ФГБОУ ВО «Сочинский государственный университет»

Инженерно-экологический факультет

Кафедра информационных технологий

**ОТЧЕТ**

о выполнении лабораторной работы №5

по дисциплине **«Алгоритмизация и программирование»**

***Выполнил:***

студент 2 курса 22-ПИ-2

Лукин.М.Е.

«\_17\_»\_\_*Февраля*\_\_2023г

***Проверил:***

проф. каф. инф. тех., д.т.н.

Попов Дмитрий Иванович

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_2023г

Оценка\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Отзыв, замечания\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Сочи 2023

учебные примеры 1-4

Программный код:

static void Main(string[] args)

{

byte[] data = new byte[10];

FileStream fs = new FileStream("FileStreamTest.txt",

FileMode.OpenOrCreate);

if (fs.Length == 0)

{

Console.WriteLine("Writing Data...");

for (short i = 0; i < 10; i++)

data[i] = (byte)i;

fs.Write(data, 0, 10); // Запись данных

}

else

{

fs.Seek(-5, SeekOrigin.End); // Ищем конец

int count = fs.Read(data, 0, 10); // Чтение данных

for (int i = 0; i < count; i++) Console.WriteLine(data[i]);

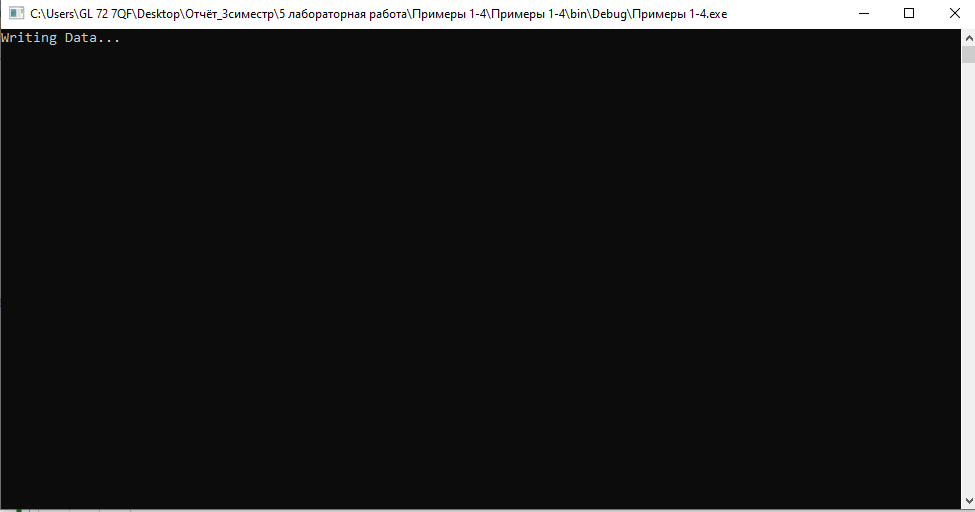
}

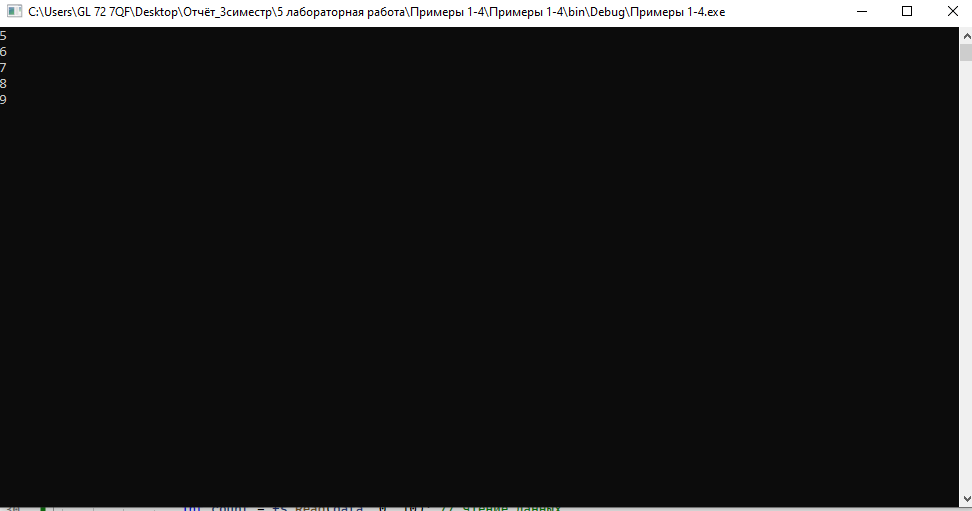
fs.Close();

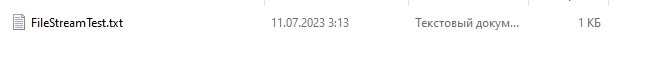
Console.ReadLine();

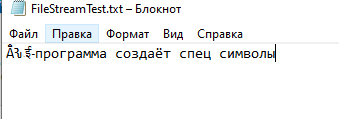
}

Результат выполнения программы:









Программный код:

static void Main(string[] args)

{

FileStream fs = new FileStream("BinaryTest.bin", FileMode.OpenOrCreate);

if (fs.Length == 0)

{

Console.WriteLine("Writing Data...");

BinaryWriter w = new BinaryWriter(fs);

for (short i = 0; i < 10; i++) w.Write(i); // Запись

w.Close();

}

else

{

BinaryReader r = new BinaryReader(fs);

for (int i = 0; i < 10; i++) Console.WriteLine(r.ReadInt16());

r.Close();

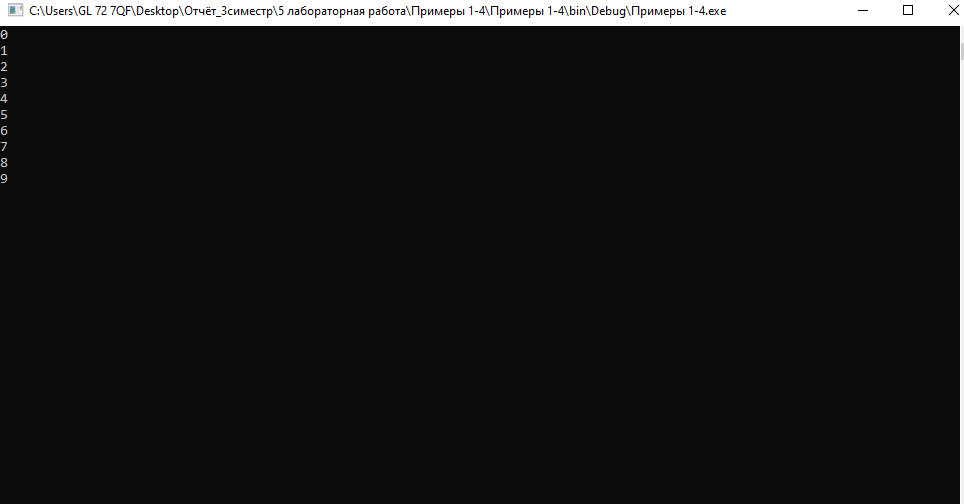
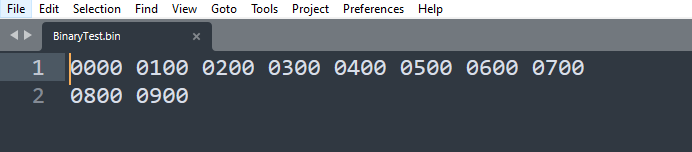
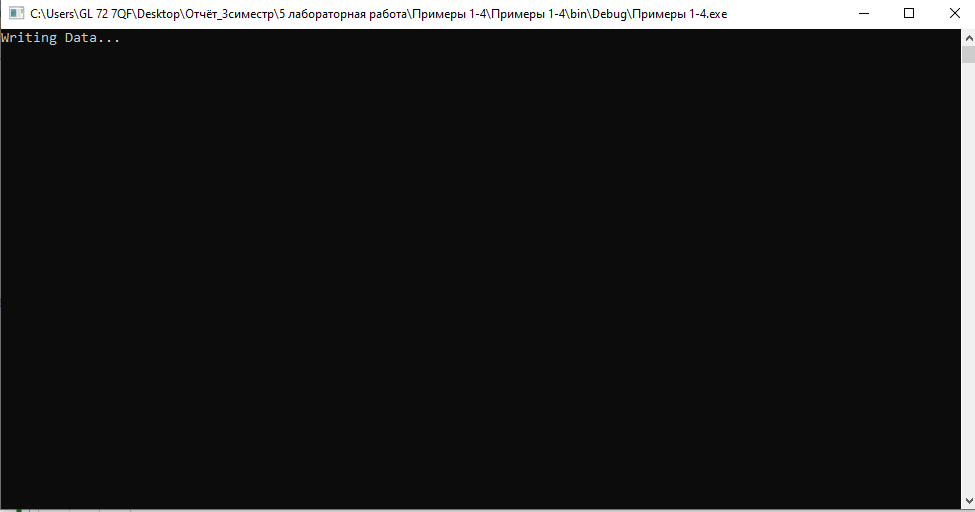
}

