СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 5](#_Toc152634034)

[1. ЦЕЛИ И ЗАДАЧИ УЧЕБНОЙ ПРАКТИКИ 6](#_Toc152634035)

[1.1 Основные требования по технике безопасности во время прохождения учебной практики 6](#_Toc152634036)

[1.2 Нормативно-правовые документы 7](#_Toc152634037)

[2. ПРЕДМЕТНАЯ ОБЛАСТЬ. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ. ПОСТАНОВКА ЗАДАЧИ 9](#_Toc152634038)

[2.1 Описание предметной области 9](#_Toc152634039)

[2.2 Группы пользователей информационной системы 10](#_Toc152634040)

[2.3 Основные требования, предъявляемые к информационной системе 10](#_Toc152634041)

[3. ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ «МОИ СОТРУДНИКИ» 12](#_Toc152634042)

[3.1 Моделирование некоторых динамических аспектов системы 12](#_Toc152634043)

[3.1.1 Создание диаграммы вариантов использования 12](#_Toc152634044)

[3.1.2 Создание диаграммы деятельности 12](#_Toc152634045)

[3.2 Разработка функциональной структуры информационной системы. Карта навигации по информационной системе 13](#_Toc152634046)

[3.3 Разработка макета информационной системы «Мои Сотрудники» 13](#_Toc152634047)

[4. РАЗРАБОТКА ПРОТОТИПА ИНФОРМАЦИОННОЙ СИСТЕМЫ «МОИ СОТРУДНИКИ» 14](#_Toc152634048)

[4.1 Моделирование и разработка базы данных 14](#_Toc152634049)

[4.2 Разработка программных модулей информационной системы «Мои Сотрудники» 15](#_Toc152634050)

[5. РАЗРАБОТКА ТЕСТОВЫХ НАБОРОВ И ТЕСТОВЫХ СЦЕНАРИЕВ 116](#_Toc152634051)

[ЗАКЛЮЧЕНИЕ 117](#_Toc152634052)

[БИБЛИОГРАФИЧЕСКИЙ СПИСОК 118](#_Toc152634053)

[ПРИЛОЖЕНИЯ А 119](#_Toc152634054)

[ПРИЛОЖЕНИЯ Б 128](#_Toc152634056)

# ВВЕДЕНИЕ

Пассажирские вагоны играют ключевую роль в сфере железнодорожного транспорта, обеспечивая комфорт и безопасность путешествий. С течением времени эта отрасль претерпела значительное развитие, внедряя новые технологии и улучшая условия для пассажиров.

Начиная с классических вагонов с деревянными отделениями, современные пассажирские поезда предлагают разнообразные варианты, от купейных до открытых салонов. Инновации в дизайне и строительстве включают использование легких и прочных материалов, улучшенные системы вентиляции и климат контроля.

С развитием высокоскоростных поездов, появлением беспроводного интернета и интеллектуальных систем управления, пассажирские вагоны становятся не только местом транспортировки, но и платформой для передовых технологий. Интеграция умных решений, таких как системы безопасности и развлечений, сделала поездки более современными и удобными.

В целом, развитие пассажирских вагонов продолжает стремительно продвигаться вперед, объединяя комфорт, технологии и ответственность перед окружающей средой.

# ЦЕЛИ И ЗАДАЧИ УЧЕБНОЙ ПРАКТИКИ

Разработка программных модулей для системы ведения учета сотрудников ВУЗа направлена на повышение эффективности, гибкости и надежности рабочего процесса отдела кадров. Целью является разработка и проектирование информационной системы, способствующей оптимизации работы и обеспечивающей высокий уровень безопасности и качества ведения учета работников института.

Задачи:

Автоматизация Учета Сотрудников ВУЗа: Разработка модулей для автоматизации добавления и редактирования информации о работниках института.

Управление Трудовыми Договорами: Реализация функционала по управлению трудовыми договорами, включая создание договора, изменение его условий и продления.

Обеспечение Безопасности Данных: Разработка модулей для защиты конфиденциальности данных, связанных с личной информацией сотрудников и условий трудового договора, чтобы предотвратить несанкционированный доступ.

Цель и задачи в совокупности направлены на повышение эффективности учета и обеспечение безопасности данных сотрудников института через современные информационные технологии.

## Основные требования по технике безопасности во время прохождения учебной практики

К работе в компьютерном классе допускаются лица, ознакомленные с данной инструкцией по технике безопасности и правилам поведения. Необходимо неукоснительно соблюдать правила по технике безопасности, т.к. нарушение этих правил может привести к поражению электрическим током, вызвать возгорание и навредить вашему здоровью.

* Работа обучающегося в компьютерном классе разрешается только в присутствии преподавателя (лаборанта).
* Во время занятий посторонние лица могут находиться в классе только с разрешения преподавателя.
* Во время перемен между уроками проводится обязательное проветривание компьютерного кабинета с обязательным выходом обучающегося из класса.

Каждый обучающийся в ответе за состояние своего рабочего места и сохранность размещенного на нем оборудования.

Обучающегося, допустившего нарушение по технике безопасности и правилам поведения в компьютерном классе, может быть привлечен к дисциплинарной ответственности. Если нарушение техники безопасности связано с причинением имущественного ущерба, он несет и материальную ответственность в установленном законом порядке.

## Нормативно-правовые документы

Оформление и содержание технического задания должно соответствовать требованиям стандарта «ГОСТ 19.201-78. ЕСПД. Общие положения.

