

## Код входа в учётную запись

```
namespace DemBolshakova
{
    Ссылка 7
    public partial class Log_in : Form
    {
        DataBase database = new DataBase();
        Ссылка 3
        public Log_in()
        {
            InitializeComponent();
        }

        Ссылка 1
        private void log_in_Load(object sender, EventArgs e)
        {
            textBox2.PasswordChar = '*';
            textBox1.MaxLength = 50;
            textBox2.MaxLength = 50;
        }

        Ссылка 1
        private void button1_Click(object sender, EventArgs e)
        {
            var loginUser = textBox1.Text;
            var passUser = textBox2.Text;

            SqlDataAdapter adapter = new SqlDataAdapter();
            DataTable table = new DataTable();

            string querystring = $"select id_user,login_user,password_user from register where login_user = '{loginUser}' and password_user = '{passUser}'";

            SqlCommand command = new SqlCommand(querystring, database.getConnection());

            SqlCommand command = new SqlCommand(querystring, database.getConnection());

            adapter.SelectCommand = command;
            adapter.Fill(table);

            if (table.Rows.Count == 1)
            {
                MessageBox.Show("Вы успешно вошли!", "Успешно!", MessageBoxButtons.OK, MessageBoxIcon.Information);
                Form1 form1 = new Form1();
                this.Hide();
                form1.ShowDialog();
                this.Show();
            }
            else
            {
                MessageBox.Show("Такого аккаунта не существует!", "Аккаунта не существует!", MessageBoxButtons.OK, MessageBoxIcon.Warning);
            }
        }

        Ссылка 1
        private void label3_Click(object sender, EventArgs e)
        {
            Sing_up sing_up = new Sing_up();
            sing_up.Show();
            this.Hide();
        }
    }
}
```

1. Конструктор Log\_in(): Инициализирует компоненты формы.

2. Метод log\_in\_Load():

- Устанавливает парольный символ (\*) для текстового поля для ввода пароля (textBox2).
- Ограничивает максимальную длину ввода для полей логина и пароля до 50 символов.

3. Метод button1\_Click (кнопка "Войти"):

- Получает логин и пароль из текстовых полей.
- Создаёт SQL-запрос для проверки введённых данных в базе данных.
- Если существует запись с таким логином и паролем, то отображается сообщение об успешном входе и открывается форма Form1. Форма входа скрывается.
- Если аккаунт не найден, выводится предупреждающее сообщение.

4. Метод label3\_Click (ссылка для перехода на форму регистрации):

- При клике на метку, которая, вероятно, представляет ссылку для регистрации нового пользователя, открывается форма Sing\_up для регистрации. Форма входа скрывается.

## Код регистрации учётной записи

```
namespace DemBolshakova
{
    Ссылка: 4
    public partial class Sing_up : Form
    {
        DataBase dataBase = new DataBase();
        Ссылка: 1
        public Sing_up()
        {
            InitializeComponent();
        }

        Ссылка: 1
        private void label3_Click(object sender, EventArgs e)
        {
            Log_in log_In = new Log_in();
            log_In.Show();
            this.Hide();
        }

        Ссылка: 1
        private void button1_Click(object sender, EventArgs e)
        {
            if (checkUser())
            {
                return;
            }

            var login = textBox_log.Text;
            var password = textBox_pass.Text;

            string querystring = $"insert into register(login_user, password_user) values('{login}','{password}')";

            SqlCommand command = new SqlCommand(querystring, dataBase.getConnection());

            dataBase.openConnection();

            if (command.ExecuteNonQuery() == 1)
            {
                MessageBox.Show("Аккаунт успешно создан!", "Успех!");
                Log_in log_In = new Log_in();
                this.Hide();
                log_In.ShowDialog();
            }
            else
            {
                MessageBox.Show("Аккаунт не создан!");
            }
            dataBase.closeConnection();
        }
        Ссылка: 1
        private Boolean checkUser()
        {
            var loginUser = textBox_log.Text;
            var passUser = textBox_pass.Text;

            SqlDataAdapter adapter = new SqlDataAdapter();
            DataTable table = new DataTable();

            string querystring = $"select id_user, login_user, password_user from register where login_user = '{loginUser}' and password_user = '{passUser}'";

            SqlCommand command = new SqlCommand(querystring, dataBase.getConnection());

            Ссылка: 1
            private Boolean checkUser()
            {
                var loginUser = textBox_log.Text;
                var passUser = textBox_pass.Text;

                SqlDataAdapter adapter = new SqlDataAdapter();
                DataTable table = new DataTable();

                string querystring = $"select id_user, login_user, password_user from register where login_user = '{loginUser}' and password_user = '{passUser}'";

                SqlCommand command = new SqlCommand(querystring, dataBase.getConnection());

                adapter.SelectCommand = command;

                adapter.Fill(table);

                if (table.Rows.Count > 0)
                {
                    MessageBox.Show("Пользователь уже существует!");
                    return true;
                }
                else
                {
                    return false;
                }
            }
        }
    }
}
```

