|  |
| --- |
| Пермский филиал федерального государственного автономного образовательного учреждения высшего образования  «Национальный исследовательский университет  «Высшая школа экономики»  *Факультет экономики, менеджмента и бизнес-информатики* |
|  |
| Меньшиков Олег Владимирович  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)

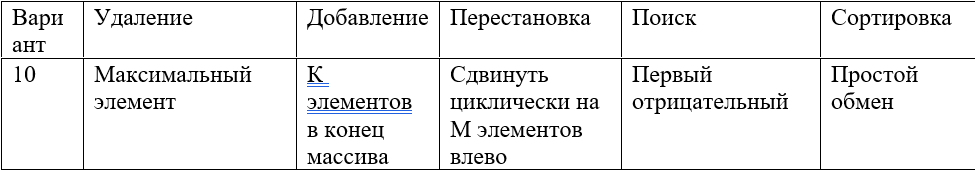
[Задача 2 10](#_Toc52393388)

[Задача 3 11](#_Toc52393389)

**Вариант 10**

# **Анализ**

Задание:



Введем переменные, которые инициализируются с помощью ввода в консоль (ввод пользователем):

1. size – количество элементов массива.
2. i – выбор пункта меню (в switch).
3. mas[j] – элементы массива, которые вводит пользователь.
4. k- количество переменных, которые нужно добавить
5. m- на сколько шагов нужно сдвинуть массив

Таблица анализа входных данных:

|  |  |  |
| --- | --- | --- |
| Класс входных данных | Входные данные | Ожидаемый результат |
| size – вещественное число | size= 13,4 | «Значение введено неверно, необходимо ввести целое число! Введите заново:» |
| size - строка | size = ‘fghgh’ | «Значение введено неверно, необходимо ввести целое число! Введите заново:» |
| size – отрицательное число | size = -5 | «Количество элементов в массиве должно положительным. Введите заново:» |
| size- неотрицательное и неположительное число | n=0 | «Количество элементов в массиве должно положительным. Введите заново:» |
| size - целое положительное число | size=8 | Выделение памяти в куче для массива с количеством элементов равным size |
| i – целое число от 1 до 9 | i=5 | 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;9] | i= 10 | "В меню нет данного варианта!" |
| k, m – вещественное число | k= 13,4 | «Значение введено неверно, необходимо ввести целое число! Введите заново:» |
| k, m - строка | k = ‘fghgh’ | «Значение введено неверно, необходимо ввести целое число! Введите заново:» |
| k, m – отрицательное число | k = -5 | «Число должно быть положительным. Введите заново:» |
| k, m - неотрицательное и неположительное число | k=0 | «Число должно быть положительным. Введите заново:» |
| k, m - целое положительное число | k=8 | Продолжение корректной работы программы |

Таблица анализа элементов массива:

|  |  |  |
| --- | --- | --- |
| Класс входных данных | Входные данные | Ожидаемый результат |
| Нецелые элементы | mas[6]= 13,4 | «Значение введено неверно, необходимо ввести целое число! Введите заново:» |
| Строчные элементы | mas[3]= ‘fghgh’ | «Значение введено неверно, необходимо ввести целое число! Введите заново:» |
| Все элементы целые | 1, 2, -5, 6 | 1)1, 2, -5  2) Продолжение работы  3) Продолжение работы программы  4) «Первый отрицательный элемент: -5, стоящий на 3 месте»  5) -6, 1, 2, 6 |
| Всего один неотрицательный элемент | 1 | 1) В массиве 1 элемент, получим пустой массив  2) Продолжение работы  3) Массив не меняется  4) В массиве нет отрицательных элементов  5) Массив не меняется |
| Все элементы равны | -5, -5, -5 | 1) Все элементы - максимумы, поэтому получится пустой массив  2) Продолжение работы  3) Массив не меняется  4) Минимальный элемент: -5, стоящий на 1 месте  5) Массив не меняется |

# **Проектирование**

1. Parsing



1. ParsingElement



1. Инициализация массива



Конец

1. Удаление максимального элемента





1. Добавление элементов



1. Сдвиг влево



1. Поиск первого отрицательного элемента



1. Сортировка простым обменом





# **Листинг**

using System;

using System.CodeDom.Compiler;

using System.Collections.Generic;

using System.Linq;

using System.Runtime.InteropServices;

using System.Text;

using System.Threading.Tasks;

using System.Xml.Schema;

namespace ConsoleApp3

{

class Program

{

const int left = -200, right = 200;

static int Parsing(string stroka)

