console.ForegroundColor = ConsoleColor.Blue;

Console.WriteLine(" Рваный массив\n1. Создать через класс Random\n2. Создать вручную\n3. Удалить все строки с четными номерами\n4. Вывести массив на экран\n" +

"5. Вернуться на главное меню\n6. Выход");

Console.ResetColor();

Console.BackgroundColor = ConsoleColor.DarkBlue;

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

Console.ResetColor();

d = ParsingPositive();

//подменю

switch (d)

{

case 1:

mas2 = CreateRandomRagged();

OutputArray(mas2);

break;

case 2:

mas2 = CreateArrRagged();

OutputArray(mas2);

break;

case 3:

mas2 = DeleteEvenStrings(mas2);

OutputArray(mas2);

break;

case 4:

OutputArray(mas2);

break;

case 5:

menu = false;

break;

case 6:

ok = false;

menu = false;

break;

default:

Console.WriteLine("Такого пункта нет в меню");

break;

}

}

while (menu);

Console.WriteLine();

break;

case 4:

ok = false;

break;

default:

Console.WriteLine("В меню нет пункта с данным номером!");

break;

}

}

while (ok);

}

}

}

# **Тесты**

(см. файл excel)