Техническое задание оформляют в соответствии с ГОСТ 19.106-78 на листах формата 11 и 12 по ГОСТ 2.301-68, как правило, без заполнения полей листа. Номера листов (страниц) проставляются в верхней части листа над текстом.

Лист утверждения и титульный лист оформляют в соответствии с ГОСТ 19.104-78.

Информационную часть (аннотацию и содержание), лист регистрации изменений допускается в документ не включать.

Для внесения изменений или дополнений в техническое задание на последующих стадиях разработки про1раммы или программного изделия выпускают дополнение к нему.

Согласование и утверждение дополнения к техническому заданию проводят в том же порядке, который установлен для технического задания.

Техническое задание должно содержать следующие разделы:

* введение;
* основания для разработки;
* назначение разработки;
* требования к программе или программному изделию;
* требования к программной документации;
* технико-экономические показатели;
* стадии и этапы разработки;
* порядок контроля и приемки;
* в техническое задание допускается включать приложения.

В зависимости от особенностей программы или программного изделия допускается уточнять содержание разделов, вводить новые разделы или объединять отдельные из них.

Вывод по главе: были описаны, техника безопасности во время прохождения учебной практики; нормативно-правовые документы, которые отвечают за стандарты разработки; цели и задачи учебной практики, включая вводную часть, где описана тематика отрасли и её тенденции развития.

# ПРЕДМЕТНАЯ ОБЛАСТЬ. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ. ПОСТАНОВКА ЗАДАЧИ

Предметная область информационной системы — это материальная система или система, характеризующая элементы материального мира, информация о которой хранится и обрабатывается. Предметная область рассматривается как некоторая совокупность реальных объектов и связей между ними.

## Описание предметной области

Необходимо разработать прикладное программное обеспечение деятельности отдела кадров университета.

В отделе кадров университета находятся данные всех сотрудников: от преподавателя до ректора, и их трудовой деятельности. Наряду с такими данными, как специальность сотрудника и занимаемая должность, обязательно учитываются сведения об ученой степени сотрудника (кандидат наук, доктор) и ученом звании (доцент, профессор). Также в отделе кадров хранится информация о трудовой деятельности сотрудника: о предыдущих местах работы, сроке работы и предприятии. Отдел кадров занимается подготовкой трудовых договоров с преподавателями после избрания их по конкурсу на очередной срок. Также в его ведении находятся сведения о наложении взысканий на сотрудников и их поощрениях. Взыскания в трудовую книжку не заносятся, а хранятся в электронном виде.

Информационная система "Мои Сотрудники" призвана содействовать повышению эффективности и безопасности в процессе учета сотрудников, а также удобной подготовки трудовых договоров и добавлении взысканий.

## Группы пользователей информационной системы

Сотрудники, в зависимости от своих функциональных обязанностей, могут выполнять действия, направленные на мониторинг, управление, анализ данных и обеспечение безопасности в рамках проектируемой информационной системы "Мои Сотрудники".

Роли и их Обязанности:

* + Администратор: управление системами аутентификации, авторизации и шифрования; оптимизация работы системы для обеспечения высокой производительности; полный доступ ко всей системе.
  + Начальник: получение и редактирование информации о сотрудниках, их образовании и рабочем месте, создание трудовых договоров и добавление взысканий.
  + Рекрутер: получение информации о сотрудниках, их образовании, рабочем месте, трудовых договорах и взысканий. Проводит анализ данных, хранимых в информационной системе.

## Основные требования, предъявляемые к информационной системе

* + Гибкость: система должна обладать высокой степенью редактирования, чтобы пользователи могли адаптировать её под различные потребности в рамках своих обязанностей; возможность легкого масштабирования системы при увеличении объема данных или числа пользователей.
  + Надежность: обеспечение бесперебойной работы системы с минимальным количеством сбоев и перерывов; регулярное создание резервных копий данных для предотвращения потери информации в случае сбоев.
  + Эффективность: обеспечение высокой производительности при обработке данных, особенно в условиях одновременного использования множества пользователей; эффективное использование вычислительных ресурсов, чтобы система функционировала быстро и эффективно.
  + Безопасность: реализация надежной системы аутентификации для проверки личности пользователей и установление строгих прав доступа; и использование шифрования для защиты передаваемой и хранимой чувствительной информации.

Вывод по главе: было описано, что такое предметная область и её описание; группы пользователей, которые будут пользоваться информационной системой и основные системные требования к ней.

# ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ «МОИ СОТРУДНИКИ»

Проектированием информационных систем называется многоступенчатый процесс их создания и/или модернизации путём применения упорядоченной совокупности методологий и инструментария. Проектирование (в отличие от моделирования) предполагает работу с пока несуществующим объектом и направлено на создание информационной системы.

## Моделирование некоторых динамических аспектов системы

Диаграммы деятельности применяются для моделирования динамических аспектов поведения системы. Эти динамические аспекты могут включать деятельность на любом уровне абстракции в любом представлении системной архитектуры, включая классы, интерфейсы, компоненты и узлы.

При разработке программного обеспечения заблаговременное планирование и моделирование значительно упрощают программирование. На этапе создания концептуальной модели для описания функционального назначения используется диаграмма вариантов использования, а для описания логики процедур и бизнес-процессов используется диаграмма деятельности.