{

bool ok;

int n;

do

{

string buf = Console.ReadLine();

ok = int.TryParse(buf, out n);

if (!ok) Console.WriteLine("Значение введено неверно, необходимо ввести целое число! Введите заново: ");

else if (n <= 0)

{

Console.WriteLine($"{stroka}! Введите заново: ");

ok = false;

}

}

while (!ok);

return n;

}

static int ParsingElement()

{

bool ok;

int n;

do

{

string buf = Console.ReadLine();

ok = int.TryParse(buf, out n);

if (!ok) Console.WriteLine("Значение введено неверно! Введите заново: ");

}

while (!ok);

return n;

}

static void Main(string[] args)

{

bool keepGo; //выход из программы

bool goOn; //выход из меню2

bool dalshe; //выход из меню1

int i;

do

{

Console.ForegroundColor = ConsoleColor.Green;

Console.Write("Шаг 1: \n1) Создание рандомного массива \n2) Создание массива вручную \n3) Выход\n\n");

Console.Write("Шаг 2: \n1) " +

"Удаление максимального элемента \n2) " +

"Добавление k элементов в конец массива \n3) Сдвинуть циклически на M элементов влево \n4) " +

"Поиск первого отрицательного элемента \n5) Поиск в отсортированном массиве(простой обмен) \n6) Переход к Шагу 1 \n7) Вывод массива на экран\n8) Выход\n\n");

Console.ResetColor();

Console.BackgroundColor = ConsoleColor.DarkRed;

Console.WriteLine("Шаг 1. Введите кол-во элементов в массиве:");

Console.ResetColor();

int size = Parsing("Количество элементов в массиве должно положительным");

int[] mas = new int[size];

Console.WriteLine();

dalshe = false;

do

{

Console.BackgroundColor = ConsoleColor.DarkRed;

Console.WriteLine("Ваше решение(Шаг 1):");

Console.ResetColor();

keepGo = true;

goOn = true;

i = Parsing("Число должно быть положительным");

switch(i) //ШАГ 1

{

case 1:

#region

{

Random rnd = new Random();

Console.WriteLine("Сформированный массив: ");

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

{

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

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

}

Console.WriteLine();

break;

}

#endregion

case 2:

#region

{

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

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

mas[j] = ParsingElement();

Console.WriteLine("Сформированный массив: ");

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

Console.WriteLine();

break;

}

#endregion

case 3:

goOn = false;

dalshe = false;

keepGo = false;

break;

default:

{

dalshe = true;

Console.WriteLine("В меню нет данного варианта!");

break;

}

}

}

while (dalshe);

Console.WriteLine();

while (goOn)

{

Console.BackgroundColor = ConsoleColor.DarkRed;

Console.WriteLine("Ваше решение(Шаг 2):");

Console.ResetColor();

i = Parsing("Число должно быть положительным");

switch (i) //ШАГ 2

{

case 1:

#region

{

//Поиск максимума

int max = mas[0];

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

if (mas[j] > max) max = mas[j];

//Поиск количества максимумов

int countMax = 0;

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

if (mas[j] == max) countMax++;

//Создание массива с удаленными максимумами

if (mas.Length == 1) Console.WriteLine("В массиве 1 элемент, получим пустой массив");

else if (countMax == mas.Length) Console.WriteLine("Все элементы - максимумы, поэтому получится пустой массив");

else

{

int[] temp = new int[mas.Length - countMax];

int number = 0;

for (int j = 0; j < mas.Length; j++) // Поиск количества максимумов

if (mas[j] != max)

{

temp[number] = mas[j];

number++;

}

Console.WriteLine("В массиве удалены максимумы: ");

foreach (int x in temp)

Console.Write(x + " ");

mas = temp;

Console.WriteLine();

}

}

Console.WriteLine();

break;

#endregion

case 2:

#region

Console.Write("Введите, сколько чисел нужно добавить в массив: ");

int k = Parsing("Число должно быть положительным");

int[] temp2 = new int[mas.Length + k]; //создание нового массива

for (int j = 0; j < mas.Length; j++) //присвоение элементов из прошлого массива

{

temp2[j] = mas[j];

