**Лабораторная работа № 7**

**«ОРГАНИЗАЦИЯ ЗАПРОСОВ К СУБД ACCESS»**

1. **Постановка задачи**

Разработать тестовую программу на основе элементов управления: Button, DataGridView, TextBox, ListBox – с возможностью записи результатов ответа пользователя, а также выполнить авторизацию пользователя из СУБД Access.

Кроме того, следует организовать SQL-запросы к базе данных: чтение данных из БД в dataGridView (2 способа), обновление записей в таблице базы данных, вставка записи в БД, удаление записей из БД, организация вывода табличных данных из БД в элемент управления listBox, чтение в текстовое окно Логина и Пароля из БД, выбор из таблицы БД только тех записей, в которых поле Логин содержит определенный символ.

Для этого было необходимо разработать следующие методы:

* Метод ***BD\_1()***, предназначенный для чтение данных из БД в dataGridView с использованием объектов класса Command.
* Метод ***BD\_2()***, предназначенный для чтения данных из БД в dataGridView.
* Метод ***Change()***, предназначенный для обновления записей в таблице базы данных.
* Метод ***NewLogPass()***, предназначенный для вставки записи в БД.
* Метод ***DeleteLog()***, предназначенный для удаления записей из БД.
* Метод ***Vivod\_LS()***, предназначенный для вывода табличных данных из БД в элемент управления listBox.
* Метод ***ReadTB()***, предназначенный для чтения в текстовое окно Логина и Пароля из БД.
* Метод ***SortDBinDGV()***, предназначенный для выбора из таблицы БД только тех записей, поле Логин которых начинается на определенный символ (либо содержит определенный символ).

1. **Разработка и результат проекта**

Разработанная форма интерфейса пользователя представлена на рисунке 1.



Рисунок 1 – Интерфейс пользователя

Выполним чтение данных из БД в dataGridView (первый способ) с использованием объектов класса Command. Результат представлен на рисунке 2.

public static void BD\_1(ref DataGridView dgw)

{

string query = "Select \* From [Таблица1]";

string f1 = @"Provider = Microsoft.Jet.OLEDB.4.0; Data Source = C:\Users\Bulat\Desktop\Мои проекты\Курсовые\АиП\\_Курсовая\_\bin\Debug\DB11.accdb";

var f = new OleDbConnection(f1);

f.Open();

OleDbCommand command = new OleDbCommand(query, f);

var Reader = command.ExecuteReader();

var Table = new DataTable();

Table.Columns.Add(Reader.GetName(0));

Table.Columns.Add(Reader.GetName(1));

Table.Columns.Add(Reader.GetName(2));

while (Reader.Read() == true)

Table.Rows.Add(Reader.GetValue(0), Reader.GetValue(1), Reader.GetValue(2));

Reader.Close();

dgw.DataSource = Table;