### Создание диаграммы вариантов использования

В приложении А на рисунке 1 представлен пример диаграммы вариантов использования информационной системы «Мои Сотрудники».

### Создание диаграммы деятельности

В приложении А на рисунках 2 – 4 представлена диаграмма деятельности (по ролям) информационной системы «Мои Сотрудники».

## Разработка функциональной структуры информационной системы. Карта навигации по информационной системе

Карта переходов по информационной системе «Мои Сотрудники» представлена в приложении А на рисунке 5.

## Разработка макета информационной системы «Мои Сотрудники»

В приложении А на рисунке 6 представлено окно авторизации.

На этапе авторизации пользователь может ввести логин и пароль, чтобы продолжить работу в информационной системе.

После того, как пользователь успешно прошел авторизацию, он может продолжить работу с таблицами на главной форме. В приложении А на рисунках 7 - 9 представлена главная форма, которая будет отображена, в зависимости от прав пользователя в системе. Например, для рекрутера будут скрыты кнопки сохранения, добавления и удаления записи, так как данной группе пользователей доступен только просмотр данных.

Дизайн выполнен в матовом стиле, были использованы мягкие цвета для создания приятной атмосферы. Все кнопки, находящиеся на боковой и верхней панели представлены в виде иконок чтобы приложение было интуитивно понятным. При переходе на другие окна редактирования, в старых остается информация, что делает программу многозадачной и гибкой в использовании.

Вывод по главе: была спроектирована диаграмма вариантов использования, диаграмма деятельности и макеты будущей информационной системы.

# РАЗРАБОТКА ПРОТОТИПА ИНФОРМАЦИОННОЙ СИСТЕМЫ «МОИ СОТРУДНИКИ»

Прототип — это один из этапов разработки, который заключается в продумывании содержания и расположения важных элементов интерфейса. Прототип — это моделирование конечного продукта. Это интерактивный макет, который может иметь любую степень точности. Основная цель создания прототипов – проверить, насколько последователен путь пользователя, и выявить препятствия, которые могут возникнуть в процессе его взаимодействия с продуктом.

## Моделирование и разработка базы данных

Исходя из анализа предметной области, можно выделить восемь сущностей: «Трудовой Договор», «Сотрудники», «Образование Сотрудников», «Личная Информация Сотрудников», «Информация О Предыдущем Рабочем Месте», «Информация О Рабочем Месте», «Дисциплинарные Меры В Отношении Сотрудников», «Пользователи».

В приложении А на рисунке 10 представлена логическая модель данных базы данных «Мои сотрудники».

Создание базы данных происходит в MS SQL, в примере представлено создание базы данных в MS SQL Server.

С помощью языка SQL созданы все таблицы и ограничения, представленные на логической модели данных.

Код SQL представлен в приложении Б в листинге кода создания таблиц в MS SQL Server.

В приложении А на рисунке 11 представлены все созданные таблицы в обозревателе объектов базы данных.

В приложении А на рисунке 12 представлена физическая модель данных, отражающая все таблицы и их поля, а также связи между таблицами.

## Разработка программных модулей информационной системы «Мои Сотрудники»

Разработанные в MS Visual Studio формы продемонстрированы в приложении А на рисунках 13 - 16.

Свойства элементов формы Авторизация представлены в таблице 1.

Таблица 1 – Свойства элементов формы «Авторизация»

| Объект | Свойство | Значение |
| --- | --- | --- |
| Авторизация | FormBorderStyle | FixedSingle |
| Text | Авторизация |
| StartPosition | CenterScreen |
| Size | 381; 493 |
| Icon | (Значок) |
| BackColor | 192; 192; 255 |

Событийно-управляемые процедуры в форме Авторизация представлены в таблице 2

Таблица 2 – Событийно-управляемые процедуры в форме «Авторизация»

| Объект | Событийно-управляемые процедуры |
| --- | --- |
| Класс Authorization | public partial class Authorization : Form  {  public Authorization()  {  InitializeComponent();  }  } |
| Кнопка button1 | private void button1\_Click(object sender, EventArgs e)  {  if ((textBox1.Text == "") || (textBox2.Text == ""))  {  MessageBox.Show("Поля не должны быть пустыми.");  }  else  {  SqlConnection sqlConnect = new SqlConnection("Data Source = ANTON\_SERVER\\SQLEXPRESS; Initial Catalog = УП01\_ИСПП\_5\_Гаевский\_АА; Integrated Security = True");  sqlConnect.Open();  SqlDataAdapter da = new SqlDataAdapter("select \* from Пользователи", sqlConnect);  DataTable dt = new DataTable();  da.Fill(dt);  int flag = -1;  for (int i = 0; i < dt.Rows.Count; i++)  {  if ((dt.Rows[i]["ПраваДоступа"].ToString() == "Администратор") && (dt.Rows[i]["Логин"].ToString() == textBox1.Text) && (dt.Rows[i]["Пароль"].ToString() == textBox2.Text))  flag = 0;  if ((dt.Rows[i]["ПраваДоступа"].ToString() == "Начальник") && (dt.Rows[i]["Логин"].ToString() == textBox1.Text) && (dt.Rows[i]["Пароль"].ToString() == textBox2.Text))  {  flag = 1;  }  if ((dt.Rows[i]["ПраваДоступа"].ToString() == "Рекрутер") && (dt.Rows[i]["Логин"].ToString() == textBox1.Text) && (dt.Rows[i]["Пароль"].ToString() == textBox2.Text))  {  flag = 2;  }  }  if (flag != -1)  {  if (flag == 0)  {  Hide();  Administrator UserForm1 = new Administrator();  UserForm1.Show();  }  if (flag == 1)  {  Hide();  Chief UserForm1 = new Chief();  UserForm1.Show();  }  if (flag == 2)  {  Hide();  Recruter UserForm1 = new Recruter();  UserForm1.Show();  }  }  else  MessageBox.Show("Неправильный логин или пароль.");  }  } |

Свойства элементов формы Администратор представлены в таблице 3.

