**Занятие № 2**

**Дата выполнения работы:** 20.04.2023

# **Тема работы: «Массивы. Алгоритмы обработки массивов. Рекурсия»**

# **Формируемые умения и навыки:**

**Ход работы**

**Задание 1**

Дан целочисленный массив размера N. Вывести все содержащиеся в

данном массиве нечетные числа в порядке возрастания их индексов, а

также их количество K.

**Листинг программы:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace \_1

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

// Парсинг введенной строки с массивом

string[] inputArray = textBox4.Text.Split(' '); // Предполагается, что числа разделены пробелами

int[] array = new int[inputArray.Length];

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

{

if (int.TryParse(inputArray[i], out int num))

{

array[i] = num;

}

else

{

MessageBox.Show($"Ошибка парсинга элемента с индексом {i + 1}. Введите целое число.", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

}

int count = 0; // Счетчик нечетных чисел

listBox1.Items.Clear(); // Очищаем список вывода

listBox1.Items.Add("Исходный массив:");

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

{

listBox1.Items.Add($"[{i}] = {array[i]}"); // Вывод элементов исходного массива с их индексами

if (array[i] % 2 != 0)

{

count++; // Если элемент нечетный, увеличиваем счетчик

}

}

listBox1.Items.Add($"Количество нечетных чисел: {count}");

listBox1.Items.Add("Нечетные числа в порядке возрастания индексов:");

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

{

if (array[i] % 2 != 0)

{

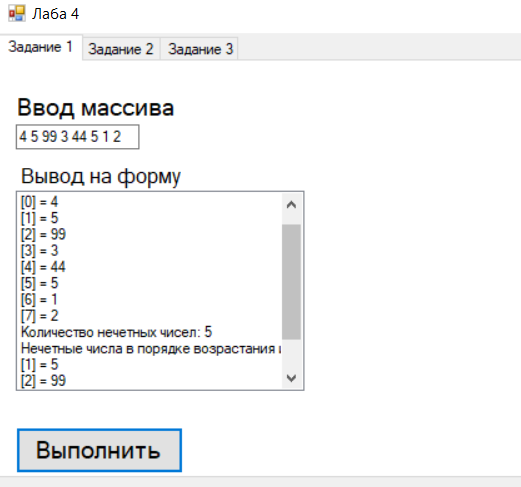
listBox1.Items.Add($"[{i}] = {array[i]}"); // Вывод нечетных чисел с их индексами

}

}

}

**Результат:**



**Вывод:**

**Задание 2**

Дан целочисленный массив размера N, не содержащий одинаковых чисел.

Проверить, образуют ли его элементы арифметическую прогрессию. Если

образуют, то вывести разность прогрессии, если нет — вывести 0. **Листинг программы:**

private void button2\_Click(object sender, EventArgs e)

{

// считываем введенные пользователем числа из текстового поля и разбиваем их на массив

int[] arr = textBox1.Text.Split(' ').Select(int.Parse).ToArray();

int diff = arr[1] - arr[0]; // вычисляем разность первых двух элементов массива

for (int i = 2; i < arr.Length; i++) // проверяем каждую пару последовательных элементов

{

if (arr[i] - arr[i - 1] != diff) // если разность не равна первой, значит прогрессия сломалась

{

diff = 0; // задаем значение 0

break; // выходим из цикла

}

}

// выводим результат в текстовое поле

if (diff == 0)

{

textBox2.Text = "Элементы не образуют арифметическую прогрессию";

}

else

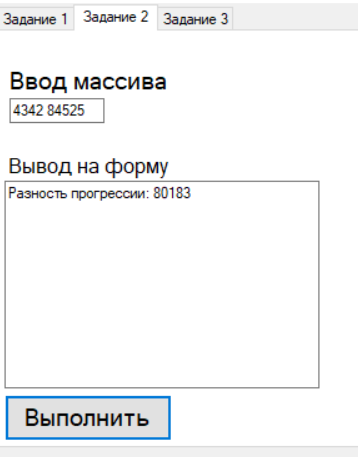
{

textBox2.Text = "Разность прогрессии: " + diff;

}

}

**Результат:**

  
**Задание 3**

Вывести на экран матрицу 4x6. Найти сумму элементов каждой строки.

Вывести результат в виде одномерного массива и найти его максимальный элемент. **Листинг программы:**  
private void button3\_Click(object sender, EventArgs e)

{

// Создаем матрицу 4x6

int[,] matrix = new int[4, 6];

Random rand = new Random();

for (int i = 0; i < 4; i++)

{

for (int j = 0; j < 6; j++)

{

matrix[i, j] = rand.Next(10); // случайные числа от 0 до 9

}

}

// Вычисляем сумму элементов каждой строки

int[] sums = new int[4];

for (int i = 0; i < 4; i++)

{

for (int j = 0; j < 6; j++)

{

sums[i] += matrix[i, j];

}

}

// Записываем результаты в TextBox

textBox3.Text = "Матрица:\r\n";

for (int i = 0; i < 4; i++)

{

for (int j = 0; j < 6; j++)

{

textBox3.Text += matrix[i, j] + " ";

}

textBox3.Text += "\r\n";

}

textBox3.Text += "Суммы строк:\r\n";

for (int i = 0; i < 4; i++)

{

textBox3.Text += sums[i] + " ";

}

textBox3.Text += "\r\n";

// Находим максимальный элемент в массиве сумм

int maxSum = sums[0];

for (int i = 1; i < 4; i++)

{

if (sums[i] > maxSum)

{

maxSum = sums[i];

}

}

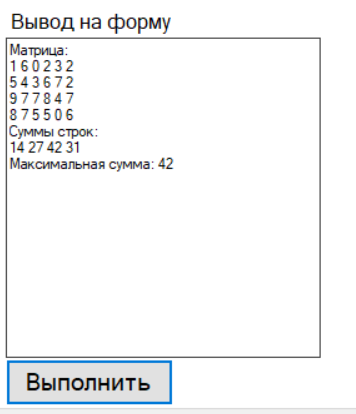
textBox3.Text += "Максимальная сумма: " + maxSum;

}

}

}

**Результат:**

  
**Вывод:**Массивы - это структуры данных, которые позволяют хранить множество значений одного типа. Они используются для удобной и эффективной организации и обработки данных. Алгоритмы обработки массивов позволяют выполнять различные операции над массивами, такие как поиск элементов, сортировка, добавление и удаление элементов, перестановка элементов и другие.

Рекурсия - это метод решения задач, в котором функция вызывает саму себя. Она используется для решения задач, которые могут быть разбиты на подзадачи одного и того же типа.

Комбинируя массивы, алгоритмы обработки и рекурсию, можно решать различные задачи. Например, можно использовать массивы для представления графов и применять алгоритмы поиска путей в графах с помощью рекурсии. Также можно использовать многомерные массивы для хранения матриц и применять к ним алгоритмы линейной алгебры.