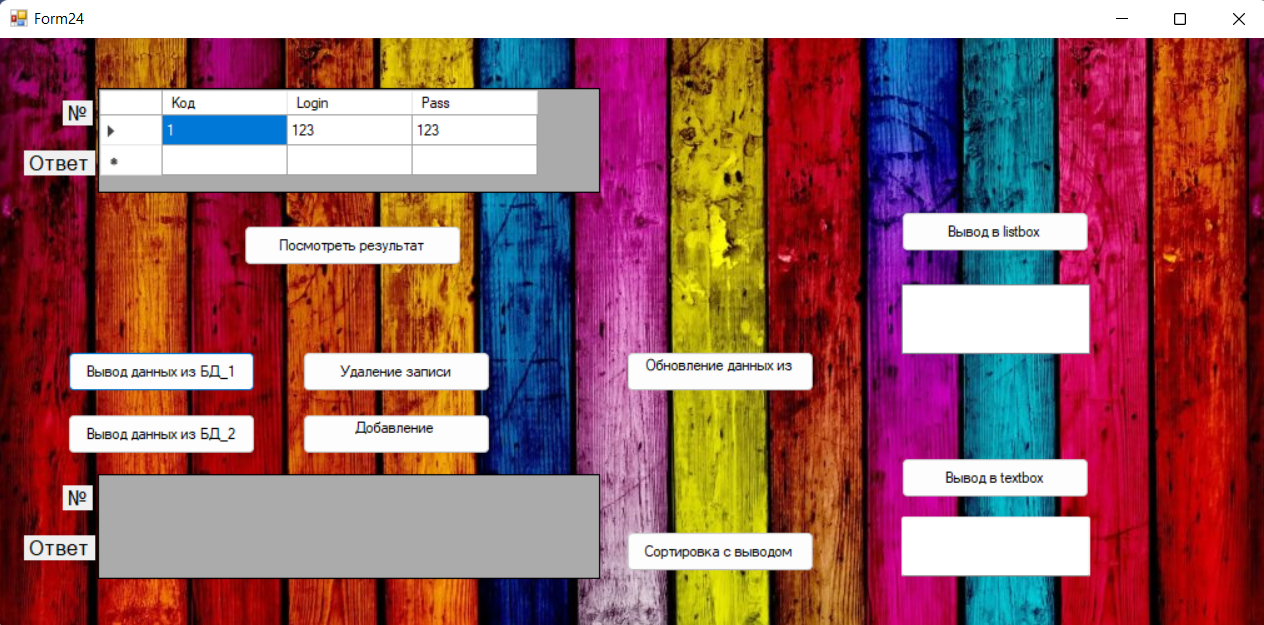
}

Рисунок 2 – Чтение БД с классом Command

Выполним чтение данных из БД в dataGridView (второй способ). Обновление записей в таблице базы данных. Результат представлен на рисунке 3-4.

public static void BD\_2(ref DataGridView dgw)

{

string p1 = @"Provider = Microsoft.Jet.OLEDB.4.0; Data Source = C:\Users\Bulat\Desktop\Мои проекты\Курсовые\АиП\\_Курсовая\_\bin\Debug\DB11.accdb";

var p = new OleDbConnection(p1);

if (p.State == ConnectionState.Closed)

p.Open();

var data = new DataSet();

var adapter = new OleDbDataAdapter("Select \* From Таблица1", p);

adapter.Fill(data, "Таблица1");

dgw.DataSource = data;

dgw.DataMember = "Таблица1";

public static void Change()

{

string p1 = @"Provider = Microsoft.Jet.OLEDB.4.0; Data Source = C:\Users\Bulat\Desktop\Мои проекты\Курсовые\АиП\\_Курсовая\_\bin\Debug\DB11.accdb";

var p = new OleDbConnection(p1);

p.Open();

string LogAp = Interaction.InputBox("Введите логин, котроый хотите измененить", "Заголовок окна", " ", -1, -1);

string Log = Interaction.InputBox("Новый логин", "Заголовок окна", " ",- 1, -1);

string Par = Interaction.InputBox("Новый пароль", "Заголовок окна", " ",-1, -1);

string update\_query = "UPDATE [Таблица1] SET [Login]=@Log, [Pass]=@Pas WHERE[Login] = @LogAp";

OleDbCommand command\_4 = new OleDbCommand(update\_query, p);

command\_4.Parameters.Add("@Log", OleDbType.VarWChar).Value = Log;

command\_4.Parameters.Add("@Par", OleDbType.VarWChar).Value = Par;

command\_4.Parameters.Add("@LogAp", OleDbType.Char).Value = LogAp;

command\_4.ExecuteNonQuery();

p.Close();

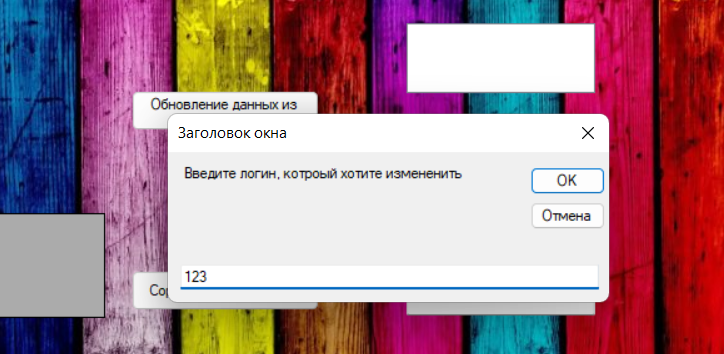
}

Рисунок 3 – Диалоговое окно обновления логина

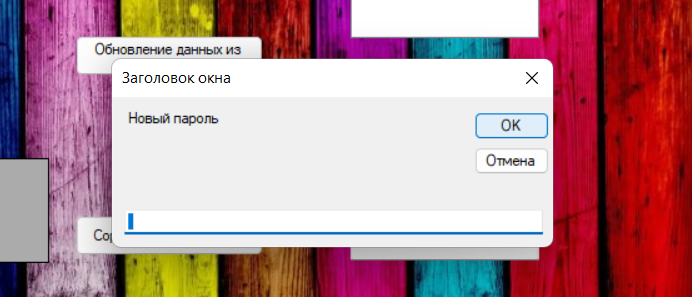


Рисунок 4 – Диалоговое окно создания нового пароля

Выполним вставку записи в БД. Результат представлен на рисунке 5-6.