fs.Close();

Console.ReadLine();

}

Результат выполнения программы:



Программный код:

static void Main(string[] args)

{

FileStream fs = new FileStream("TextTest.txt", FileMode.OpenOrCreate);

if (fs.Length == 0)

{

Console.WriteLine("Writing Data..."); // Запись данных

StreamWriter sw = new StreamWriter(fs);

sw.Write(100); // Запись

sw.WriteLine(" One Hundred"); // Сто

sw.WriteLine("End of File"); // Конец Файла

sw.Close();

}

else

{

String text; // Строка

StreamReader sr = new StreamReader(fs);

text = sr.ReadLine(); // текст

while (text != null)

{

Console.WriteLine(text); text = sr.ReadLine();

}

sr.Close();

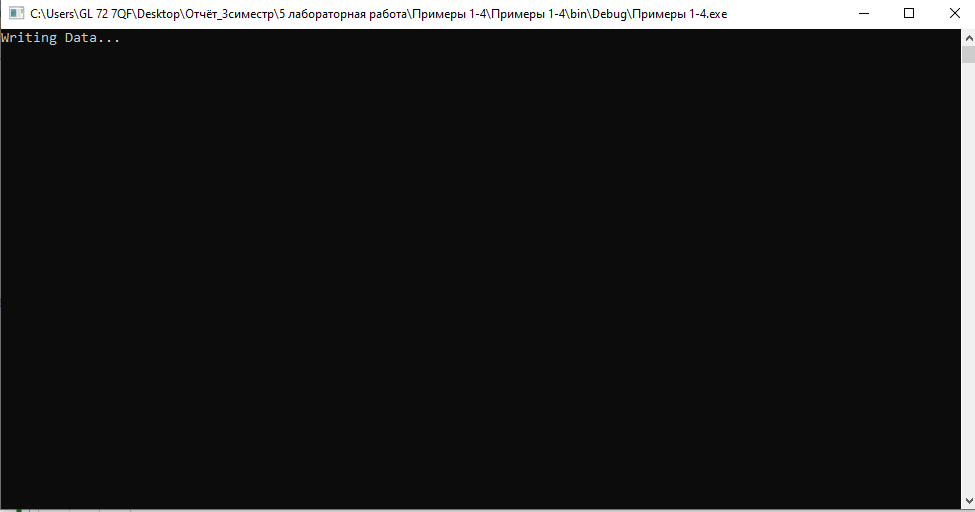
}

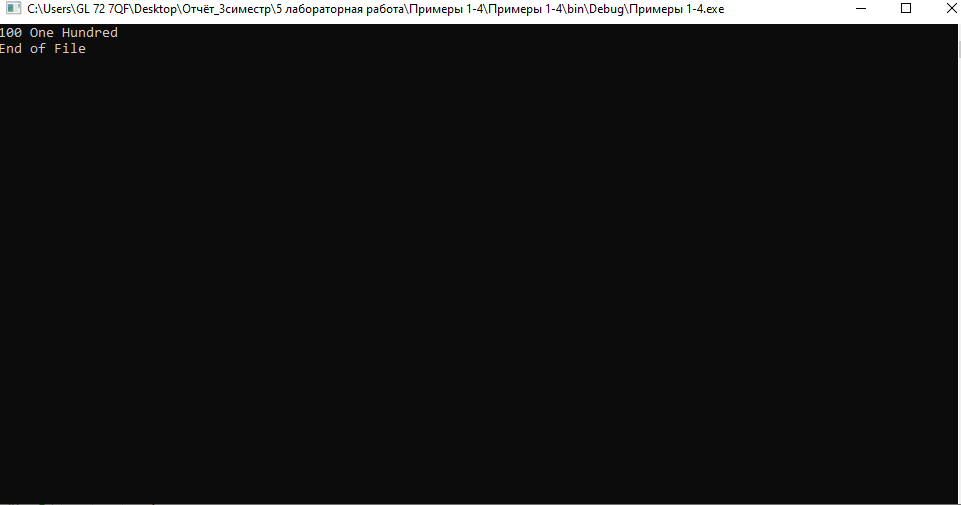
fs.Close();

Console.ReadLine();

}

Результат выполнения программы:





Программный код:

static void Main(string[] args)

{

File.Delete("file2.txt"); // Удалить файл "file2.txt"

StreamWriter sw = System.IO.File.CreateText("file.txt");

sw.Write("Пусть каждый день твой будет светлым, ");

sw.WriteLine("приятным, радостным и щедрым!");

sw.Close();

File.Move("file.txt", "file2.txt"); // Переименование

FileInfo fileInfo = new FileInfo("file2.txt");

Console.WriteLine("File {0} is {1} bytes in length, created on {2}",

fileInfo.FullName, fileInfo.Length, fileInfo.CreationTime);

Console.WriteLine("");

StreamReader sr = fileInfo.OpenText();

String s = sr.ReadLine();

while (s != null)

{

Console.WriteLine(s); s = sr.ReadLine();

