Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное   
учреждение высшего образования   
«РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНТ Ф.В. УТКИНА»

Рязанский станкостроительный колледж РГРТУ

ОТЧЁТ

О ПРОХОЖДЕНИИ УЧЕБНОЙ ПРАКТИКИ

Студент: Филяк Илья Игоревич

Специальность: 09.02.07 Информационные системы и программирование

Группа: ИСП-31

Место практики: ООО «Квантрон Групп»

Начало практики: «05» апреля 2021 г.

Окончание практики: «18» апреля 2021 г.

Подпись студента \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель практики от образовательной организации

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.Н. Юдаев

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2021 г.

**Содержание**

1. Паспорт модуля TaskMethods

2. Тестовые наборы

2.1 Ввести двузначное число. Определить: оканчивается ли оно чётной цифрой

2.2 Ввести три целых числа. Найти сумму тех чисел, которые больше пяти

2.3 Дан массив. Требуется найти максимальное значение элементов массива и отклонение от максимального для каждого из чисел

2.4 Дана матрица A(N, M). Строку, содержащей максимальный элемент, поменять местами со строкой, содержащей минимальный элемент

3. Листинг программы и модулей с комментариями

3.1 Листинг кода главной формы

3.2 Листинг кода формы с заданием №1

3.3 Листинг кода формы с заданием №2

3.4 Листинг кода формы с заданием №3

3.5 Листинг кода формы с заданием №4

3.6 Листинг кода модуля TaskMethods

**1. Паспорт модуля TaskMethods**

Разработчик студент группы ИСП-31 Филяк Илья

**Спецификация модуля**

Ввести двузначное число. Определить: оканчивается ли оно чётной цифрой.

public bool EvenLastCifra(int number)

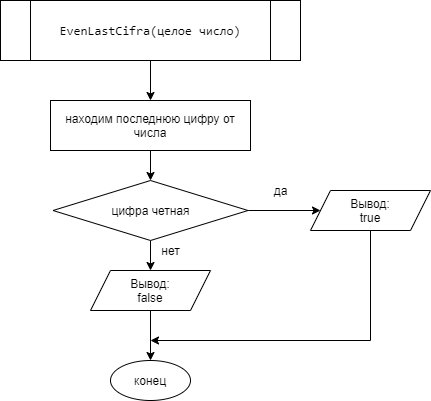
**Входные параметры:**

int number – параметр, устанавливающий целое число.

**Выходные параметры:**

Возвращаем результат выполнения функции.

**Алгоритм** public bool EvenLastCifra(int number)



**Спецификация модуля**

Ввести три целых числа. Найти сумму тех чисел, которые больше пяти.

public int SummThreeNumbers(int[] numbers)

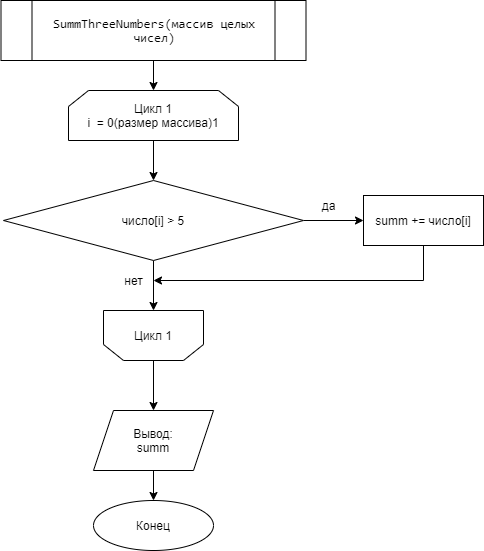
**Входные параметры:**

int[] numbers – параметр, устанавливающий массив целых чисел.

**Выходные параметры:**

Возвращаем результат выполнения функции.

**Алгоритм** public int SummThreeNumbers(int[] numbers)



**Спецификация модуля**

Дан массив. Требуется найти максимальное значение элнментов массива и отклонение от максимального для каждого из чисел.

public void RejectionMaxMass(int[] mass, out int maxNumber, out int[] rejections)