Таблица 3 – Свойства элементов формы Администратор

|  |  |  |
| --- | --- | --- |
| Объект | Свойство | Значение |
| Администратор | BackColor | 192; 192; 255 |
| FormBorderStyle | None |
| Icon | (Значок) |
| MaximizeBox | False |
| MinimizeBox | False |
| StartPosition | CenterScreen |
| Size | 1270; 1114 |
| Text | Мои сотрудники |

Событийно-управляемые процедуры в форме Администратор представлены в таблице 4.

Таблица 4 – Событийно-управляемые процедуры в форме «Администратор»

| Объект | Событийно-управляемые процедуры |
| --- | --- |
| Класс Administrator | public partial class Administrator : Form  {  public Administrator()  {  InitializeComponent();  }  private SqlConnection sqlConnection = null;  private SqlDataAdapter sqlDataAdapter = null;  private OpenFileDialog openFileDialog = new OpenFileDialog();  private int x, y;  private int indexOfPanel = 7;  private int indexOfDataGridView = 0;  private string imagePath = null;  private string nameOfTable = null;  private const int widthFlowLayoutPanel = 95;  private const int heightFlowLayoutPanel = 60;  } |
| Событие Administrator\_Load | private void Administrator\_Load(object sender, EventArgs e)  {  sqlConnection = new SqlConnection(ConfigurationManager.ConnectionStrings["Отдел\_кадров\_университета.Properties.Settings.УП01\_ИСПП\_5\_Гаевский\_ААConnectionString"].ConnectionString);  sqlConnection.Open();  OpenPanels.openPanels(tableLayoutPanel1, indexOfPanel);  dataGridView3.Click += HideFlowLayoutPanel;  panel1.Click += HideFlowLayoutPanel;  panel2.Click += HideFlowLayoutPanel;  panel3.Click += HideFlowLayoutPanel;  panel4.Click += HideFlowLayoutPanel;  panel5.Click += HideFlowLayoutPanel;  panel6.Click += HideFlowLayoutPanel;  panel7.Click += HideFlowLayoutPanel;  panel8.Click += HideFlowLayoutPanel;  panel9.Click += HideFlowLayoutPanel;  dataGridView3.ColumnHeadersDefaultCellStyle.Font = new Font("Microsoft Sans Serif", 12);  dataGridView3.DefaultCellStyle.Font = new Font("Microsoft Sans Serif", 11);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП\_5\_Гаевский\_ААDataSet.Пользователи". При необходимости она может быть перемещена или удалена.  this.пользователиTableAdapter.Fill(this.уП01\_ИСПП\_5\_Гаевский\_ААDataSet.Пользователи);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ОбразованиеСотрудников". При необходимости она может быть перемещена или удалена.  this.образованиеСотрудниковTableAdapter.Fill(this.уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ОбразованиеСотрудников);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ДисциплинарныеМерыВОтношенииСотрудников". При необходимости она может быть перемещена или удалена.  this.дисциплинарныеМерыВОтношенииСотрудниковTableAdapter.Fill(this.уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ДисциплинарныеМерыВОтношенииСотрудников);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ИнформацияОПредыдущемРабочемМесте". При необходимости она может быть перемещена или удалена.  this.информацияОПредыдущемРабочемМестеTableAdapter.Fill(this.уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ИнформацияОПредыдущемРабочемМесте);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ИнформацияОРабочемМестеСотрудников". При необходимости она может быть перемещена или удалена.  this.информацияОРабочемМестеСотрудниковTableAdapter.Fill(this.уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ИнформацияОРабочемМестеСотрудников);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ЛичнаяИнформацияСотрудников". При необходимости она может быть перемещена или удалена.  this.личнаяИнформацияСотрудниковTableAdapter.Fill(this.уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ЛичнаяИнформацияСотрудников);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП\_5\_Гаевский\_ААDataSet.Сотрудники". При необходимости она может быть перемещена или удалена.  this.сотрудникиTableAdapter.Fill(this.уП01\_ИСПП\_5\_Гаевский\_ААDataSet.Сотрудники);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ТрудовойДоговор". При необходимости она может быть перемещена или удалена.  this.трудовойДоговорTableAdapter.Fill(this.уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ТрудовойДоговор);  } |