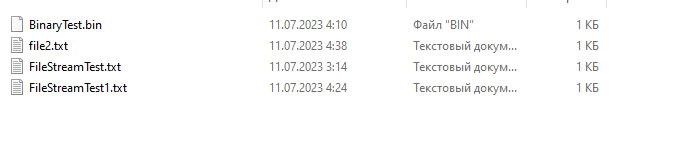
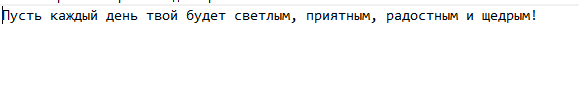
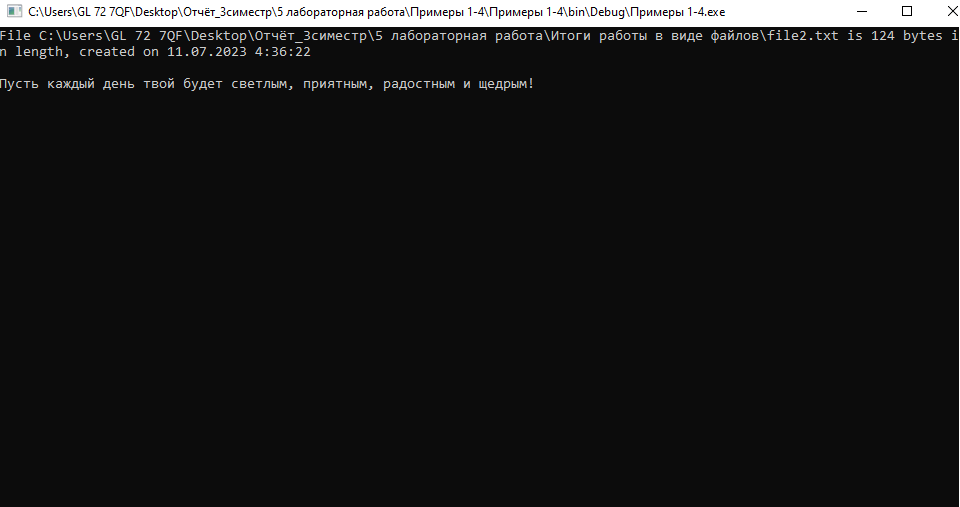
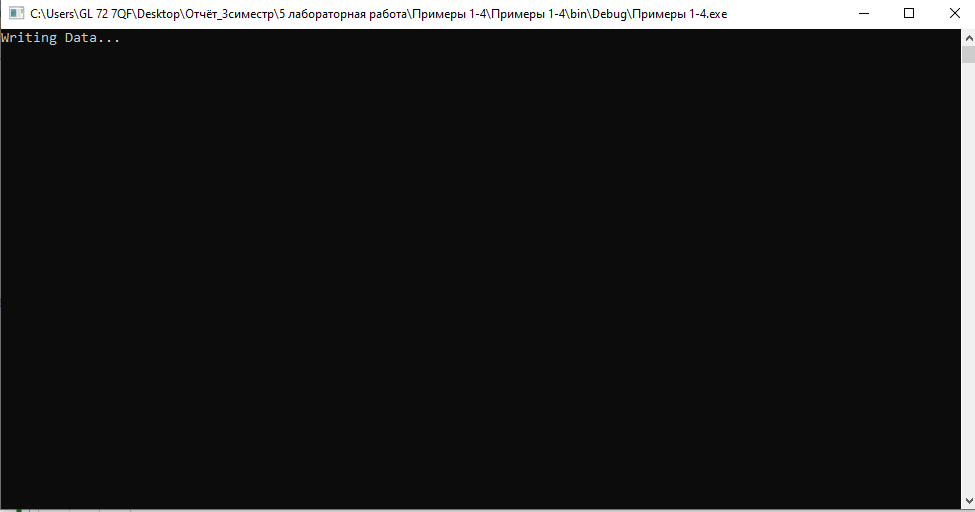
}

sr.Close();

Console.ReadLine();

}

Результат выполнения программы



Программа Frmmain

Программный код:

Окно Form1:

public partial class Frmmain : Form

{

private int openDoc;

private blank frm;

private inform inf;

public Frmmain()

{

InitializeComponent();

mnuSave.Enabled = false;

}

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

{

frm = new blank();

frm.DocName = "Документ" + ++openDoc;

frm.Text = frm.DocName; frm.MdiParent = this;

frm.Show();

}

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

{

this.LayoutMdi(MdiLayout.ArrangeIcons);

}

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

{

this.LayoutMdi(MdiLayout.Cascade);

}

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

{

this.LayoutMdi(MdiLayout.TileHorizontal);

}

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

{

this.LayoutMdi(MdiLayout.TileVertical);

}

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

{

blank frm = (blank) (this.ActiveMdiChild);

frm.Cut();

}

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

{

blank frm = (blank)(this.ActiveMdiChild);

frm.Copy();

}

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

{

blank frm = (blank)(this.ActiveMdiChild);

frm.Paste();

}

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

{

blank frm = (blank)(this.ActiveMdiChild);

frm.Delete();

}

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

{

blank frm = (blank)(this.ActiveMdiChild);

frm.SelectAll();

}

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

{

blank frm = (blank)(this.ActiveMdiChild);

frm.Fon\_Times\_New\_Roman();

}

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

{

blank frm = (blank)(this.ActiveMdiChild);

frm.Color\_Red();

}

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

{

// задание доступных расширений файлов программно

openFileDialog1.Filter = "Text Files (\*.txt)|\*.txt|All Files(\*.\*)|\*.\*";

if (openFileDialog1.ShowDialog() == DialogResult.OK)

{

blank frm = new blank(); frm.Open(openFileDialog1.FileName);

frm.MdiParent = this;

frm.DocName = openFileDialog1.FileName;

frm.Text = frm.DocName;

frm.Show();

}

mnuSave.Enabled = true;

}

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

{

blank frm = (blank)(this.ActiveMdiChild);

frm.Save(frm.DocName);

saveFileDialog1.Filter = "Text Files (\*.txt)|\*.txt|All Files(\*.\*)|\*.\*";

if (saveFileDialog1.ShowDialog() == DialogResult.OK)

{

frm.Save(saveFileDialog1.FileName); frm.MdiParent = this;

frm.DocName = saveFileDialog1.FileName; frm.Text = frm.DocName;

}

frm.IsSaved = true;

}

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

{

saveFileDialog1.Filter = "Text Files (\*.txt)|\*.txt|All Files(\*.\*)|\*.\*";

if (saveFileDialog1.ShowDialog() == DialogResult.OK)

{

blank frm = (blank)(this.ActiveMdiChild);

frm.Save(saveFileDialog1.FileName); frm.MdiParent = this;

frm.DocName = saveFileDialog1.FileName; frm.Text = frm.DocName;

}

mnuSave.Enabled = true;

frm.IsSaved = true;

}

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

{

this.Close();

}

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

{

inf = new inform();

inf.MdiParent = this;

inf.Show();

}

}

Окно blank:

public partial class blank : Form

{

public String DocName;

private String BufferText;

public bool IsSaved = false;

public void Cut() // Вырезание текста

{

this.BufferText = richTextBox1.SelectedText;

richTextBox1.SelectedText = "";

}

public void Copy() // Копирование текста

{

this.BufferText = richTextBox1.SelectedText;

}

public void Paste() // Вставка

{

richTextBox1.SelectedText = this.BufferText;

}

// Выделение всего текста – используем свойство SelectAll элемента управления RichTextBox

public void SelectAll()

{

richTextBox1.SelectAll();

}

public void Delete() // Удаление

{

richTextBox1.SelectedText = "";

this.BufferText = "";

}

public void Color\_Red() // Удаление

{

richTextBox1.ForeColor = Color.Red;

}

public void Fon\_Times\_New\_Roman() // Удаление

{

richTextBox1.SelectionFont = new Font("Times New Roman", 14);

}

public void Open(String OpenFileName)

{

if (OpenFileName == null) { return; }

else

{

StreamReader sr = new StreamReader(OpenFileName);

richTextBox1.Text = sr.ReadToEnd(); sr.Close();

DocName = OpenFileName;

}

}

public void Save(String SaveFileName)

{

if (SaveFileName == null)

{

return;

}

else

{

StreamWriter sw = new StreamWriter(SaveFileName);

sw.WriteLine(richTextBox1.Text); sw.Close(); //Устанавливаем имя документа

DocName = SaveFileName;

}

}

public blank()

{

InitializeComponent();

}

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

{

Cut();

}

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

{

Copy();

}

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

{

Paste();

}

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

{

Delete();

}

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

{

SelectAll();

}

private void richTextBox1\_MouseUp(object sender, MouseEventArgs e)

{

// ассоциируем контекстное меню с текстовым полем

richTextBox1.ContextMenuStrip = contextMenuStrip2;

}

private void blank\_FormClosed(object sender, FormClosedEventArgs e)

{

if (IsSaved == true)

if (MessageBox.Show("Do you want save changes in "

+ this.DocName + "?", "Message", MessageBoxButtons.YesNo,

MessageBoxIcon.Question) == DialogResult.Yes)