Ножницы

1. Конструктор `Sing_up()`: Инициализирует компоненты формы.

2. Метод `label3_Click` (ссылка для перехода на форму входа): При клике на метку (ссылку) происходит переход на форму входа (`Log_in`). Форма регистрации скрывается.

3. Метод `button1_Click` (кнопка "Зарегистрироваться"):

- Проверяет, существует ли уже пользователь с таким логином и паролем, вызывая метод `checkUser()`.
- Если пользователь уже существует, метод возвращает `true`, и регистрация не продолжается.
- Если пользователь не найден, создаётся SQL-запрос для добавления нового пользователя в базу данных.
- Выполняется команда для вставки данных (логин и пароль) в таблицу `register`.
- Если запрос выполнен успешно (затронуто 1 строка), выводится сообщение об успешной регистрации и открывается форма входа. Текущая форма скрывается.
- Если команда не выполнена, выводится сообщение о неудаче.

4. Метод `checkUser` (проверка на существующего пользователя):

- Получает логин и пароль из текстовых полей.
- Формирует SQL-запрос, чтобы проверить, существует ли уже пользователь с такими логином и паролем в базе данных.
- Если пользователь существует, выводится сообщение, и метод возвращает `true`.
- Если пользователь не найден, метод возвращает `false`.

#### 4 практическая работа

```
namespace DemBolshakova
{
    Ссылка 6
    public partial class Form1 : Form
    {
        DataBase database = new DataBase();

        Ссылка 2
        public Form1()
        {
            InitializeComponent();
        }

        Ссылка 1
        private void LoadData(object sender, EventArgs e)
        {
            string connectionString = "Data Source= LAPTOP-BLTQNR9D;Initial Catalog=DataGridView;" + "Integrated Security=true;";

            SqlConnection myConnection = new SqlConnection(connectionString);

            myConnection.Open();

            string query = "SELECT * FROM Group_IS ORDER BY fspo_id";

            SqlCommand command = new SqlCommand(query, myConnection);

            SqlDataReader reader = command.ExecuteReader();

            List<string[]> data = new List<string[]>();
```

```
            while (reader.Read())
            {
                data.Add(new string[3]);

                data[data.Count - 1][0] = reader["fspo_id"].ToString();
                data[data.Count - 1][1] = reader["fspo_fio"].ToString();
                data[data.Count - 1][2] = reader["fspo_name"].ToString();
            }

            reader.Close();

            myConnection.Close();

            foreach (string[] s in data)
                dataGridView1.Rows.Add(s);
        }
    }
}
```

```
        Ссылка 1
        private void button1_Click(object sender, EventArgs e)
        {
            string fspoFio = textBox1.Text;
            string fspoName = textBox2.Text;

            string connectionString = "Data Source= LAPTOP-BLTQNR9D;Initial Catalog=DataGridView;Integrated Security=true;";

            using (SqlConnection myConnection = new SqlConnection(connectionString))
            {
                myConnection.Open();
```

```

        myConnection.Open();

        string query = "INSERT INTO Group_IS (fspo_fio,fspo_name) VALUES (@fio, @name); SELECT SCOPE_IDENTITY();";

        using (SqlCommand command = new SqlCommand(query, myConnection))
        {
            command.Parameters.AddWithValue("@fio", fspoFio);
            command.Parameters.AddWithValue("@name", fspoName);

            int newId = Convert.ToInt32(command.ExecuteScalar());

            dataGridView1.Rows.Add(new object[] {newId, fspoFio, fspoName });
            MessageBox.Show("Вы успешно добавили", "Успешно!", MessageBoxButtons.OK, MessageBoxIcon.Information);
        }

        textBox1.Clear();
        textBox2.Clear();
    }
}

