**Лабораторная работа № 2**

**«Запись одномерных массивов в СУБД Microsoft Access»**

1. **Постановка задачи**

Дан динамический целочисленный массив. Опишите алгоритм, позволяющий найти и вывести наименьшую нечётную сумму двух соседних элементов массива.

Разработайте метод, который формирует новый массив, из тех элементов исходного массива, значения которых больше найденного числа.

Кроме того, необходимо реализовать алгоритм, позволяющий найти и вывести сумму всех трёхзначных чисел, десятичная запись которых оканчивается на 9, но не на 99, если их нет вывести -1. После отсортировать массив двумя видами сортировок простой вставкой и простым обменом.

Для выполнения задания следует разобрать нижеперечисленные методы:

* Метод ***Zapis\_v\_PDF(int lenght, int[] mas, int count, int[] rezmas***, предназначенный для вывода массивов в PDF-документ;
* Метод ***Zapis\_v\_bloknot(int length, int[] mas, int count, int[] rezmas)***, предназначенный для записи массивов в редактор Блокнот;
* Метод ***Vvod()***, предназначенный для ввода данных;
* Метод ***Vivod()***, предназначенный для вывода данных;
* Метод ***arrayGenerate()***, предназначенный для создания массива, состоящего из случайного набора чисел;
* Метод ***output\_mas()***, предназначенный для вывода массива в таблицу;
* Метод ***set\_mas()***, формирующий новый массив, из тех элементов исходного массива, которые больше наименьшей нечетной суммы соседних элементов;
* Метод ***poisk()***, созданный для поиска пары соседних чисел с наименьшей нечётной суммой;
* Метод ***VivodDGV***(), созданный для вывода значений больших наименьшей суммы с нечётным значением;
* Метод ***InputInt()***, выводит число из ***textbox***;
* Метод ***Sort\_prost***, сортирующая простым методом;
* Метод ***Sort\_obmen***, сортирующая обменом;
* Метод ***Zad\_3***, выполняющая задание, описанное выше;

1. **Разработка проекта**
   1. Разработка графического интерфейса пользователя

Интерфейс пользователя представлена на рисунке 1.