public static void NewLogPass()

{

string p1 = @"Provider = Microsoft.Jet.OLEDB.4.0; Data Source = C:\Users\Bulat\Desktop\Мои проекты\Курсовые\АиП\\_Курсовая\_\bin\Debug\DB11.accdb";

var p = new OleDbConnection(p1);

p.Open();

string Log = Interaction.InputBox("Регистрация логина", "Заголовок окна", " ", -1, -1);

string Par = Interaction.InputBox("Регистрация пароля", "Заголовок окна", " ", -1, -1);

string query = "INSERT INTO Таблица1 (Login, Pass) VALUES (@Log, @Par)";

OleDbCommand command = new OleDbCommand(query, p);

command.Parameters.Add("@Log", OleDbType.VarWChar).Value = Log;

command.Parameters.Add("@Par", OleDbType.VarWChar).Value = Par;

command.ExecuteNonQuery();

p.Close();

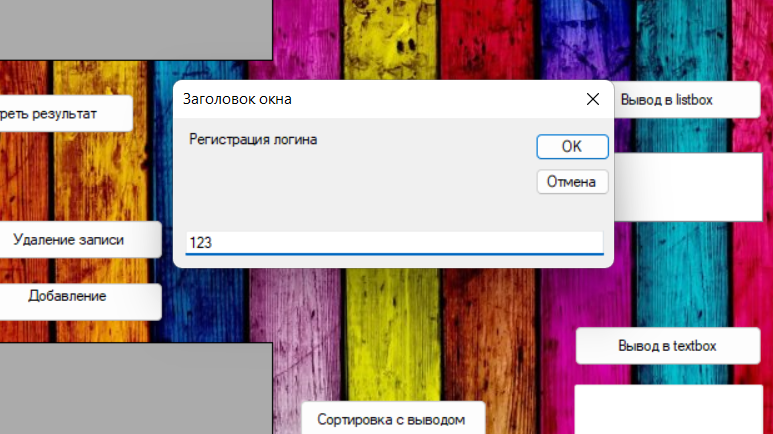
}

Рисунок 5 – Регистрация логина

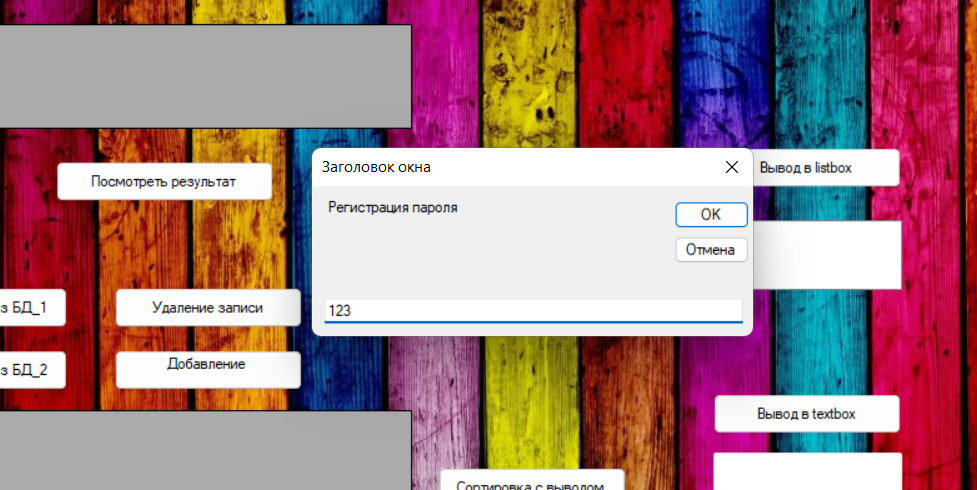


Рисунок 6 – Регистрация пароля

Выполним удаление записей из БД. Результат представлен на рисунке 7.

public static void DeleteLog()

{

string p1 = @"Provider = Microsoft.Jet.OLEDB.4.0; Data Source = C:\Users\Bulat\Desktop\Мои проекты\Курсовые\АиП\\_Курсовая\_\bin\Debug\DB11.accdb";

var p = new OleDbConnection(p1);

p.Open();

string Log = Interaction.InputBox("Введите логин для удаления записи из БД", "Заголовок окна", " ", -1, -1);

string query = "DELETE \* FROM Таблица1 WHERE " + "Login Like @Log";

OleDbCommand command = new OleDbCommand(query, p);

command.Parameters.Add("@Log", OleDbType.VarWChar).Value = Log;

var i = command.ExecuteNonQuery();

if (i > 0)

MessageBox.Show("Запись удалена", "Заголовок окна",

MessageBoxButtons.OK, MessageBoxIcon.Information);

if (i == 0)