**Входные параметры:**

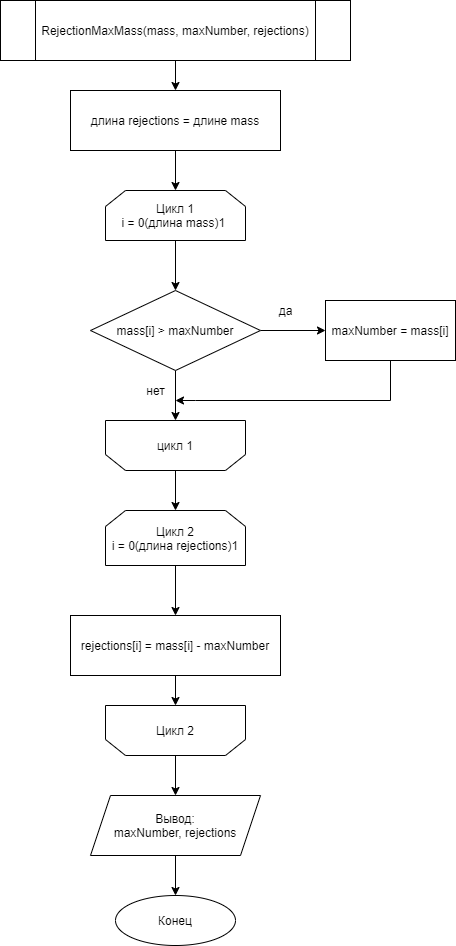
int[] mass – параметр, устанавливающий массив целых чисел.

**Выходные параметры:**

out int maxNumber – максимальное число в массиве.

out int[] rejections – отклонение от максимального для каждого из чисел массива.

**Алгоритм** public void RejectionMaxMass(int[] mass, out int maxNumber, out int[] rejections)



**Спецификация модуля**

Дана матрица A(N, M). Строку, содержащей максимальный элемент, поменять местами со строкой, содержащей минимальный элемент.

public void ExchangeMaxMinRows(double[,] table, out double[,] changedTable)

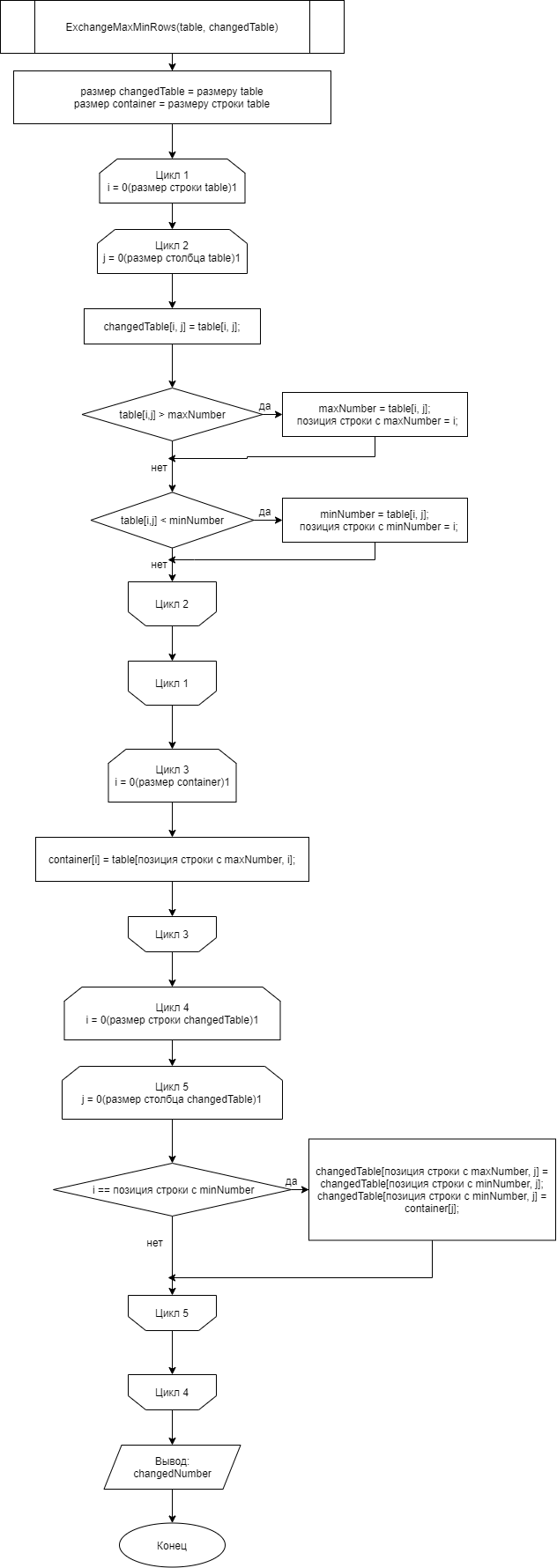
**Входные параметры:**

double[,] table – параметр, устанавливающий матрицу вещественных чисел.

**Выходные параметры:**

out double[,] changedTable – измененная матрица.