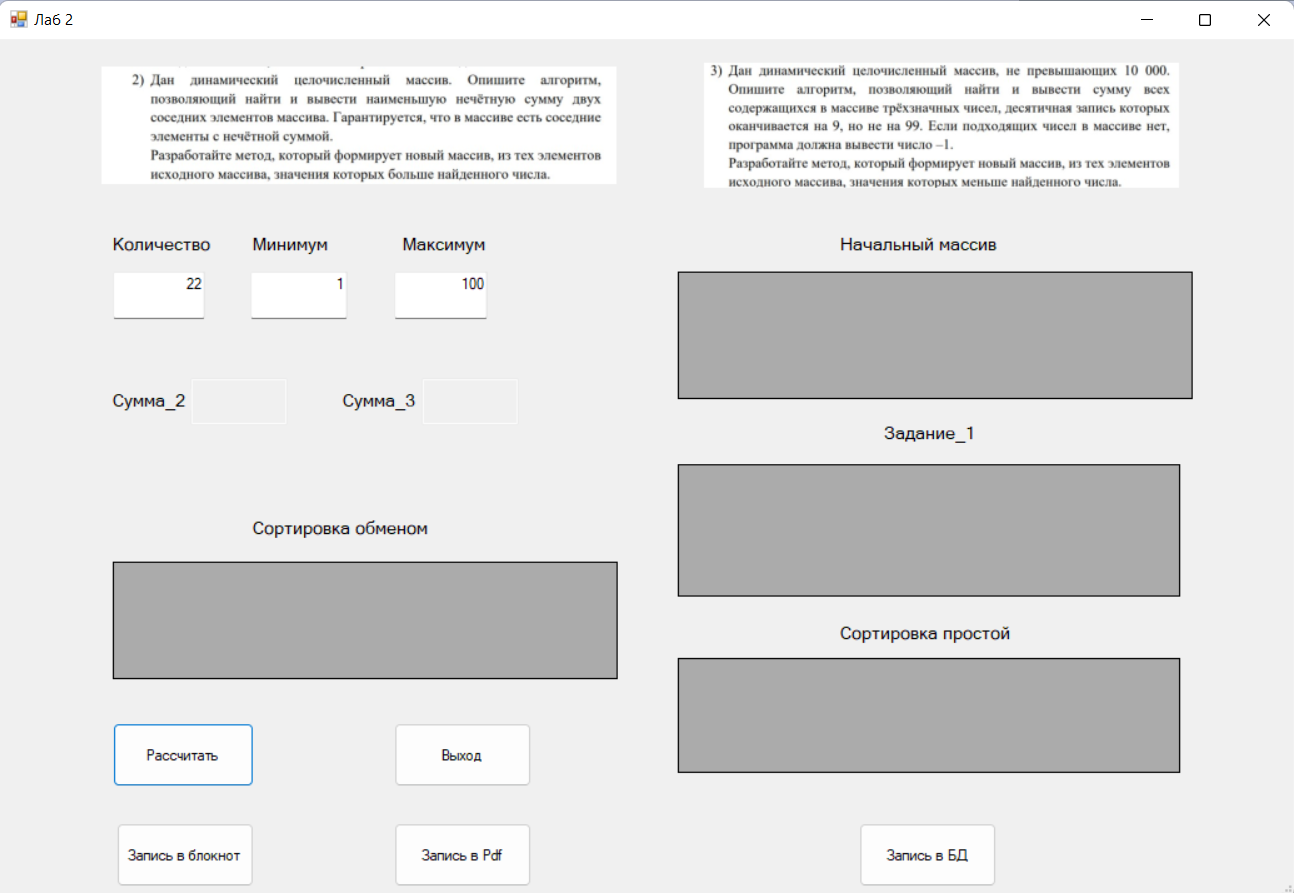
****

Рисунок 1 - Интерфейс пользователя

* 1. Разработка схемы алгоритмов

Блок-схемы событийной части проекта представлен на рисунках 2 - 6.