{

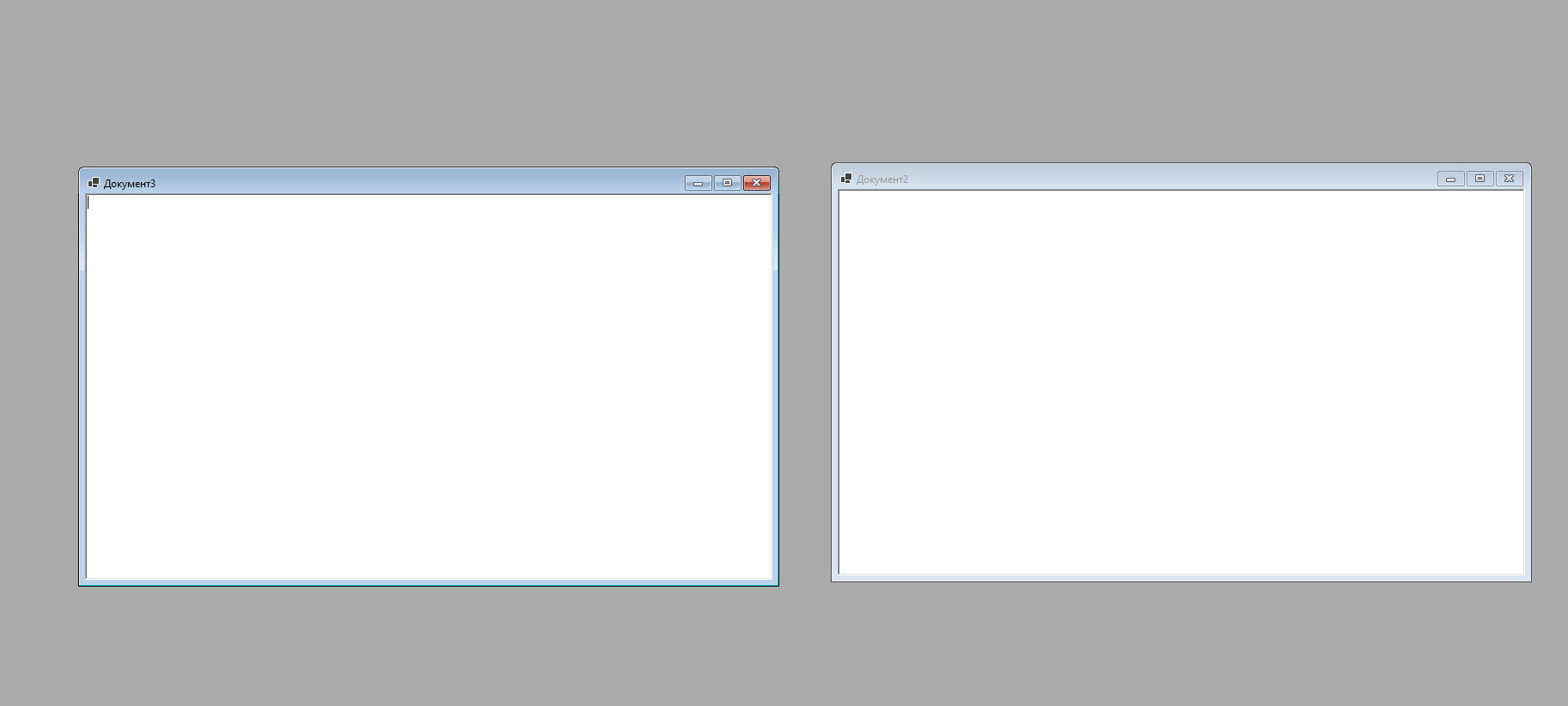
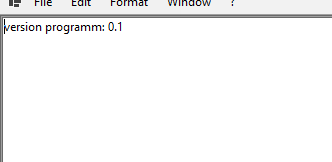
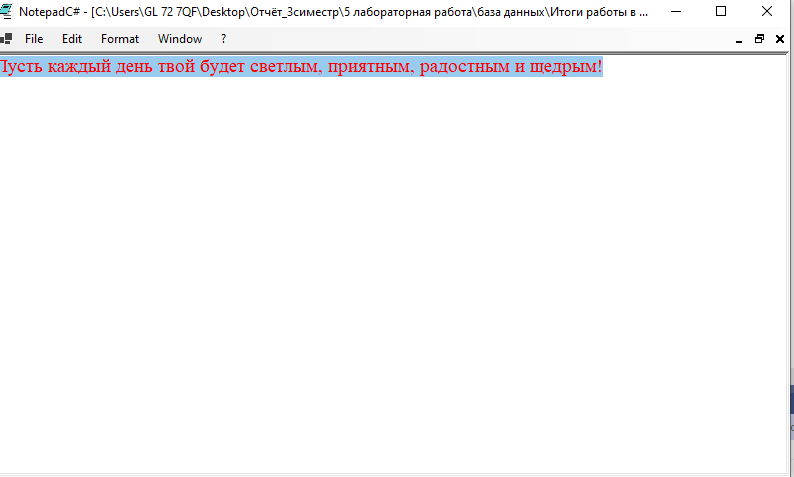
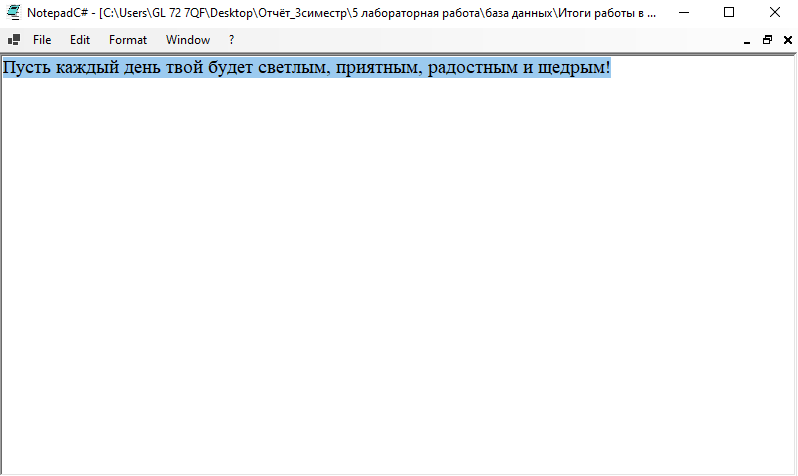
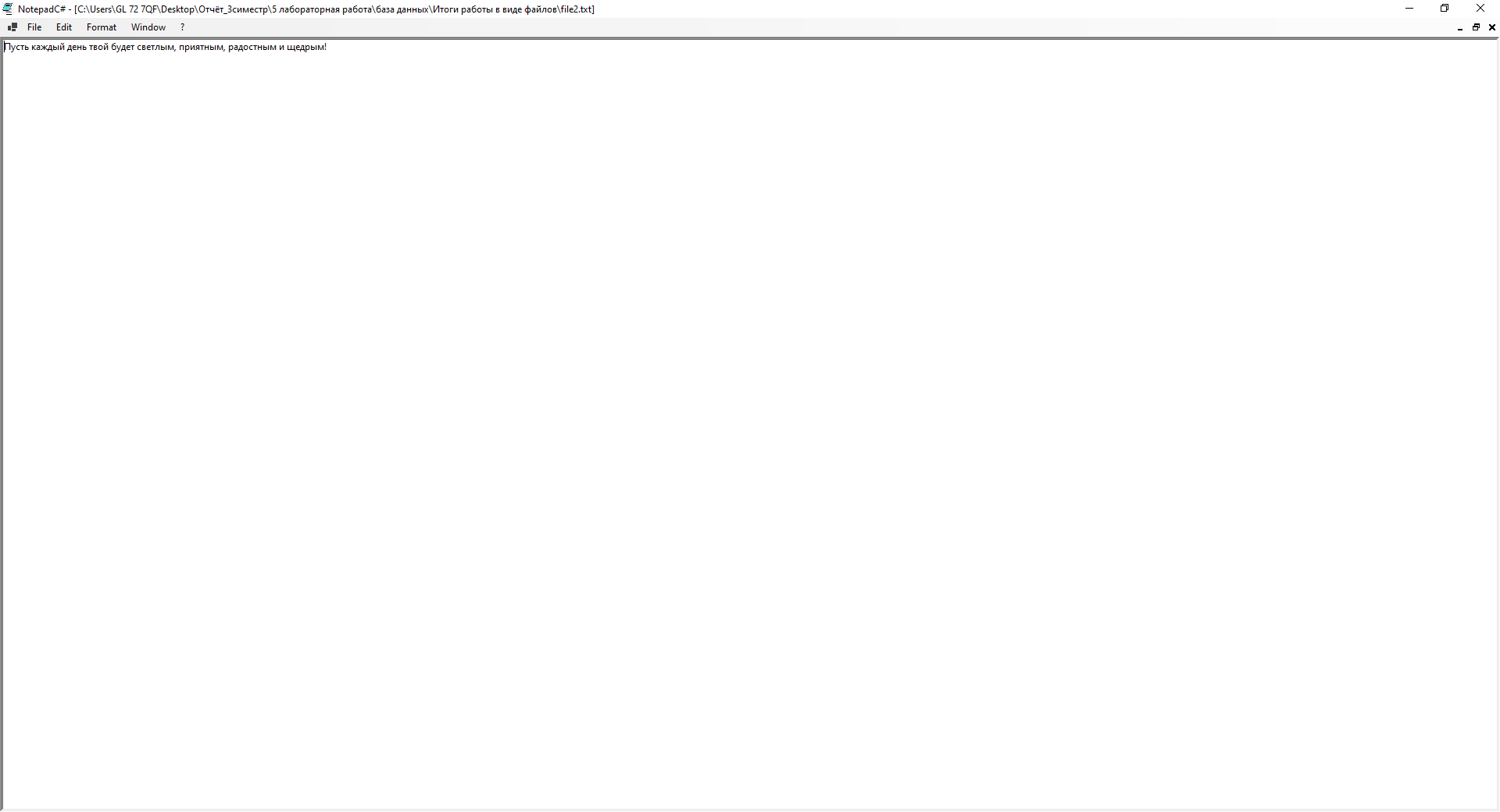
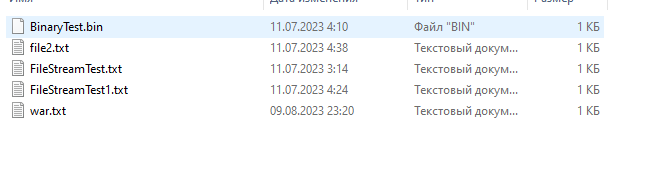
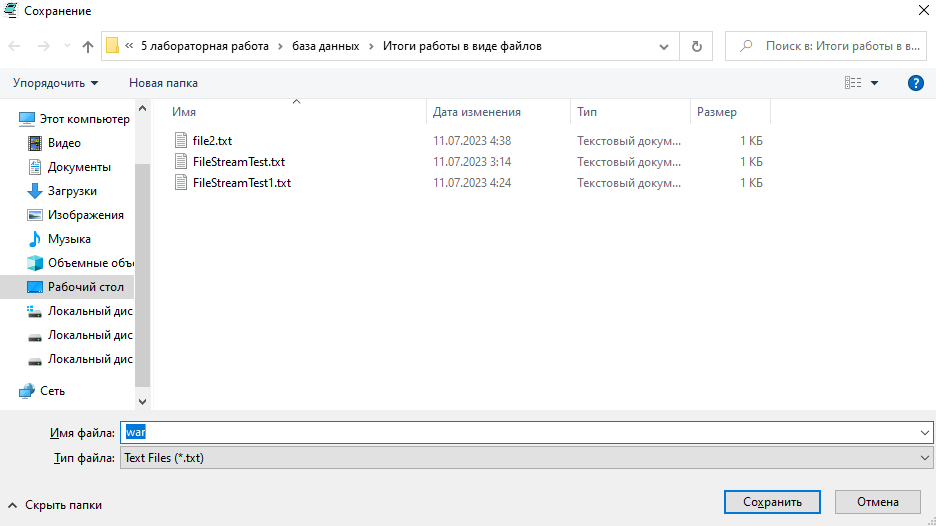
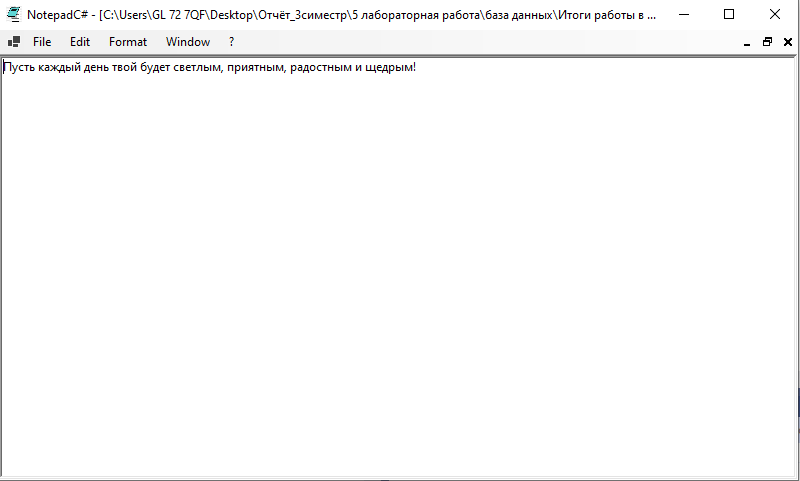
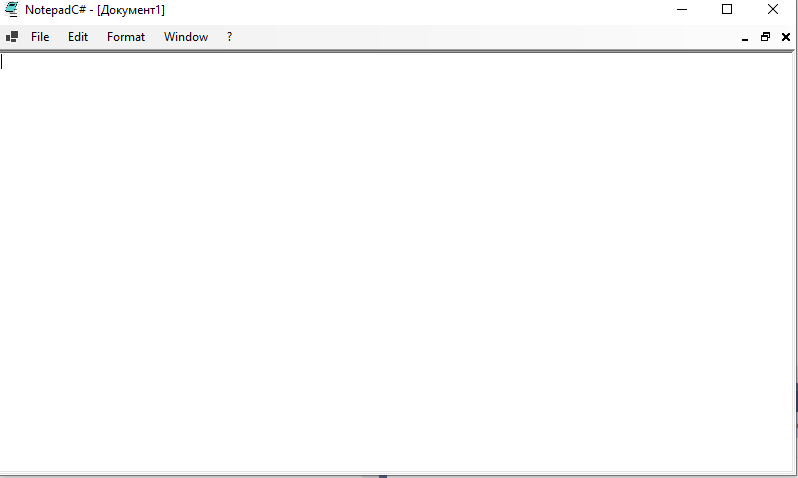