MessageBox.Show("Запись не найдена", "Заголовок окна", MessageBoxButtons.OK, MessageBoxIcon.Error);

p.Close();

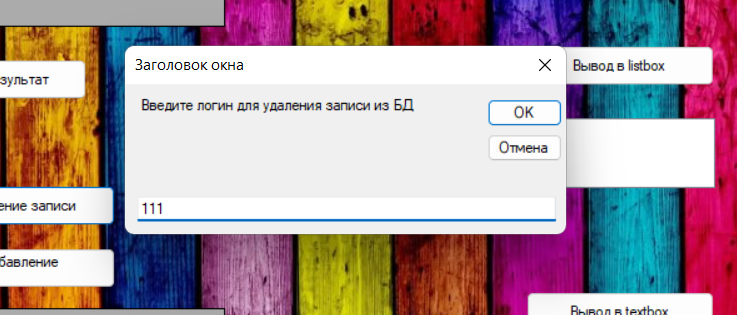
}

Рисунок 7 – Диалоговое окно удаления записи

Организуем вывод табличных данных из БД в элемент управления listBox. Результат представлен на рисунке 8.

public static void Vivod\_LS(ref ListBox LB)

{

string p1 = @"Provider = Microsoft.Jet.OLEDB.4.0; Data Source = C:\Users\Bulat\Desktop\Мои проекты\Курсовые\АиП\\_Курсовая\_\bin\Debug\DB11.accdb";

var p = new OleDbConnection(p1);

p.Open();

string query = "SELECT Login, Pass FROM Таблица1 ORDER BY Login";

OleDbCommand command = new OleDbCommand(query, p);

OleDbDataReader reader = command.ExecuteReader();

LB.Items.Clear();

while (reader.Read())

{

LB.Items.Add(reader[0].ToString() + " " + reader[1].ToString() + "");

}

reader.Close();

}

Рисунок 8 – Вывод в ListBox

Выполним чтение в текстовое окно Логина и Пароля из БД. Результат представлен на рисунке 13.

public static void ReadTB(ref TextBox TB)

{

string p1 = @"Provider = Microsoft.Jet.OLEDB.4.0; Data Source = C:\Users\Bulat\Desktop\Мои проекты\Курсовые\АиП\\_Курсовая\_\bin\Debug\DB11.accdb";

var p = new OleDbConnection(p1);

p.Open();

string query = "SELECT Login, Pass FROM Таблица1 ORDER BY Login";

OleDbCommand command = new OleDbCommand(query, p);

OleDbDataReader reader = command.ExecuteReader();

while (reader.Read())

{

TB.Text += reader[0].ToString() + " " + reader[1].ToString() + "; ";

}

reader.Close();

p.Close();

}



Рисунок 13 – Вывод в TextBox

Выполним выбор из таблицы БД только тех записей, поле Логин которых содержит определенный символ. Результат представлен на рисунке 14.

public static void SortDBinDGV(ref DataGridView DGV)

{

string p1 = @"Provider = Microsoft.Jet.OLEDB.4.0; Data Source = C:\Users\Bulat\Desktop\Мои проекты\Курсовые\АиП\\_Курсовая\_\bin\Debug\DB11.accdb";

var p = new OleDbConnection(p1);

p.Open();

string Sort = Interaction.InputBox("Выбор из таблицы БД только тех записей, поле Логин которых содержит определенный(-ые) символ(-ы).", "Заголовок окна", " ", -1, -1);

var adapter = new OleDbDataAdapter($"SELECT \* FROM Таблица1 WHERE (Login LIKE '%{Sort}%')", p);

var data = new DataSet();

adapter.Fill(data, "Users");

DGV.DataSource = data;

DGV.DataMember = "Users";

p.Close();

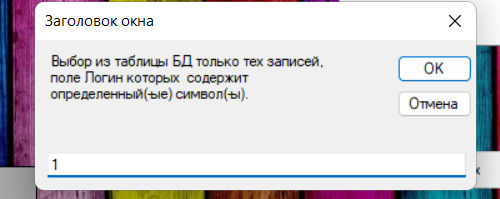
}

Рисунок 14 – Диалоговое окно выборки

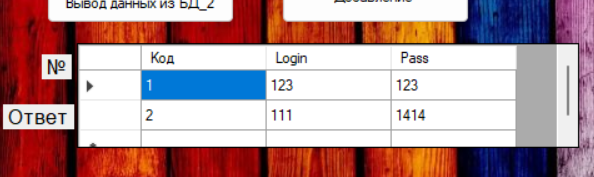


Рисунок 15 – Результат выборки