```

Ссылка: 1

```

private void button2_Click(object sender, EventArgs e)
{
    string Id = textBox3.Text;

    string connectString = "Data Source= LAPTOP-BLTQNR9D;Initial Catalog=DataGridView;Integrated Security=true;";

    string Delete = "DELETE FROM Group_IS " + "WHERE fspo_Id = " + Id;

    SqlConnection myConnection = new SqlConnection(connectString);
    string Id = textBox3.Text;
}

```

```

    string connectString = "Data Source= LAPTOP-BLTQNR9D;Initial Catalog=DataGridView;Integrated Security=true;";

    string Delete = "DELETE FROM Group_IS " + "WHERE fspo_Id = " + Id;

    SqlConnection myConnection = new SqlConnection(connectString);

    using (SqlCommand command = new SqlCommand(Delete, myConnection))
    {
        myConnection.Open();

        command.ExecuteNonQuery();

        myConnection.Close();

        MessageBox.Show("Вы успешно удалили", "Успешно!", MessageBoxButtons.OK, MessageBoxIcon.Information);
    }

    textBox3.Clear();
}
}

```

Ссылка: 1

```

private void button3_Click(object sender, EventArgs e)
{
    Form2 form2 = new Form2();
    this.Hide();
    form2.ShowDialog();
    this.Show();
}
}

```

1. Конструктор Form1(): Инициализирует форму. Этот метод автоматически вызывается при создании объекта формы.

2. Метод LoadData():

- При загрузке формы этот метод извлекает все записи из таблицы Group\_IS базы данных и отображает их в DataGridView.
- Используется SQL-запрос для извлечения всех данных, после чего они добавляются в таблицу через DataGridView.

3. Метод button1\_Click():

- Когда пользователь нажимает кнопку добавления, программа считывает значения из текстовых полей textBox1 и textBox2.
- Создаёт SQL-запрос для добавления новой записи в базу данных и выполняет его.

- После успешного добавления записи в базу, данные выводятся в DataGridView, а текстовые поля очищаются.

#### 4. Метод button2\_Click():

- Когда пользователь нажимает кнопку удаления, программа получает ID из текстового поля textBox3.
- Создаёт SQL-запрос для удаления записи с указанным ID из базы данных.
- После успешного удаления записи показывается сообщение об успехе, и текстовое поле очищается.

#### 5. Метод button3\_Click():

- Открывает другую форму (Form2) и скрывает текущую форму (Form1), чтобы пользователь мог перейти на другую страницу.
- После закрытия второй формы текущая форма снова отображается.
- Основная цель этого кода — работать с базой данных: отображать данные в таблице, добавлять новые записи и удалять старые через интерфейс пользователя.

## Практическая 5

```
namespace Dembolshakova
{
    Ссылка: 6
    public partial class Form2 : Form
    {
        private ImageList imageList;
        private int customImageIndex = 0;

        // Строка подключения к базе данных
        string connectionString = "Data Source=LAPTOP-BLTQNR9D;Initial Catalog=lab4;Integrated Security=true;";

        Ссылка: 2
        public Form2()
        {
            InitializeComponent();
            listView1.MouseDown += listView1_MouseClick;
            button1.Click += addButton_Click;
            button2.Click += browseImageButton_Click;
        }

        Ссылка: 1
        private void Form2_Load(object sender, EventArgs e)
        {
            listView1.Items.Clear();

            imageList = new ImageList();
            imageList.ImageSize = new Size(50, 50); // Размер изображения

            LoadDataFromDatabase(); // Загружаем данные из базы данных

            listView1.SmallImageList = imageList;
            listView1.View = View.Details; // Для отображения как таблицы
            listView1.Columns.Add("Image", 60); // Колонка для изображений
            listView1.Columns.Add("First Name", 100);
            listView1.Columns.Add("Last Name", 100);
        }

        Ссылка: 1
        private void listView1_MouseClick(object sender, MouseEventArgs e)
        {
            ListViewItem item = listView1.GetItemAt(e.X, e.Y);
            if (item != null && MessageBox.Show("Вы уверены, что хотите удалить этот элемент?", "Подтвердить удаление", MessageBoxButtons.YesNo) == DialogResult.Yes)
            {
                DeleteDataFromDatabase(item);
                listView1.Items.Remove(item);
            }
        }

        Ссылка: 1
        private void addButton_Click(object sender, EventArgs e)
        {
            if (string.IsNullOrEmpty(textBox1.Text) || string.IsNullOrEmpty(textBox2.Text))
            {
                MessageBox.Show("Введите имя и фамилию!", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);
                return;
            }

            byte[] imageBytes = null;

            if (pictureBox1.Image != null)
            {
                using (MemoryStream ms = new MemoryStream())
                {
                    pictureBox1.Image.Save(ms, System.Drawing.Imaging.ImageFormat.Png);
                    imageBytes = ms.ToArray();
                }
            }

            imageList.Images.Add(new Bitmap(pictureBox1.Image, imageList.ImageSize)); // Добавляем изображение в ImageList
            customImageIndex++; // Увеличиваем индекс
        }
    }
}
```