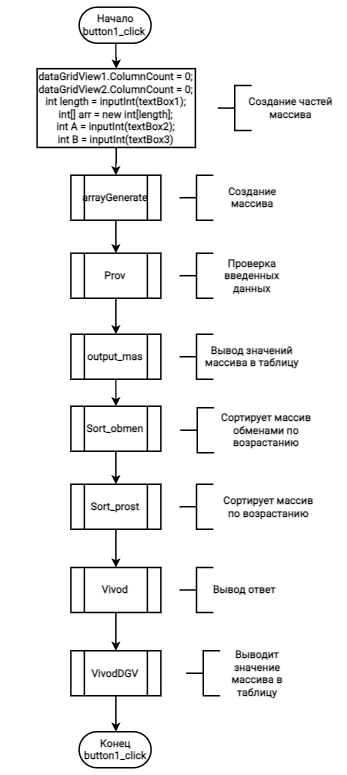


Рисунок 2 - Схема алгоритмов процедур **button1\_Click**

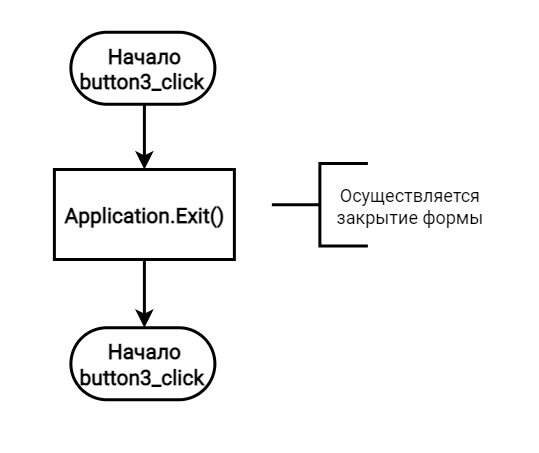
****

Рисунок 3 - Схема алгоритмов процедур **button3\_Click**

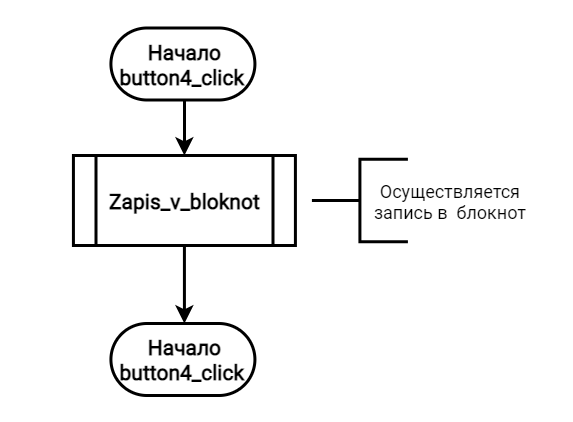


Рисунок 4 - Схема алгоритмов процедур **button4\_Click**

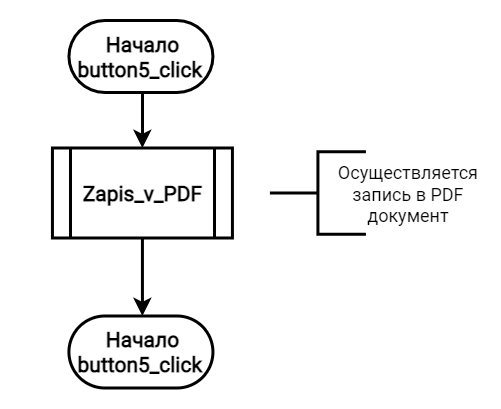
****

Рисунок 5 - Схема алгоритмов процедур **button5\_Click**

* 1. Написание программного кода

Все подпрограммы-методы размещаются в классе (class1).

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 AnP\_proj

{

public partial class Form10 : Form

{

public static int[] arr;

public static int[] mas;

public static int[] rezmas;

public static int length;

public static int count;

public Form10()

{

InitializeComponent();

}

private void textBox3\_KeyPress(object sender, KeyPressEventArgs e)

{

\_3sem.Class1.Prof(textBox3, e);

}

private void textBox2\_KeyPress(object sender, KeyPressEventArgs e)

{

\_3sem.Class1.Prof(textBox2, e);

}

private void textBox1\_KeyPress(object sender, KeyPressEventArgs e)

{

\_3sem.Class1.Prof(textBox1, e);

}

private void button6\_Click(object sender, EventArgs e)

{

\_3sem.Class1.add();

\_3sem.Class1.add\_struct();

\_3sem.Class1.add\_zap(ref arr, length, ref mas,count);

}

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

{

dataGridView1.Rows.Clear();

dataGridView2.Rows.Clear();

dataGridView1.ColumnCount = 0;

dataGridView2.ColumnCount = 0;

//vvod parametrov

length = \_3sem.Class1.inputInt(textBox1); //count

if (length <= 0) return;

//generation and print array

arr = new int[length];

int A = \_3sem.Class1.inputInt(textBox2); //start

int B = \_3sem.Class1.inputInt(textBox3); //stop

if (A > B)

{

MessageBox.Show("Ошибка, верхняя граница не должна превышать нижнюю");

A = \_3sem.Class1.inputInt(textBox2); //start

B = \_3sem.Class1.inputInt(textBox3); //stop

}

\_3sem.Class1.arrayGenerate(ref arr, length, A, B);

//\_3sem.Class1arrayOutput(ref arr, lenght, ref dataGridView1);

\_3sem.Class1.output\_mas(ref dataGridView1, arr);

double suma = \_3sem.Class1.poisk(arr);

count = 0;

\_3sem.Class1.poisk\_razm(arr, suma, ref count);

\_3sem.Class1.Vivod(textBox4, suma);

mas = new int[count];

\_3sem.Class1.VivodDGV(ref arr, suma, dataGridView2, ref mas);

double suma\_2 = \_3sem.Class1.Zad\_3(arr);

\_3sem.Class1.Vivod(textBox5, suma\_2);

\_3sem.Class1.Sort\_prost(ref arr);

\_3sem.Class1.output\_mas(ref dataGridView3, arr);

\_3sem.Class1.Sort\_obmen(ref arr);

\_3sem.Class1.output\_mas(ref dataGridView4, arr);

}

private void button4\_Click(object sender, EventArgs e)

{

\_3sem.Class1.Zapis\_v\_bloknot(ref arr, ref mas);

}

private void button5\_Click(object sender, EventArgs e)

{

\_3sem.Class1.Zapis\_v\_PDF(length,count, arr, mas);

}

private void button3\_Click(object sender, EventArgs e)

{

Application.Exit();

}

}

}

Основной код программы

1. **Результат выполнения проекта**

Результат работы программы представлен на рисунке 6.