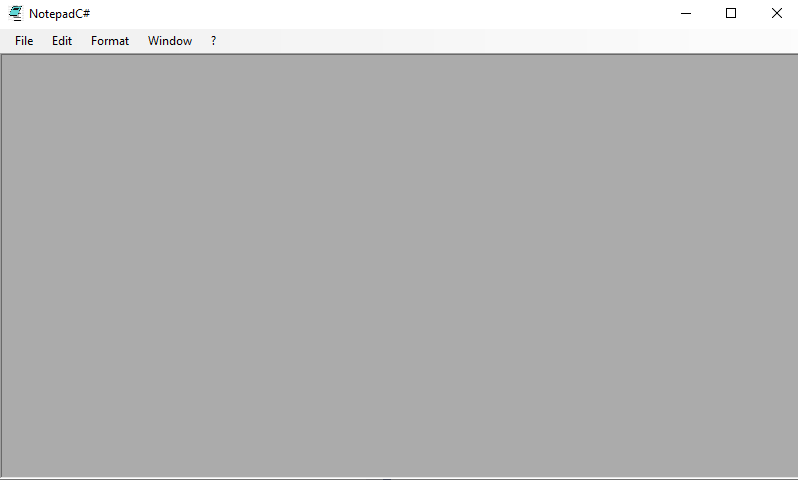
this.Save(this.DocName);

}

}

}

Результат работы:



Я не сдал расписывать данный пример всё взято с отчёта

Только последние изображение показывает нам принцип много поточности в приложениях этот принцип позволяет выполнять действия пользователей в разных каналах памяти что позволяет значительно упростить работу людей и программистов.