Ножницы

Снимок экрана скопирован в буфер обмена

```

        imageBytes = new byte[customImageBytes.Length]; // Копируем байты изображения
        customImageIndex++; // Увеличиваем индекс
    }

    SaveDataToDatabase(textBox1.Text, textBox2.Text, imageBytes);

    ListViewItem listViewItem = new ListViewItem(new string[] { "", textBox1.Text, textBox2.Text });
    listViewItem.ImageIndex = customImageIndex - 1; // Используем индекс последнего изображения

    listView1.Items.Add(listViewItem); // Добавляем элемент в ListView

    textBox1.Text = "";
    textBox2.Text = "";
    pictureBox1.Image = null;
}

private void browseImageButton_Click(object sender, EventArgs e)
{
    using (OpenFileDialog openFileDialog = new OpenFileDialog())
    {
        openFileDialog.Filter = "Image Files|*.jpg;*.jpeg;*.png;*.bmp";
        if (openFileDialog.ShowDialog() == DialogResult.OK)
        {
            pictureBox1.Image = Image.FromFile(openFileDialog.FileName);
        }
    }
}

private void SaveDataToDatabase(string firstName, string lastName, byte[] image)
{
    using (SqlConnection connection = new SqlConnection(connectString))
    {
        connection.Open();
    }
}

```

```

        if (openFileDialog.ShowDialog() == DialogResult.OK)
        {
            pictureBox1.Image = Image.FromFile(openFileDialog.FileName);
        }
    }
}

Ссылка 1
private void SaveDataToDatabase(string firstName, string lastName, byte[] image)
{
    using (SqlConnection connection = new SqlConnection(connectString))
    {
        connection.Open();

        string query = "INSERT INTO ListViewData (FirstName, LastName, Image) VALUES (@FirstName, @LastName, @Image)";
        using (SqlCommand command = new SqlCommand(query, connection))
        {
            command.Parameters.AddWithValue("@FirstName", firstName);
            command.Parameters.AddWithValue("@LastName", lastName);
            command.Parameters.AddWithValue("@Image", image ?? (object)DBNull.Value);

            command.ExecuteNonQuery();
        }
    }
}

Ссылка 1
private void LoadDataFromDatabase()
{
    using (SqlConnection connection = new SqlConnection(connectString))
    {
        connection.Open();

        string query = "SELECT FirstName, LastName, Image FROM ListViewData";
    }
}

```



```

private void LoadDataFromDatabase()
{
    using (SqlConnection connection = new SqlConnection(connectString))
    {
        connection.Open();

        string query = "SELECT FirstName, LastName, Image FROM ListViewData";
        using (SqlCommand command = new SqlCommand(query, connection))
        {
            using (SqlDataReader reader = command.ExecuteReader())
            {
                while (reader.Read())
                {
                    string firstName = reader.GetString(0);
                    string lastName = reader.GetString(1);
                    byte[] imageBytes = reader.IsDBNull(2) ? null : (byte[])reader[2];

                    int imageIndex = imageList.Images.Count; // Индекс нового изображения
                    if (imageBytes != null)
                    {
                        using (MemoryStream ms = new MemoryStream(imageBytes))
                        {
                            Bitmap bitmap = new Bitmap(ms);
                            imageList.Images.Add(bitmap); // Добавляем изображение в ImageList
                            imageIndex = imageList.Images.Count - 1; // Получаем индекс изображения
                        }
                    }

                    ListViewItem listViewItem = new ListViewItem(new string[] { "", firstName, lastName });
                    listViewItem.ImageIndex = imageIndex;

                    listView1.Items.Add(listViewItem); // Добавляем элемент в ListView
                }
            }
        }
    }
}

