Министерство образования Республики Беларусь

Оршанский колледж ВГУ имени П.М.Машерова

**Отчет**

по учебной практике по программированию

«Модуль 1.2»

Выполнил учащийся Липинский М.А.

группы 3ПОИС23 30.09.2025г.

Проверил Алейников М.А.

30.09.2025г.

Орша, 2025

**Цель:** разработать Модуль 1.2.

**Задание 1.**

Определить функцию (статический метод) для вычисления наибольшего общего делителя двух целых натуральных чисел (Greatest Common Measure). В основной программе, используя функцию, сократить неотрицательную обыкновенную дробь. Дробь вводится с клавиатуры в виде неотрицательного числителя и положительного знаменателя.

**Задание 2.**

Присваивая последовательным элементам массива случайные значения от 1 до 9, создать массив с минимальным количеством элементов, сумма которых не превышает заданного пользователем числа.

**Задание 3.**

Сформировав квадратную целочисленную матрицу со случайными значениями элементов, упорядочить ее строки по возрастанию сумм их элементов. Пусть элементы принимают значения в диапазоне от –50 до +50.

**Ход работы.**

**Выполнение практических заданий.**

**Задание 1.**

**Выполнение:**

Для выполнения седьмого задания нужно определить функцию (статический метод) для вычисления наибольшего общего делителя двух целых натуральных чисел (Greatest Common Measure). В основной программе, используя функцию, сократить неотрицательную обыкновенную дробь. Дробь вводится с клавиатуры в виде неотрицательного числителя и положительного знаменателя

Код на языке C#, представленный ниже в листинге 1.

Листинг 1. Код программы на языке программирования C#.

using System;

namespace zadanie\_1

{

internal class Program

{

static void Main(string[] args)

{

// Ввод числителя и знаменателя

Console.Write("Введите числитель (неотрицательное число): ");

int chislitel = Convert.ToInt32(Console.ReadLine());

Console.Write("Введите знаменатель (положительное число): ");

int znamenatel = Convert.ToInt32(Console.ReadLine());

// Проверка на корректность ввода

if (znamenatel <= 0)

{

Console.WriteLine("Знаменатель должен быть положительным числом.");

return;

}

// Вычисление НОД для числителя и знаменателя

int nod = NOD(chislitel, znamenatel);

// Сокращение дроби

int SokrChislitel = chislitel / nod;

int SokrZnamenatel = znamenatel / nod;

// Вывод сокращенной дроби

Console.WriteLine($"Сокращенная дробь: {SokrChislitel}/{SokrZnamenatel}");

}

// Метод для вычисления НОД (алгоритм Евклида)

static int NOD(int a, int b)

{

while (b != 0)

{

int temp = b;

b = a % b;

a = temp;

}

return a;

}

}

}

Далее нужно выполнить запуск приложения для демонстрирования его работоспособности, представлено на рисунке 1.

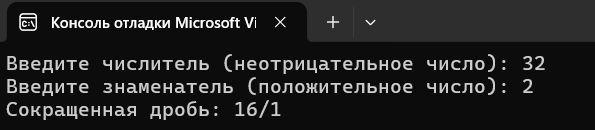


Рисунок 1 – Метод для вычисления NOD

**Задание 2.**

**Выполнение:**

Для выполнения восьмого задания нужно присваивая последовательным элементам массива случайные значения от 1 до 9, создать массив с минимальным количеством элементов, сумма которых не превышает заданного пользователем числа.

Код на языке C#, представленный ниже в листинге 2.

Листинг 2. Код программы на языке программирования C#.

using System;

class Program

{

static Random random = new Random();

// Статический метод для создания массива

public static int[] CreateArray(int maxSum)

{

int sum = 0;

int count = 0;

int[] tempArray = new int[100]; // Максимально возможное количество элементов

while (sum <= maxSum)

{

int value = random.Next(1, 10); // Случайное число от 1 до 9

sum += value;

if (sum > maxSum) break;

tempArray[count++] = value;

}

int[] resultArray = new int[count];

Array.Copy(tempArray, resultArray, count);

return resultArray;

}

static void Main()

{

Console.Write("Введите максимальную сумму: ");

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

int[] array = CreateArray(maxSum);

Console.WriteLine("Сгенерированный массив:");

foreach (int element in array)

{

Console.Write(element + " ");

}

}

}

Далее нужно выполнить запуск приложения для демонстрирования его работоспособности, представлено на рисунке 2.