А теперь перейдём к самому интересному:

Приложение Database:

Программный Код:

1. Cтруктура программы:
2. struct VariablesStruct
3. {
4. static public MySqlDataAdapter msDataAdapter;
5. static public MySqlConnection mySqlconnection;
6. static public MySqlCommand msCommand;
7. static public MySqlConnection connection;
8. static public string server;
9. static public string database;
10. static public string user;
11. static public string password;
12. static public string port;
13. static public string connectionString;
14. static public string sslM;
15. }

Я вынес в отдельный файл так как накопились переменные

Класс для подключение к базе данных DBconnection:

public class DBconnection

{

public static DataGridView dgv = new DataGridView();// переменная для передачи между формами

public static bool ConnectionDB() // буллевый метод

{

server = "localhost"; // Сервер

database = "avto"; // база данных

user = "root"; // имя пользовтеля

password = "z147z147"; // пароль от бд

port = "3307"; // порт

sslM = "none"; // безопасность в бд так как у нас локальканая бд она не нужна

try

{

connectionString = String.Format("server={0};port={1};user id={2}; password={3}; database={4}; SslMode={5}", server, port, user, password, database, sslM); // переменная для подключение бд

mySqlconnection = new MySqlConnection(connectionString); // создание экземпляра класса подключения

mySqlconnection.Open(); // открытие соединения

msCommand = new MySqlCommand(); // создания класса для SQL COMMANT

msCommand.Connection = mySqlconnection; // реализация SQL COMMANTOMMANT

msDataAdapter = new MySqlDataAdapter(msCommand); // класс для адаьтации в таблице например

return true;

}

catch

{

MessageBox.Show("Ошибка соединение с базой данных", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error); // обработка исключения

return false;

}

}

public static void CloseDB() // метод для закрытия соединения

{

mySqlconnection.Close();

}

public MySqlConnection GetConnection() // get метод возращает соединенеи

{

return mySqlconnection;

}

}

Код окна Form1:

public partial class Form1 : Form

{

private Table table; // объявляем переменную для создание окна Table

public Form1()

{

InitializeComponent();

}

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

{

DBconnection.ConnectionDB(); // Открываем соединение с бд

table = new Table(); // объявляем экземпляр

table.MdiParent = this; // конструкция позволяет открывать окно в нашем окне this обращение к текущему окну

table.Show(); // вызываем метод Show

}

}

Форма Table:

public partial class Table : Form

{

public bool IsSaved = false; // обьявление перемменой для проверки

public String DocName; // обьявление String переменной

public SD.DataSet data; // обьявление DataSet переменной для обработки наших данных из бд

public object Table1 { get; set; } // обьявляем обьект который влзвращает get и set методы

public Table()

{

InitializeComponent();

string script = "SELECT \* FROM avtocar"; // отображение данных из бд ( формируем скрипт)

MySqlConnection connection = new MySqlConnection(connectionString);// создание экземпляра класса подключения

connection.Open();// открытие соединения

MySqlDataAdapter reader = new MySqlDataAdapter(script, connectionString); // реализация скрипта

SD.DataTable table = new SD.DataTable(); // создание экземпляра таблицы

reader.Fill(table); // реализация sql запроса и рендерим в записи в таблицу

dataGridView1.DataSource = table; // отображение таблицы в datagrid

connection.Close();// закрытие соединения

}

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

{

DBconnection.dgv.DataSource = dataGridView1.DataSource; // отопражения запроса в окна новый\_Запрос

Новый\_запрос новый\_Запрос = new Новый\_запрос();

новый\_Запрос.Show();

}

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

{

dataGridView1.DataSource = DBconnection.dgv.DataSource;// отображение запроса из окна новый\_запрос

}

}

Форма Новый запрос:

public partial class Новый\_запрос : Form

{

public Новый\_запрос()

{

InitializeComponent();

}

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

{

string script = richTextBox1.Text; // записываем передоваемое значение из richTextBox1.Text

MySqlConnection connection = new MySqlConnection(connectionString); // создание экземпляра класса подключения

connection.Open();// открытие соединения

MySqlDataAdapter reader = new MySqlDataAdapter(script, connectionString);// реализация скрипта

SD.DataTable table = new SD.DataTable();// создание экземпляра таблицы

reader.Fill(table);// реализация sql запроса и рендерим в записи в таблицу

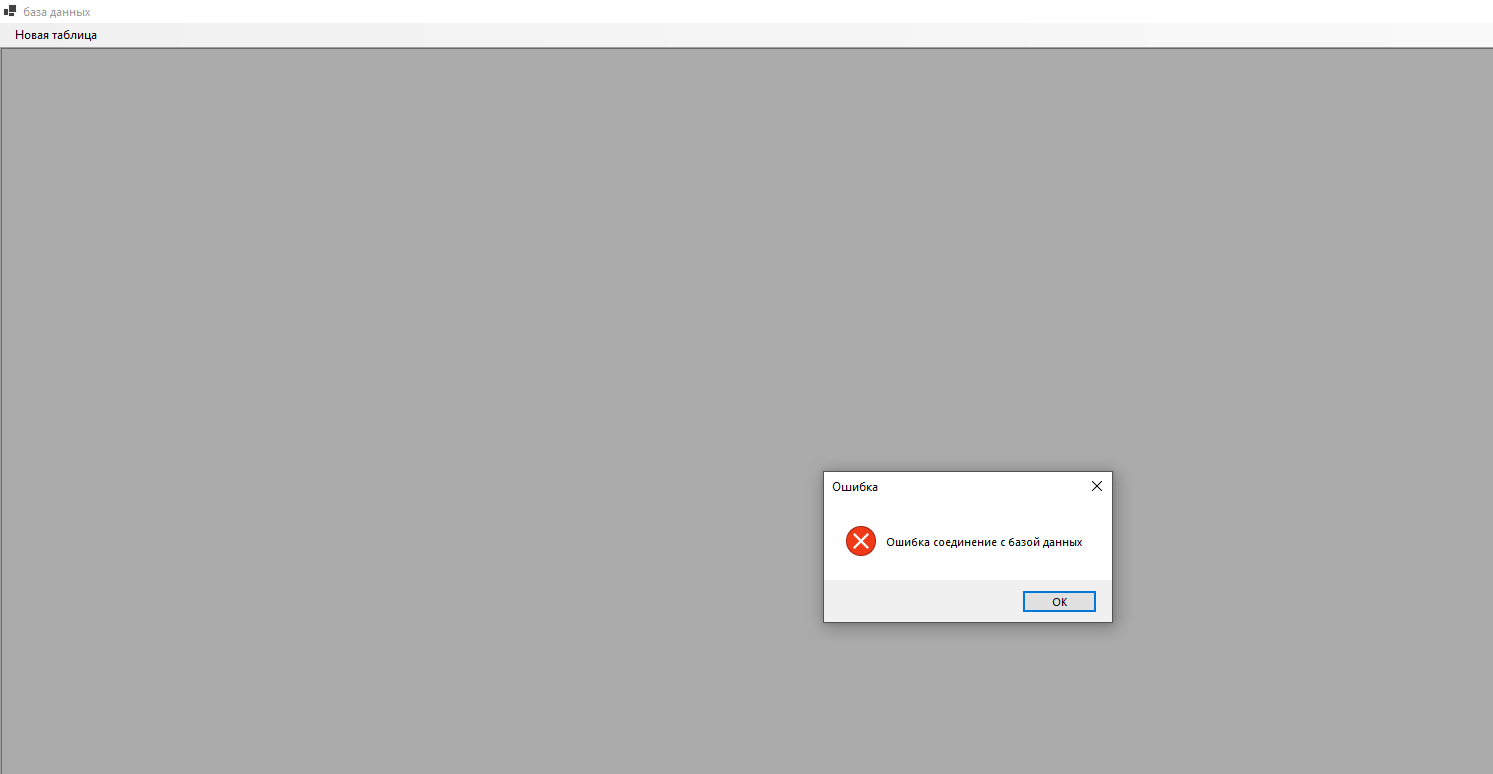
DBconnection.dgv.DataSource = table;// отправляем запрос