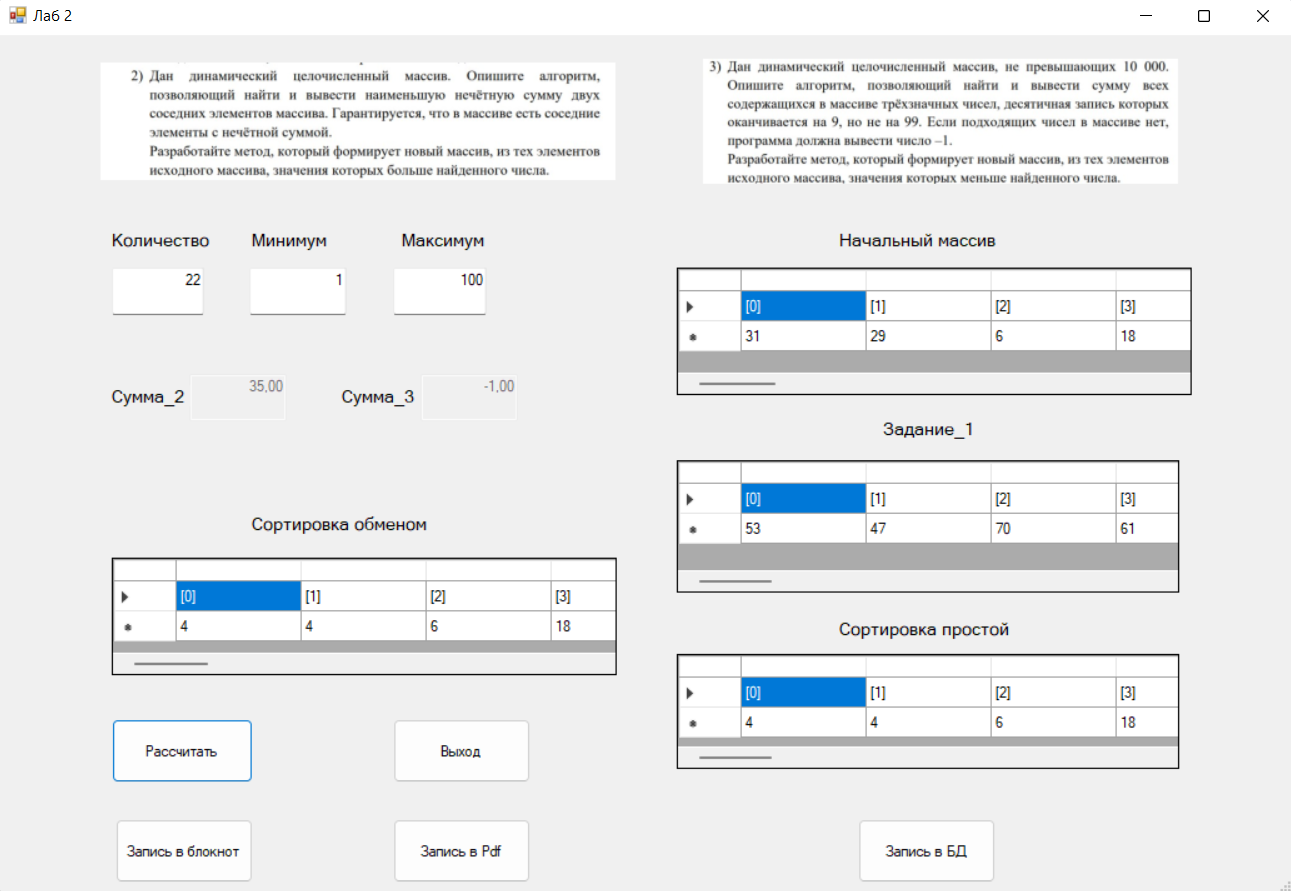


Рисунок 6 - Результат выполнения проекта

Создание Базы данных и структуры Базы данных представлена на рисунках 7-8.

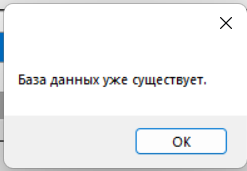
****

Рисунок 7 – Создание Базы данных

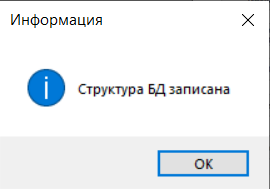
****

Рисунок 8 – Создание структуры Базы данных

Запись массивов в Базу данных представлена на рисунке 9.

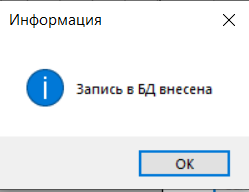
****

Рисунок 9 – Запись массивов в Базу данных.

Интерфейс Базы данных представлен на рисунке 10.

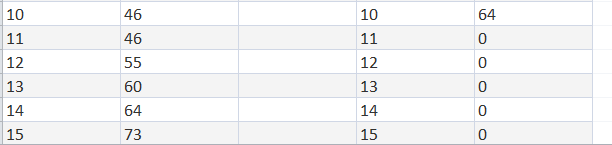
****

Рисунок 10 – Массивы в Базе данных.