| Кнопка button6\_click | private void button6\_Click(object sender, EventArgs e)  {  flowLayoutPanel1.Width = widthFlowLayoutPanel;  flowLayoutPanel1.Height = heightFlowLayoutPanel;  x = button6.Location.X + button6.Size.Width;  y = button6.Location.Y;  flowLayoutPanel1.Controls.Clear();  flowLayoutPanel1.Location = new Point(x, y);  flowLayoutPanel1.Width = 200;  flowLayoutPanel1.BackColor = Color.FromArgb(192, 192, 255);  flowLayoutPanel1.Visible = true;  Button buttonТрудовыеДоговоры = new Button();  buttonТрудовыеДоговоры.Text = "Трудовые договоры";  buttonТрудовыеДоговоры.Width = flowLayoutPanel1.Width - 5;  buttonТрудовыеДоговоры.Click += buttonТрудовыеДоговоры\_Click;  flowLayoutPanel1.Controls.Add(buttonТрудовыеДоговоры);  Button buttonСоздатьТрудовойДоговор = new Button();  buttonСоздатьТрудовойДоговор.Text = "Добавить трудовой договор";  buttonСоздатьТрудовойДоговор.Width = flowLayoutPanel1.Width - 5;  buttonСоздатьТрудовойДоговор.Click += buttonСоздатьТрудовойДоговор\_Click;  flowLayoutPanel1.Controls.Add(buttonСоздатьТрудовойДоговор);  } |
| Кнопка buttonСоздатьТрудовойДоговор\_Click | private void buttonСоздатьТрудовойДоговор\_Click(object sender, EventArgs e)  {  button10.Enabled = true;  button27.Enabled = true;  button17.Enabled = true;  indexOfPanel = 0;  OpenPanels.openPanels(tableLayoutPanel1, indexOfPanel);  } |
| Кнопка buttonТрудовыеДоговоры\_Click | private void buttonТрудовыеДоговоры\_Click(object sender, EventArgs e)  {  button10.Enabled = true;  button27.Enabled = true;  button17.Enabled = true;  indexOfPanel = 7;  indexOfDataGridView = 0;  nameOfTable = "ТрудовойДоговор";  OpenPanels.openPanels(tableLayoutPanel1, indexOfPanel);  dataGridView3.DataSource = трудовойДоговорbindingSource;  dataGridView3.Columns["КодТрудовогоДоговора"].HeaderText = "Код трудового договора";  dataGridView3.Columns["РегистрационныйНомерСотрудника"].HeaderText = "Регистрационный номер сотрудника";  dataGridView3.Columns["КодРабочегоМеста"].HeaderText = "Код рабочего места";  dataGridView3.Columns["КоличествоРабочихСроков"].HeaderText = "Количество рабочих сроков";  dataGridView3.Columns["ДатаНачалаТрудовойДеятельности"].HeaderText = "Дата начала трудовой деятельности";  dataGridView3.Columns["ДатаОкончанияТрудовойДеятельности"].HeaderText = "Дата окончания трудовой деятельности";  foreach (DataGridViewColumn column in dataGridView3.Columns)  {  column.AutoSizeMode = DataGridViewAutoSizeColumnMode.ColumnHeader;  }  typeof(DataGridView).InvokeMember("DoubleBuffered", BindingFlags.NonPublic | BindingFlags.Instance | BindingFlags.SetProperty, null, dataGridView3, new object[] { true });  dataGridView3.Scroll += (scrollSender, scrollEventArgs) =>  {  if (scrollEventArgs.ScrollOrientation == ScrollOrientation.VerticalScroll)  {  dataGridView3.FirstDisplayedScrollingRowIndex = scrollEventArgs.NewValue;  }  else if (scrollEventArgs.ScrollOrientation == ScrollOrientation.HorizontalScroll)  {  dataGridView3.FirstDisplayedScrollingColumnIndex = scrollEventArgs.NewValue;  }  };  } |
| Кнопка button2\_Click | private void button2\_Click(object sender, EventArgs e)  {  flowLayoutPanel1.Width = widthFlowLayoutPanel;  flowLayoutPanel1.Height = heightFlowLayoutPanel;  x = button2.Location.X + button2.Size.Width;  y = button2.Location.Y;  flowLayoutPanel1.Controls.Clear();  flowLayoutPanel1.Location = new Point(x, y);  flowLayoutPanel1.Width = 150;  flowLayoutPanel1.BackColor = Color.FromArgb(192, 192, 255);  flowLayoutPanel1.Visible = true;  Button buttonСотрудники = new Button();  buttonСотрудники.Text = "Сотрудники";  buttonСотрудники.Width = flowLayoutPanel1.Width - 5;  buttonСотрудники.BackColor = button2.BackColor;  buttonСотрудники.Click += buttonСотрудники\_Click;  flowLayoutPanel1.Controls.Add(buttonСотрудники);  Button buttonСоздатьСотрудника = new Button();  buttonСоздатьСотрудника.Text = "Добавить сотрудника";  buttonСоздатьСотрудника.Width = flowLayoutPanel1.Width - 5;  buttonСотрудники.BackColor = button2.BackColor;  buttonСоздатьСотрудника.Click += buttonСоздатьСотрудника\_Click;  flowLayoutPanel1.Controls.Add(buttonСоздатьСотрудника);  } |
| Кнопка buttonСоздатьСотрудника\_Click | private void buttonСоздатьСотрудника\_Click(object sender, EventArgs e)  {  button10.Enabled = true;  button27.Enabled = true;  button17.Enabled = true;  indexOfPanel = 1;  OpenPanels.openPanels(tableLayoutPanel1, indexOfPanel);  } |