**Алгоритм** public void ExchangeMaxMinRows(double[,] table, out double[,] changedTable)



**2. Тестовые наборы**

**2.1 Ввести двузначное число. Определить: оканчивается ли оно чётной цифрой.**

**Тест 1 – проверка последней цифры на четность:**

|  |
| --- |
| **Входные данные** |
| 34 |
| **Выходные данные** |
| true |

**Тест 2 – проверка последней цифры на нечетность:**

|  |
| --- |
| **Входные данные** |
| 31 |
| **Выходные данные** |
| false |

**2.2 Ввести три целых числа. Найти сумму тех чисел, которые больше пяти.**

**Тест 1 – проверка на сложение чисел, которые больше пяти:**

|  |  |  |
| --- | --- | --- |
| **Входные данные** | | |
| 6 | 6 | 6 |
| **Выходные данные** | | |
| 18 | | |

**Тест 2 – проверка на не сложение чисел, которые меньше пяти:**

|  |  |  |
| --- | --- | --- |
| **Входные данные** | | |
| 1 | 2 | 3 |
| **Выходные данные** | | |
| 0 | | |

**2.3 Дан массив. Требуется найти максимальное значение элементов массива и отклонение от максимального для каждого из чисел.**

**Тест 1 – проверка нахождения максимального числа в массиве и отклонение от максимального для каждого элемента в порядке возрастания. Элементы массива - положительные:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Входные данные** | | | | |
| 1 | 2 | 3 | 4 | 5 |
| **Выходные данные** | | | | |
| **5** | | | | |
| 4 | 3 | 2 | 1 | 0 |

**Тест 2 – проверка нахождения максимального числа в массиве и отклонение от максимального для каждого элемента в порядке убывания. Элементы массива – положительные:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Входные данные** | | | | |
| 5 | 4 | 3 | 2 | 1 |
| **Выходные данные** | | | | |
| 5 | | | | |
| 0 | 1 | 2 | 3 | 4 |

**Тест 3 – проверка нахождения максимального числа в массиве и отклонение от максимального для каждого элемента в порядке возрастания. Элементы массива - отрицательные:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Входные данные** | | | | |
| -5 | -4 | -3 | -2 | -1 |
| **Выходные данные** | | | | |
| -1 | | | | |
| 4 | 3 | 2 | 1 | 0 |

**2**.**4 Дана матрица A(N, M). Строку, содержащей максимальный элемент, поменять местами со строкой, содержащей минимальный элемент.**

**Тест 1 – проверка на изменение таблицы по заданию:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Входные данные** | | | |
| 0 | 1 | 2 | 3 |
| -1 | 3 | 6 | 3 |
| 10 | 7 | 5 | 6 |
| **Выходные данные** | | | |
| 0 | 1 | 2 | 3 |
| 10 | 7 | 5 | 6 |
| -1 | 3 | 6 | 3 |

**3. Листинг программы и модулей с комментариями**

**3.1 Листинг кода главной формы**

using System;

using System.Windows.Forms;

namespace practProject1

{

public partial class MainForm : Form

{

public MainForm()

{

InitializeComponent();

}

//События, в которых вызывается окна с заданиями

//Событие, где вызывается окно с заданием №1

private void Task1\_Click(object sender, EventArgs e)

{

Task1 task = new Task1();

task.ShowDialog();

}

//Событие, где вызывается окно с заданием №2

private void Task2\_Click(object sender, EventArgs e)

{

Task2 task = new Task2();

task.ShowDialog();

}

//Событие, где вызывается окно с заданием №3

private void Task3\_Click(object sender, EventArgs e)

{

Task3 task = new Task3();

task.ShowDialog();

}

//Событие, где вызывается окно с заданием №4

private void Task4\_Click(object sender, EventArgs e)

{

Task4 task = new Task4();

task.ShowDialog();

}

//Событие, где вызывается окно "О программе"

private void AboutUs\_Click(object sender, EventArgs e)

{

MessageBox.Show("Программа разработана студентом группы ИСП-31 Филяком Ильёй", "О программе");

}

}

}

**3.2 Листинг кода формы с заданием №1**

using System;

using System.Windows.Forms;

using LibMethods;

namespace practProject1

{

public partial class Task1 : Form

{

public Task1()

{

InitializeComponent();

}

//Событитие, в котором определяется четность или нечетность последней цифры в двузначном числе

private void GetAnswer\_Click(object sender, EventArgs e)

{

bool returned;

TaskMethods task1 = new TaskMethods();

returned = task1.EvenLastCifra((int)number.Value);

if (returned == true)

{

answer.Text = "Последняя цифра является четной";

}

else

{

answer.Text = "Последняя цифра является нечетной";

}

}

}

}

**3.3 Листинг кода формы с заданием №2**

using System;

using System.Windows.Forms;

using LibMethods;

namespace practProject1

{

public partial class Task2 : Form

{

public Task2()

{

InitializeComponent();

}

//Событие, в котором рассчитывается сумма 3 чисел, если они больше 5.

private void GetAnswer\_Click(object sender, EventArgs e)