```

```

private void DeleteDataFromDatabase(ListViewItem item)
{
    using (SqlConnection connection = new SqlConnection(connectString))
    {
        connection.Open();

        string query = "DELETE FROM ListViewData WHERE FirstName = @FirstName AND LastName = @LastName";
        using (SqlCommand command = new SqlCommand(query, connection))
        {
            command.Parameters.AddWithValue("@FirstName", item.SubItems[1].Text);
            command.Parameters.AddWithValue("@LastName", item.SubItems[2].Text);

            command.ExecuteNonQuery();
        }
    }
}

```

Ссылка: 1

```

private void button3_Click(object sender, EventArgs e)
{
    Form3 form3 = new Form3();
    this.Hide();
    form3.ShowDialog();
    this.Show();
}

```

Ссылка: 1

```

private void button4_Click(object sender, EventArgs e)
{
    Form1 form1 = new Form1();
    this.Hide();
    form1.ShowDialog();
    this.Show();
}

```

1.Конструктор Form2(): Инициализирует форму и настраивает обработчики событий для кнопок и ListView. Также настраивает обработку клика по элементу в ListView.

2.Метод Form2\_Load()

- Загружает данные из базы данных в ListView при загрузке формы.

- Создаётся ImageList для хранения изображений, которые будут отображаться рядом с текстовыми данными в ListView.
- Устанавливаются параметры для отображения ListView (например, добавление колонок).

### 3.Метод listView1\_MouseClick()

- Реагирует на клик по элементам ListView. При клике открывается диалоговое окно с запросом на удаление выбранного элемента. Если пользователь подтверждает удаление, данные удаляются из базы данных и элемента ListView.

### 4.Метод addButton\_Click()

- Этот метод добавляет новые записи в базу данных и отображает их в ListView.
- Пользователь вводит имя и фамилию, а также может загрузить изображение. Если изображение добавлено, оно сохраняется в формате byte[].
- После добавления данных в базу данных, новая запись с изображением добавляется в ListView.

### 6.Метод browseImageButton\_Click()

- Позволяет пользователю выбрать изображение с помощью OpenFileDialog, которое затем отображается в PictureBox.

### 7.Метод SaveDataToDatabase()

- Сохраняет данные (имя, фамилию и изображение) в базу данных. Изображение сохраняется в виде массива байтов.

### 8.Метод LoadDataFromDatabase()

- Загружает все записи из базы данных в ListView. Каждая запись состоит из имени, фамилии и изображения.
- Изображения также сохраняются в ImageList для отображения в ListView.

### 9.Метод DeleteDataFromDatabase()

- Удаляет запись из базы данных, используя имя и фамилию для поиска соответствующей записи в таблице.

### 10.Методы button3\_Click() и button4\_Click()

- Эти методы отвечают за переход между формами. В зависимости от кнопки, открывается либо Form3, либо Form1, а текущая форма скрывается.

Общая логика:

1. Данные (имя, фамилия и изображение) добавляются в базу данных и отображаются в ListView.
2. Пользователь может удалить элемент, кликнув по нему, что вызовет запрос на удаление и удаление записи из базы данных.
3. Вся информация, включая изображения, загружается и отображается с использованием ListView и ImageList.

## Практическая 6

```
namespace Dembotshakova
{
    Ссылка: 4
    public partial class Form3 : Form
    {
        Ссылка: 1
        public Form3()
        {
            InitializeComponent();
            InitializeListView();
            string Host = Dns.GetHostName();

            string IP = Dns.GetHostByName(Host).AddressList[0].ToString();
            textBox1.Text = Host;
            textBox2.Text = IP;
        }

        Ссылка: 1
        private void InitializeListView()
        {
            listView1.View = View.Details;
            listView1.Columns.Add("Имя", 100);
            listView1.Columns.Add("Должность", 160);
            listView1.Columns.Add("Статус", 108);
        }

        private void listView1_SelectedIndexChanged(object sender, EventArgs e)
        {
            if (listView1.SelectedItems.Count > 0)
            {
                ListViewItem selectedItem = listView1.SelectedItems[0];
                textBox3.Text = selectedItem.SubItems[0].Text;
                textBox4.Text = selectedItem.SubItems[1].Text;
            }
        }
    }
}
```

```

        textBox3.Text = selectedItem.SubItems[0].Text;
        textBox4.Text = selectedItem.SubItems[1].Text;
        textBox5.Text = selectedItem.SubItems[2].Text;
    }
}

private void ClearInputFields()
{
    textBox3.Clear();
    textBox4.Clear();
    textBox5.Clear();
}

private void btnAdd_Click_1(object sender, EventArgs e)
{
    string[] row = new string[]
    {
        textBox3.Text, textBox4.Text, textBox5.Text
    };

    ListViewItem item = new ListViewItem(row);
    listView1.Items.Add(item);

    ClearInputFields();
}

private void btnEdit_Click_1(object sender, EventArgs e)
{
    if (listView1.SelectedItems.Count > 0) // Проверка на то, что выбран хотя бы один элемент
    {
        ListViewItem selectedItem = listView1.SelectedItems[0]; // Получаем выбранный элемент
    }
}