| Кнопка buttonСотрудники\_Click | private void buttonСотрудники\_Click(object sender, EventArgs e)  {  button10.Enabled = true;  button27.Enabled = true;  button17.Enabled = true;  indexOfPanel = 7;  indexOfDataGridView = 1;  nameOfTable = "Сотрудники";  OpenPanels.openPanels(tableLayoutPanel1, indexOfPanel);  dataGridView3.DataSource = сотрудникиbindingSource;  int imageColumnIndex = 6;  int desiredColumnWidth = 200;  int desiredColumnHeight = 200;  foreach (DataGridViewColumn column in dataGridView3.Columns)  {  if (column.Index != imageColumnIndex)  {  column.AutoSizeMode = DataGridViewAutoSizeColumnMode.ColumnHeader;  }  }  dataGridView3.Columns[imageColumnIndex].Width = desiredColumnWidth;  foreach (DataGridViewRow row in dataGridView3.Rows)  {  if (row.Cells[imageColumnIndex].Value is Image image)  {  Image resizedImage = new Bitmap(image, desiredColumnWidth, desiredColumnHeight);  row.Cells[imageColumnIndex].Value = resizedImage;  }  row.Height = desiredColumnHeight + dataGridView3.RowTemplate.Height - dataGridView3.DefaultCellStyle.Padding.Vertical;  }  dataGridView3.Columns["РегистрационныйНомерСотрудника"].HeaderText = "Регистрационный номер сотрудника";  dataGridView3.Columns["Дата\_рождения"].HeaderText = "Дата рождения";  dataGridView3.Columns["Место\_рождения"].HeaderText = "Место рождения";  } |
| Кнопка button3\_Click | private void button3\_Click(object sender, EventArgs e)  {  flowLayoutPanel1.Width = widthFlowLayoutPanel;  flowLayoutPanel1.Height = heightFlowLayoutPanel;  x = button3.Location.X + button3.Size.Width;  y = button3.Location.Y;  flowLayoutPanel1.Controls.Clear();  flowLayoutPanel1.Location = new Point(x, y);  flowLayoutPanel1.Width = 200;  flowLayoutPanel1.BackColor = Color.FromArgb(192, 192, 255);  flowLayoutPanel1.Visible = true;  Button buttonЛичнаяИнформацияСотрудников = new Button();  buttonЛичнаяИнформацияСотрудников.Text = "Личная ифнормация сотрудников";  buttonЛичнаяИнформацияСотрудников.Width = flowLayoutPanel1.Width - 5;  buttonЛичнаяИнформацияСотрудников.Click += buttonЛичнаяИнформацияСотрудников\_Click;  flowLayoutPanel1.Controls.Add(buttonЛичнаяИнформацияСотрудников);  Button buttonСоздатьЛичнуюИнформациюСотрудника = new Button();  buttonСоздатьЛичнуюИнформациюСотрудника.Text = "Добавить личную информацию";  buttonСоздатьЛичнуюИнформациюСотрудника.Width = flowLayoutPanel1.Width - 5;  buttonСоздатьЛичнуюИнформациюСотрудника.Click += buttonСоздатьЛичнуюИнформациюСотрудника\_Click;  flowLayoutPanel1.Controls.Add(buttonСоздатьЛичнуюИнформациюСотрудника);  } |
| Кнопка buttonСоздатьЛичнуюИнформациюСотрудника\_Click | private void buttonСоздатьЛичнуюИнформациюСотрудника\_Click(object sender, EventArgs e)  {  button10.Enabled = true;  button27.Enabled = true;  button17.Enabled = true;  indexOfPanel = 2;  OpenPanels.openPanels(tableLayoutPanel1, indexOfPanel);  } |
| Кнопка buttonЛичнаяИнформацияСотрудников\_Click | private void buttonЛичнаяИнформацияСотрудников\_Click(object sender, EventArgs e)  {  button10.Enabled = true;  button27.Enabled = true;  button17.Enabled = true;  indexOfPanel = 7;  indexOfDataGridView = 2;  nameOfTable = "ЛичнаяИнформацияСотрудников";  OpenPanels.openPanels(tableLayoutPanel1, indexOfPanel);  dataGridView3.DataSource = личнаяИнформацияСотрудниковbindingSource;  dataGridView3.Columns["КодЛичнойИнформацииСотрудника"].HeaderText = "Код личной информации сотрудника";  dataGridView3.Columns["РегистрационныйНомерСотрудника"].HeaderText = "Регистрационный номер сотрудника";  dataGridView3.Columns["ДатаВыдачиПаспорта"].HeaderText = "Дата выдачи паспорта";  dataGridView3.Columns["КемВыдан"].HeaderText = "Кем выдан";  dataGridView3.Columns["ДомашнийАдрес"].HeaderText = "Домашний адрес";  dataGridView3.Columns["ДомашнийТелефон"].HeaderText = "Домашний телефон";  dataGridView3.Columns["АдресЭлектроннойПочты"].HeaderText = "Адрес электронной почты";  foreach (DataGridViewColumn column in dataGridView3.Columns)  {  column.AutoSizeMode = DataGridViewAutoSizeColumnMode.ColumnHeader;  }  } |