}

Console.WriteLine("Введите новые элементы: ");

for (int j = mas.Length; j < mas.Length + k; j++) // добавление в массив вводимых новых элементов

temp2[j] = ParsingElement();

Console.WriteLine("Массив после добавления элементов: ");

foreach (int x in temp2) Console.Write(x + " ");

Console.WriteLine();

mas = temp2;

Console.WriteLine();

break;

#endregion

case 3:

#region

Console.WriteLine("Введите, на сколько элементов необходимо сдвинуть влево");

int m = Parsing("Число должно быть положительным");

for (int j = 0; j < m; j++) // сдвиг массива на j элементов влево

{

int s = mas[0];

for (int t = 0; t < mas.Length - 1; t++) //сдвиг элементов на один шаг влево

mas[t] = mas[t + 1];

mas[mas.Length - 1] = s;

}

Console.WriteLine($"Массив, сдвинутый на {m} элементов влево: ");

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

Console.WriteLine();

Console.WriteLine();

break;

#endregion

case 4:

#region

int firstOtr = 0;

int indexOtr = 0;

for (int j = 0; j < size; j++) //поиск первого отрицательного элемента и его индекса

if (mas[j] < 0)

{

firstOtr = mas[j];

indexOtr = j + 1;

break;

}

if (firstOtr == 0) Console.WriteLine("В массиве нет отрицательных элементов");

else Console.WriteLine($"Первый отрицательный элемент в массиве: {firstOtr}, стоящий на {indexOtr} месте");

Console.WriteLine();

break;

#endregion

case 5:

#region

Console.WriteLine("Оставить массив отсортированным?\n1. Да\n2. Нет");

int choiceMenu = Parsing("Число должно быть положительным");

switch(choiceMenu)

{

case 1:

#region

int l;

for (i = 1; i < mas.Length; i++) //сортировка массива по возрастанию простым обменом

for (l = mas.Length - 1; l >= i; l--)

if (mas[l] < mas[l - 1])

{

int temp = mas[l];

mas[l] = mas[l - 1];

mas[l - 1] = temp;

}

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

Console.WriteLine();

Console.WriteLine();

Console.WriteLine("Введите число для поиска");

int numberForFind = ParsingElement();

int left = 0, right = size - 1, sred;

do

{

sred = (left + right) / 2;//средний элемент

if (mas[sred] < numberForFind) left = sred + 1;//перенести левую границу

else right = sred;//перенести правую границу

} while (left != right);

if (mas[left] == numberForFind) Console.WriteLine("Номер Элемента {0} равен {1}", numberForFind, left + 1);

else Console.WriteLine("Элемент не найден");

break;

#endregion

case 2:

#region

int[] sort = new int[mas.Length];

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

sort[j] = mas[j];

int cikle;

for (i = 1; i < mas.Length; i++) //сортировка массива по возрастанию простым обменом

for (cikle = mas.Length - 1; cikle >= i; cikle--)

if (mas[cikle] < mas[cikle - 1])

{

int temp = mas[cikle];

mas[cikle] = mas[cikle - 1];

mas[cikle - 1] = temp;

}

Console.WriteLine();

Console.WriteLine();

Console.WriteLine("Введите число для поиска");

int numberFind = ParsingElement();

int left1 = 0, right2 = size - 1, sred1;

do

{

sred1 = (left1 + right2) / 2;//средний элемент

if (mas[sred1] < numberFind) left1 = sred1 + 1;//перенести левую границу

else right2 = sred1;//перенести правую границу

} while (left1 != right2);

if (mas[left1] == numberFind) Console.WriteLine("Элемент {0} существует в данном массиве", numberFind);

else Console.WriteLine("Элемент не найден");

break;

#endregion

default:

Console.WriteLine("В меню нет такого пункта!");

break;

}

break;

#endregion

case 6:

#region

goOn = false; //переход к шагу1

Console.WriteLine();

break;

#endregion

case 7:

Console.WriteLine("Текущий массив: ");

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

Console.WriteLine();

break;

case 8:

#region

keepGo = false; //заврешение работы

goOn = false; //с программой

Console.WriteLine();

break;

#endregion

default:

Console.WriteLine("В меню нет данного варианта!");

break;

}

}

} while (keepGo);

}

}

}

# **ЧЯ**

(см. файл Excel)