{

int[] mass = new int[] {(int)firstNumber.Value, (int)secondNumber.Value, (int)thridNumber.Value };

TaskMethods task = new TaskMethods();

answer.Text = task.SummThreeNumbers(mass).ToString();

}

}

}

**3.4 Листинг кода формы с заданием №3**

using System;

using System.Windows.Forms;

using LibMethods;

namespace practProject1

{

public partial class Task3 : Form

{

Random rnd = new Random();

int sizeMass;

int minNumber;

int maxNumber;

int max;

int[] rejections;

public Task3()

{

InitializeComponent();

}

//Событие, в котором генерируется массив, после чего находится максимальное значение и отклонения.

private void GenerateAnswer\_Click(object sender, EventArgs e)

{

originalMass.Clear();

answer.Clear();

TaskMethods task = new TaskMethods();

minNumber = rnd.Next(-100, -1);

maxNumber = rnd.Next(0, 100);

sizeMass = rnd.Next(3, 8);

int[] mass = new int[sizeMass];

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

{

mass[i] = rnd.Next((int)minNumber, (int)maxNumber);

originalMass.Text += mass[i] + " ";

}

task.RejectionMaxMass(mass, out max, out rejections);

answerMaxNumber.Text = max.ToString();

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

{

answer.Text += rejections[i].ToString() + " ";

}

}

}

}

**3.5 Листинг кода формы с заданием №4**

using System;

using System.Windows.Forms;

using LibMethods;

namespace practProject1

{

public partial class Task4 : Form

{

Random rnd = new Random();

public Task4()

{

InitializeComponent();

}

//Событие, в котором генерируестся исходная таблица, котороя сразу записывается в рич текст бокс, после записывается преобразованная таблица.

private void GenerateAnswer\_Click(object sender, EventArgs e)

{

tables.Clear();

int amountRows = rnd.Next(3, 6);

int amountColumns = rnd.Next(3, 6);

TaskMethods task = new TaskMethods();

double[,] matr = new double[amountRows, amountColumns];

double[,] changeMatr;

tables.Text += "Исходная матрица" + "\n";

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

{

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

{

matr[i, j] = rnd.Next(-10, 10);

tables.Text += matr[i, j] + " ";

}

tables.Text += "\n";

}

task.ExchangeMaxMinRows(matr, out changeMatr);

tables.Text += "Измененная матрица" + "\n";

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

{

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

{

tables.Text += changeMatr[i, j] + " ";

}

tables.Text += "\n";

}

}

}

}

**3.6 Листинг кода модуля TaskMethods**

namespace LibMethods

{

public class TaskMethods

{

//Метод задания №1. Определить четная или нечетная ли последняя цифра в двузначном числе.

public bool EvenLastCifra(int number)

{

number %= 10;

if (number % 2 == 0) return true;

else return false;

}

//Метод задания №2. Если 3 числа(или даже больше в моем случае) больше 5, то мы их складываем.

public int SummThreeNumbers(int[] numbers)

{

int summ = 0;

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

{

if(numbers[i] > 5)

{

summ += numbers[i];

}

}

return summ;

}

//Метод задания №3. Определить максимальное число в массиве и найти отклонение в каждом числе от максимального.

public void RejectionMaxMass(int[] mass, out int maxNumber, out int[] rejections)

{

maxNumber = mass[0];

rejections = new int[mass.Length];

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

{

if(mass[i] > maxNumber)

{

maxNumber = mass[i];

}

}

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

{

rejections[i] = maxNumber - mass[i];

}

}

//Метод задания №4. В таблице поменять местами строку, содержащей минимальный элемент таблицы, со строкой, содержащей макисмальный элемент.

public void ExchangeMaxMinRows(double[,] table, out double[,] changedTable)

{

changedTable = new double[table.GetLength(0), table.GetLength(1)];

double[] container = new double[table.GetLength(1)];

int rowMinPosition = 0;

int rowMaxPosition = 0;

double minNumber = table[0,0];

double maxNumber = 0;

//Определяем позицию строк с максимальным и минимальныи элементами

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

{

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

{

changedTable[i, j] = table[i, j];

if(table[i,j] > maxNumber)

{

maxNumber = table[i, j];

rowMaxPosition = i;

}

if(table[i,j] < minNumber)

{

minNumber = table[i, j];

rowMinPosition = i;

}

}

}

//записываем строку, содержащую максимальный элемент.

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

{

container[i] = table[rowMaxPosition, i];

}

//меняем местами строки методом "стаканчика"

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

{

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

{

if( i == rowMinPosition)

{

changedTable[rowMaxPosition, j] = changedTable[rowMinPosition, j];

changedTable[rowMinPosition, j] = container[j];

}

}

}

}

}

}