| Кнопка button4\_Click | private void button4\_Click(object sender, EventArgs e)  {  flowLayoutPanel1.Width = widthFlowLayoutPanel;  flowLayoutPanel1.Height = heightFlowLayoutPanel;  x = button4.Location.X + button4.Size.Width;  y = button4.Location.Y;  flowLayoutPanel1.Controls.Clear();  flowLayoutPanel1.Location = new Point(x, y);  flowLayoutPanel1.Width = 200;  flowLayoutPanel1.Height = 120;  flowLayoutPanel1.BackColor = Color.FromArgb(192, 192, 255);  flowLayoutPanel1.Visible = true;  Button buttonРабочееМестоСотрудника = new Button();  buttonРабочееМестоСотрудника.Text = "Рабочие места";  buttonРабочееМестоСотрудника.Width = flowLayoutPanel1.Width - 5;  buttonРабочееМестоСотрудника.Click += buttonРабочееМестоСотрудника\_Click;  flowLayoutPanel1.Controls.Add(buttonРабочееМестоСотрудника);  Button buttonБывшееРабочееМестоСотрудника = new Button();  buttonБывшееРабочееМестоСотрудника.Text = "Предыдущие рабочие места";  buttonБывшееРабочееМестоСотрудника.Width = flowLayoutPanel1.Width - 5;  buttonБывшееРабочееМестоСотрудника.Click += buttonБывшееРабочееМестоСотрудника\_Click;  flowLayoutPanel1.Controls.Add(buttonБывшееРабочееМестоСотрудника);  Button buttonДобавитьРабочееМесто = new Button();  buttonДобавитьРабочееМесто.Text = "Добавить рабочее место";  buttonДобавитьРабочееМесто.Width = flowLayoutPanel1.Width - 5;  buttonДобавитьРабочееМесто.Click += buttonДобавитьРабочееМесто\_Click;  flowLayoutPanel1.Controls.Add(buttonДобавитьРабочееМесто);  Button buttonДобавитьБывшееРабочееМесто = new Button();  buttonДобавитьБывшееРабочееМесто.Text = "Добавить предыдущее место";  buttonДобавитьБывшееРабочееМесто.Width = flowLayoutPanel1.Width - 5;  buttonДобавитьБывшееРабочееМесто.Click += buttonДобавитьБывшееРабочееМесто\_Click;  flowLayoutPanel1.Controls.Add(buttonДобавитьБывшееРабочееМесто);  } |
| Кнопка buttonДобавитьБывшееРабочееМесто\_Click | private void buttonДобавитьБывшееРабочееМесто\_Click(object sender, EventArgs e)  {  button10.Enabled = true;  button27.Enabled = true;  button17.Enabled = true;  indexOfPanel = 4;  OpenPanels.openPanels(tableLayoutPanel1, indexOfPanel);  } |
| Кнопка buttonДобавитьРабочееМесто\_Click | private void buttonДобавитьРабочееМесто\_Click(object sender, EventArgs e)  {  button10.Enabled = true;  button27.Enabled = true;  button17.Enabled = true;  indexOfPanel = 3;  OpenPanels.openPanels(tableLayoutPanel1, indexOfPanel);  } |
| Кнопка buttonБывшееРабочееМестоСотрудника\_Click | private void buttonБывшееРабочееМестоСотрудника\_Click(object sender, EventArgs e)  {  button10.Enabled = true;  button27.Enabled = true;  button17.Enabled = true;  indexOfPanel = 7;  indexOfDataGridView = 3;  nameOfTable = "ИнформацияОПредыдущемРабочемМесте";  OpenPanels.openPanels(tableLayoutPanel1, indexOfPanel);  dataGridView3.DataSource = информацияОПредыдущемРабочемМестеbindingSource;  dataGridView3.Columns["КодПредыдущегоРабочегоМеста"].HeaderText = "Код предыдущего рабочего места";  dataGridView3.Columns["РегистрационныйНомерСотрудника"].HeaderText = "Регистрационный номер сотрудника";  dataGridView3.Columns["НазваниеПредприятия"].HeaderText = "Название предприятия";  dataGridView3.Columns["ДатаУстройства"].HeaderText = "Дата устройства";  dataGridView3.Columns["ДатаУвольнения"].HeaderText = "Дата увольнения";  dataGridView3.Columns["ПричинаУвольнения"].HeaderText = "Причина увольнения";  dataGridView3.Columns["АдресПредприятия"].HeaderText = "Адрес предприятия";  dataGridView3.Columns["ТелефонПредприятия"].HeaderText = "Телефон предприятия";  foreach (DataGridViewColumn column in dataGridView3.Columns)  {  column.AutoSizeMode = DataGridViewAutoSizeColumnMode.ColumnHeader;  }  } |
| Кнопка buttonРабочееМестоСотрудника\_Click | private void buttonРабочееМестоСотрудника\_Click(object sender, EventArgs e)  {  button10.Enabled = true;  button27.Enabled = true;  button17.Enabled = true;  indexOfPanel = 7;  indexOfDataGridView = 4;  nameOfTable = "ИнформацияОРабочемМестеСотрудников";  OpenPanels.openPanels(tableLayoutPanel1, indexOfPanel);  dataGridView3.DataSource = информацияОРабочемМестеСотрудниковbindingSource;  dataGridView3.Columns["КодРабочегоМеста"].HeaderText = "Код рабочего места";  dataGridView3.Columns["РегистрационныйНомерСотрудника"].HeaderText = "Регистрационный номер сотрудника";  dataGridView3.Columns["НазваниеКафедры"].HeaderText = "Название кафедры";  dataGridView3.Columns["ЗанимаемаяДолжность"].HeaderText = "Занимаемая должность";  foreach (DataGridViewColumn column in dataGridView3.Columns)  {  column.AutoSizeMode = DataGridViewAutoSizeColumnMode.ColumnHeader;  }  } |