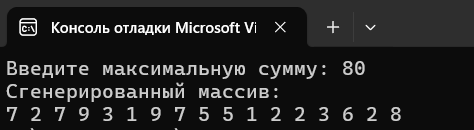


Рисунок 2 – Сумма элементов массива

**Задание 3.**

Для выполнения девятого задания нужно сформировав квадратную целочисленную матрицу со случайными значениями элементов, упорядочить ее строки по возрастанию сумм их элементов. Пусть элементы принимают значения в диапазоне от –50 до +50.

Код на языке C#, представленный ниже в листинге 3.

Листинг 3. Код программы на языке программирования C#.

using System;

class Program

{

static Random random = new Random();

// Метод для создания квадратной матрицы

public static int[,] CreateMatrix(int size)

{

int[,] matrix = new int[size, size];

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

{

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

{

matrix[i, j] = random.Next(-50, 51); // Случайные значения от -50 до 50

}

}

return matrix;

}

// Метод для сортировки строк матрицы по возрастанию сумм их элементов

public static void SortMatrix(int[,] matrix)

{

int size = matrix.GetLength(0);

for (int i = 0; i < size - 1; i++)

{

for (int j = i + 1; j < size; j++)

{

if (RowSum(matrix, i) > RowSum(matrix, j))

{

SwapRows(matrix, i, j);

}

}

}

}

// Метод для вычисления суммы элементов строки

public static int RowSum(int[,] matrix, int row)

{

int sum = 0;

int size = matrix.GetLength(1);

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

{

sum += matrix[row, j];

}

return sum;

}

// Метод для обмена двух строк матрицы

public static void SwapRows(int[,] matrix, int row1, int row2)

{

int size = matrix.GetLength(1);

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

{

int temp = matrix[row1, j];

matrix[row1, j] = matrix[row2, j];

matrix[row2, j] = temp;

}

}

// Метод для вывода матрицы

public static void PrintMatrix(int[,] matrix)

{

int size = matrix.GetLength(0);

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

{

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

{

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

}

Console.WriteLine();

}

}

static void Main()

{

Console.Write("Введите размер квадратной матрицы: ");

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

int[,] matrix = CreateMatrix(size);

Console.WriteLine("Исходная матрица:");

PrintMatrix(matrix);

SortMatrix(matrix);

Console.WriteLine("Матрица после сортировки строк по возрастанию сумм их элементов:");

PrintMatrix(matrix);

}

}

Далее нужно выполнить запуск приложения для демонстрирования его работоспособности, представлено на рисунке 3.

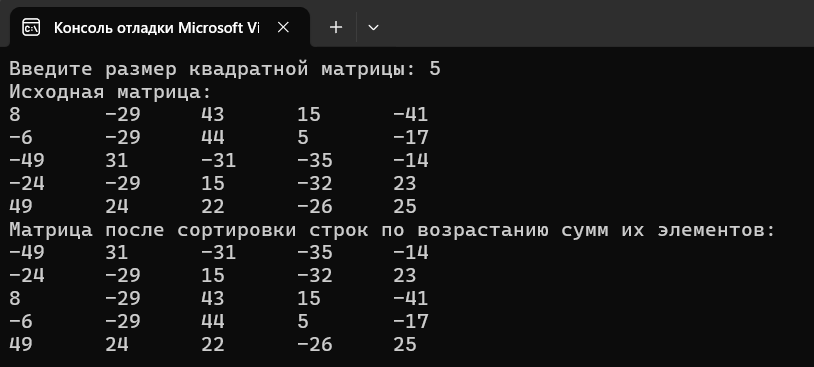


Рисунок 3 – Отсортированная матрица

Все коды хранятся на GitHub в репозитории (https://github.com/MarkUncev/repository1.git).

**Вывод:** в ходе выполнения практического задания был разработан модуль 1.3.