```

```

    if (listView1.SelectedItems.Count > 0) // Проверка на то, что выбран хотя бы один элемент
    {
        ListViewItem selectedItem = listView1.SelectedItems[0]; // Получаем выбранный элемент

        // Обновляем значения в колонках выбранного элемента
        selectedItem.SubItems[0].Text = textBox3.Text; // Название
        selectedItem.SubItems[1].Text = textBox4.Text; // Откуда
        selectedItem.SubItems[2].Text = textBox5.Text; // Категория

        // Очищаем поля ввода после редактирования
        ClearInputFields();
    }
    else
    {
        MessageBox.Show("Пожалуйста, выберите сотрудника для редактирования");
    }
}

Ссылка: 1
private void btnDelete_Click_1(object sender, EventArgs e)
{
    if (listView1.SelectedItems.Count > 0)
    {
        listView1.Items.Remove(listView1.SelectedItems[0]);
    }
    else
    {
        MessageBox.Show("Пожалуйста, выберите сотрудника для удаления");
    }
}

Ссылка: 1
private void button1_Click(object sender, EventArgs e)
{
}

```

```

Ссылка: 1
private void button1_Click(object sender, EventArgs e)
{
    Form2 form2 = new Form2();
    this.Hide();
    form2.ShowDialog();
    this.Show();
}
}

```

#### 1. Конструктор Form3():

- В конструкторе вызывается метод InitializeListView(), который настраивает внешний вид ListView (добавление колонок).
- Также происходит извлечение имени хоста и IP-адреса текущего компьютера с помощью классов Dns.GetHostName() и Dns.GetHostByName(). Эти данные отображаются в текстовых полях textBox1 и textBox2.

#### 2. Метод InitializeListView()

- Инициализирует ListView, добавляя три колонки: "Имя", "Должность" и "Статус".
- Устанавливает режим отображения как "View.Details", что позволяет отобразить элементы как таблицу с колонками.

#### 3. Метод listView1\_SelectedIndexChanged()

- Срабатывает при изменении выбранного элемента в ListView. Когда элемент выбирается, его данные (имя, должность, статус) отображаются в соответствующих текстовых полях (textBox3, textBox4, textBox5).

#### 4. Метод ClearInputFields()

- Очищает все текстовые поля для ввода данных.

#### 5. Метод btnAdd\_Click\_1()

- Этот метод добавляет новую строку в ListView с данными, введенными в текстовые поля. После добавления строки поля ввода очищаются.

#### 6. Метод btnEdit\_Click\_1()

- Обработывает редактирование выбранного элемента. Если выбран элемент, его данные обновляются согласно значениям в текстовых полях, и поля очищаются. Если элемент не выбран, выводится сообщение с просьбой выбрать сотрудника для редактирования.

#### 7. Метод btnDelete\_Click\_1()

- Удаляет выбранный элемент из ListView. Если элемент не выбран, выводится сообщение с просьбой выбрать сотрудника для удаления.

#### 8. Метод button1\_Click()

- Переход к другой форме Form2. Текущая форма скрывается, а новая форма отображается в модальном режиме.

#### Общая логика работы:

- Форма позволяет пользователю управлять списком сотрудников: добавлять, редактировать и удалять записи, которые содержат имя, должность и статус.
- Данные, введённые в текстовые поля, сохраняются в ListView. Когда пользователь выбирает элемент, его данные отображаются в полях для редактирования.
- Также отображается информация о сетевых параметрах компьютера (имя хоста и IP).