connection.Close();// закрытие соединения

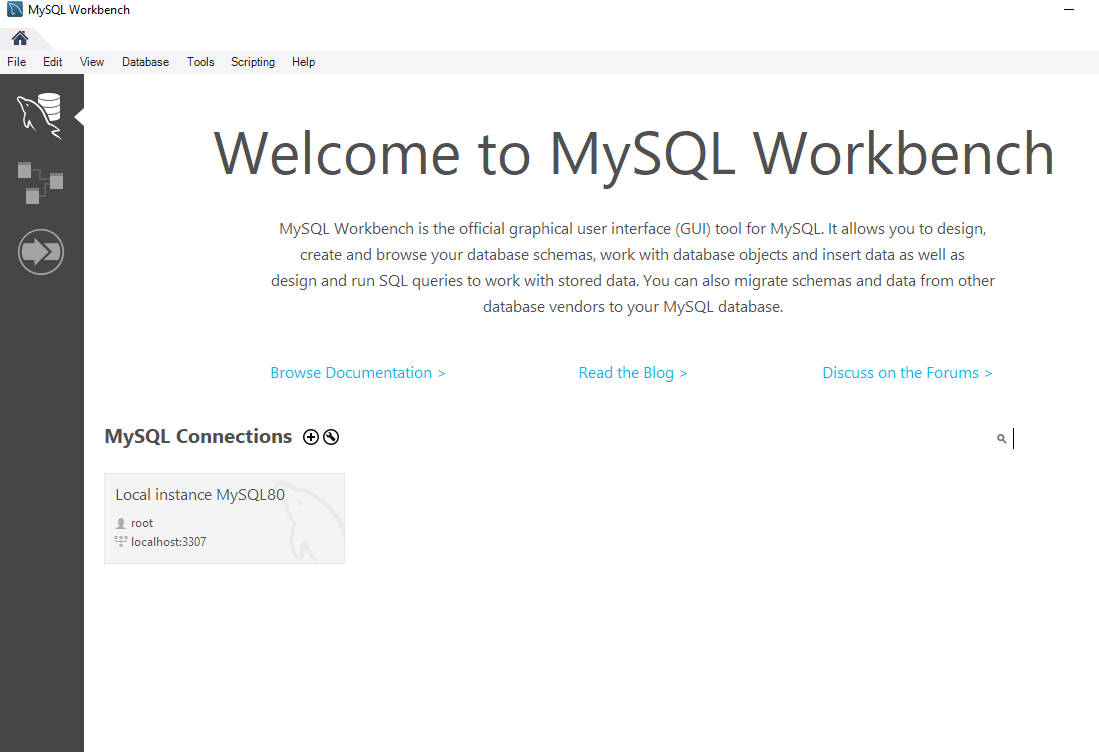
}

}

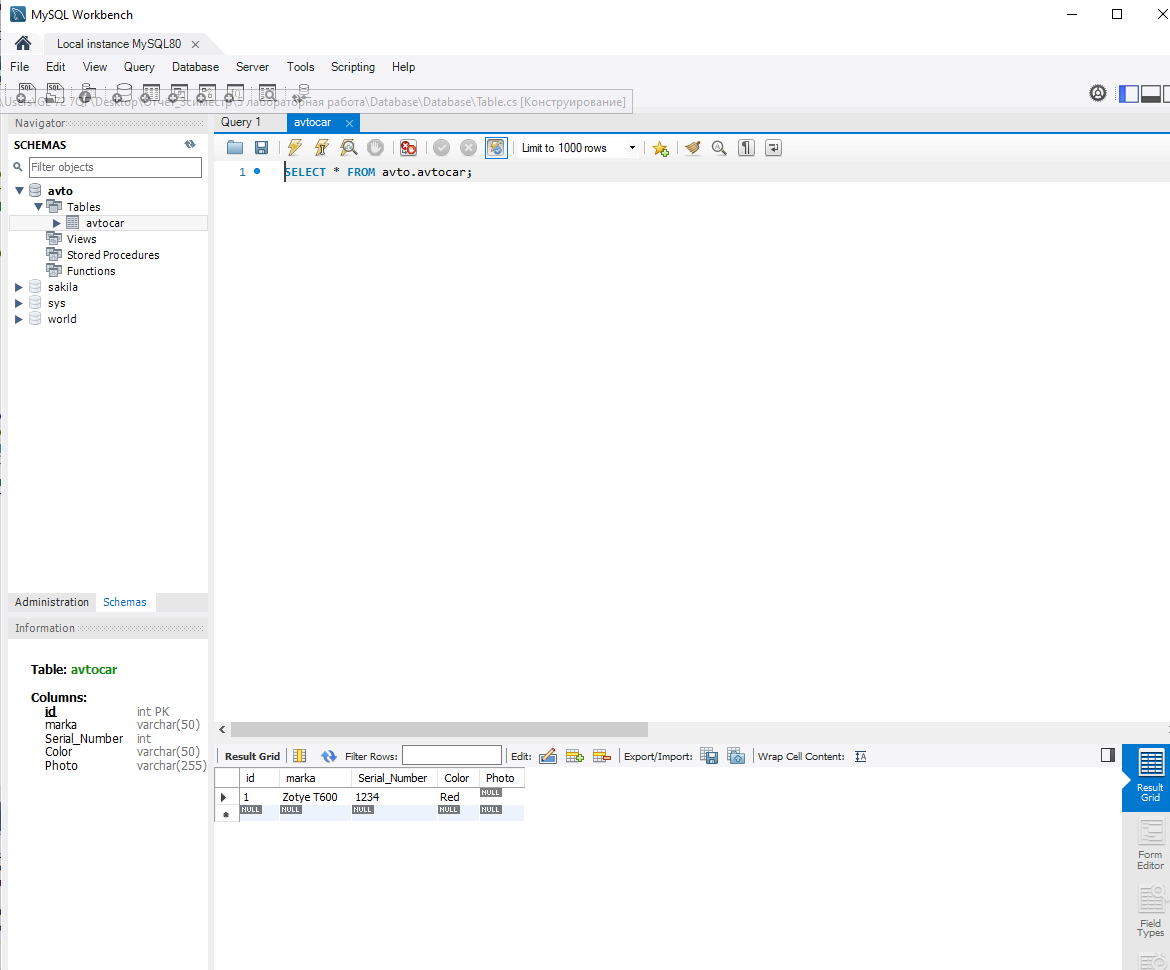
Результат работы:



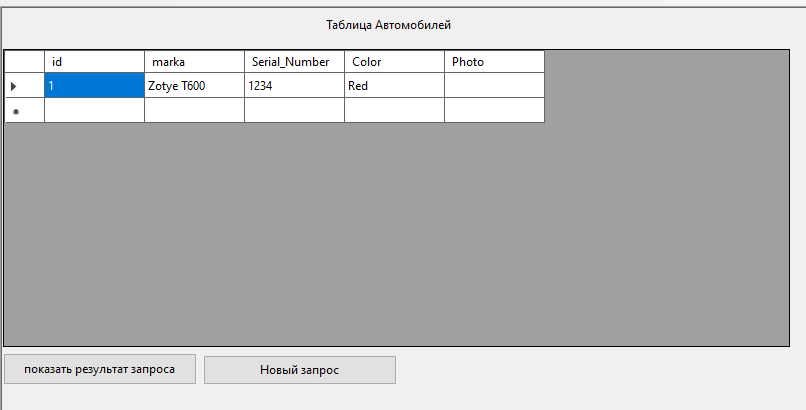
Витим что надо подключиться к базе данных выполняем следующие:



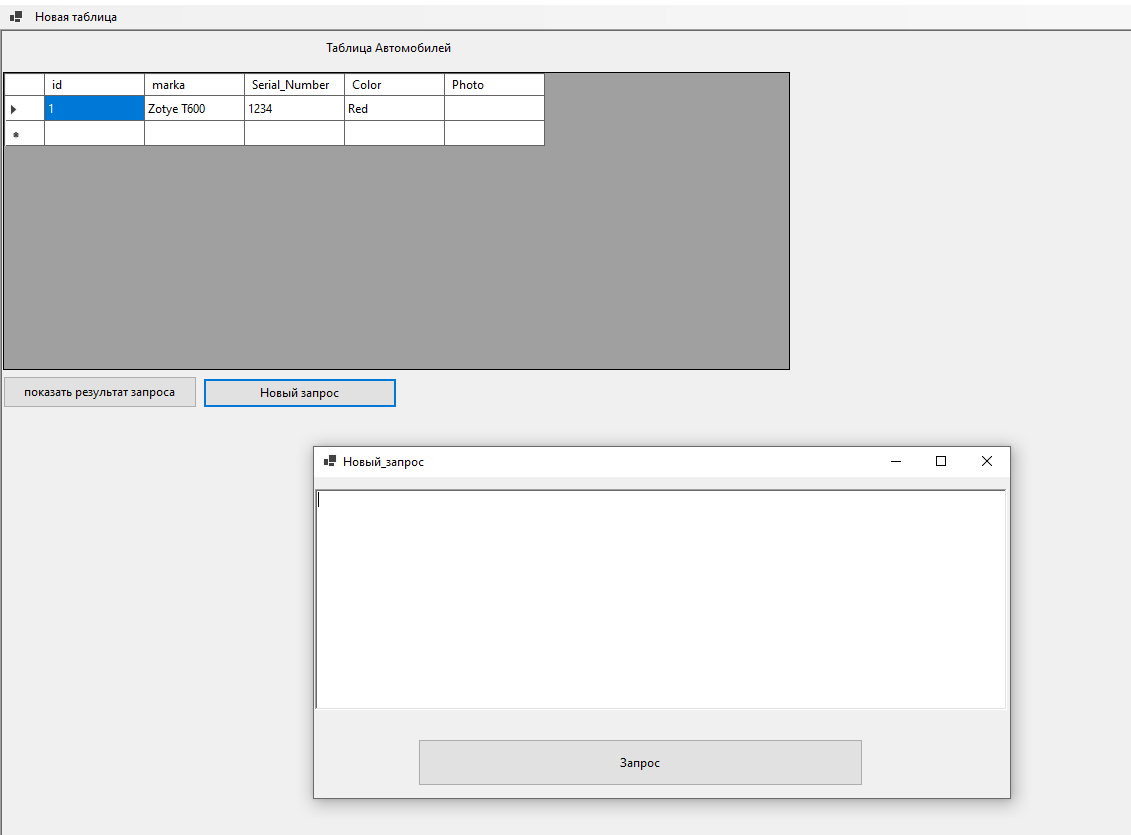
Заходим



Витим что есть данные в базе данных для теста

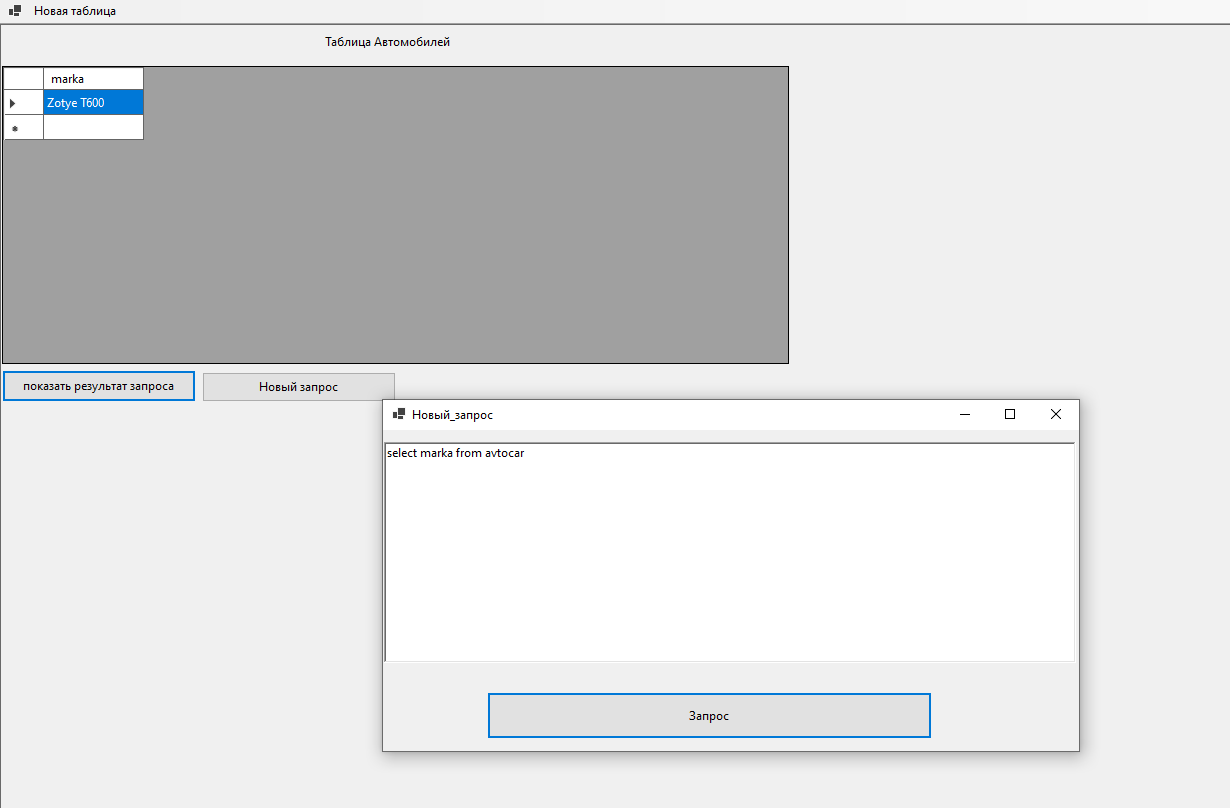


Нажимаем на “Новая таблица” и видим вот такой результат



Теперь нажимаем на кнопку Новый запрос сплывает окно для запросов к базе данных я реализовал обработку SQL запросов

SQL запросы позволяют нам влиять на данные в таблице их как я знаю 4 типа это: “DELETE” “INSERT INTO VALUES” “SELECT” “UPDATE SET” и при нажатие на кнопку “Запрос” отправляем SQL скрипт.



Примерный результат “Запрос ”отправляет скрипт перевод на русский выбрать marka из таблицы avtocar.

Кнопка “Показать результат запроса” реализована для отображения скриптов по запросу.