| Кнопка button5\_Click | private void button5\_Click(object sender, EventArgs e)  {  flowLayoutPanel1.Width = widthFlowLayoutPanel;  flowLayoutPanel1.Height = heightFlowLayoutPanel;  x = button5.Location.X + button5.Size.Width;  y = button5.Location.Y;  flowLayoutPanel1.Controls.Clear();  flowLayoutPanel1.Location = new Point(x, y);  flowLayoutPanel1.Width = 200;  flowLayoutPanel1.BackColor = Color.FromArgb(192, 192, 255);  flowLayoutPanel1.Visible = true;  Button buttonДисциплинарныеМеры = new Button();  buttonДисциплинарныеМеры.Text = "Дисциплинарные меры";  buttonДисциплинарныеМеры.Width = flowLayoutPanel1.Width - 5;  buttonДисциплинарныеМеры.Click += buttonДисциплинарныеМеры\_Click;  flowLayoutPanel1.Controls.Add(buttonДисциплинарныеМеры);  Button buttonДобавитьВзыскание = new Button();  buttonДобавитьВзыскание.Text = "Вознаграждения и поощрения";  buttonДобавитьВзыскание.Width = flowLayoutPanel1.Width - 5;  buttonДобавитьВзыскание.Click += buttonДобавитьВзыскание\_Click;  flowLayoutPanel1.Controls.Add(buttonДобавитьВзыскание);  } |
| Кнопка buttonДобавитьВзыскание\_Click | private void buttonДобавитьВзыскание\_Click(object sender, EventArgs e)  {  button10.Enabled = true;  button27.Enabled = true;  button17.Enabled = true;  indexOfPanel = 5;  OpenPanels.openPanels(tableLayoutPanel1, indexOfPanel);  } |
| Кнопка buttonДисциплинарныеМеры\_Click | private void buttonДисциплинарныеМеры\_Click(object sender, EventArgs e)  {  button10.Enabled = true;  button27.Enabled = true;  button17.Enabled = true;  indexOfPanel = 7;  indexOfDataGridView = 5;  nameOfTable = "ДисциплинарныеМерыВОтношенииСотрудников";  OpenPanels.openPanels(tableLayoutPanel1, indexOfPanel);  dataGridView3.DataSource = дисциплинарныеМерыВОтношенииСотрудниковbindingSource;  dataGridView3.Columns["КодДисциплинарнойМеры"].HeaderText = "Код дисциплинарной меры";  dataGridView3.Columns["РегистрационныйНомерСотрудника"].HeaderText = "Регистрационный номер сотрудника";  dataGridView3.Columns["СведенияОВзысканиях"].HeaderText = "Сведения о взысканиях";  dataGridView3.Columns["СведенияОНаграждениях"].HeaderText = "Сведения о награждениях";  } |
| Кнопка button7\_Click | private void button7\_Click(object sender, EventArgs e)  {  flowLayoutPanel1.Width = widthFlowLayoutPanel;  flowLayoutPanel1.Height = heightFlowLayoutPanel;  x = button7.Location.X + button7.Size.Width;  y = button7.Location.Y;  flowLayoutPanel1.Controls.Clear();  flowLayoutPanel1.Location = new Point(x, y);  flowLayoutPanel1.Width = 200;  flowLayoutPanel1.BackColor = Color.FromArgb(192, 192, 255);  flowLayoutPanel1.Visible = true;  Button buttonОбразованиеСотрудников = new Button();  buttonОбразованиеСотрудников.Text = "Образование сотрудников";  buttonОбразованиеСотрудников.Width = flowLayoutPanel1.Width - 5;  buttonОбразованиеСотрудников.Click += buttonОбразованиеСотрудников\_Click;  flowLayoutPanel1.Controls.Add(buttonОбразованиеСотрудников);  Button buttonДобавитьОбразованиеСотрудника = new Button();  buttonДобавитьОбразованиеСотрудника.Text = "Добавить образование";  buttonДобавитьОбразованиеСотрудника.Width = flowLayoutPanel1.Width - 5;  buttonДобавитьОбразованиеСотрудника.Click += buttonДобавитьОбразованиеСотрудника\_Click;  flowLayoutPanel1.Controls.Add(buttonДобавитьОбразованиеСотрудника);  } |
| Кнопка buttonДобавитьОбразованиеСотрудника\_Click | private void buttonДобавитьОбразованиеСотрудника\_Click(object sender, EventArgs e)  {  button10.Enabled = true;  button27.Enabled = true;  button17.Enabled = true;  indexOfPanel = 6;  OpenPanels.openPanels(tableLayoutPanel1, indexOfPanel);  } |
| Кнопка buttonДобавлениеТрудовогоДоговора\_Click | private void buttonДобавлениеТрудовогоДоговора\_Click(object sender, EventArgs e)  {  try  {  SqlCommand sqlCommand = new SqlCommand("INSERT INTO ТрудовойДоговор(РегистрационныйНомерСотрудника, КодРабочегоМеста, КоличествоРабочихСроков, ДатаНачалаТрудовойДеятельности, ДатаОкончанияТрудовойДеятельности, Оклад) " +  "VALUES (@РегистрационныйНомерСотрудника, @КодРабочегоМеста, @КоличествоРабочихСроков, @ДатаНачалаТрудовойДеятельности, @ДатаОкончанияТрудовойДеятельности, @Оклад)", sqlConnection);  DateTime firstDate = DateTime.Parse(maskedTextBoxДатаНачТрудДеят.Text);  DateTime? secondDate = null;  if (!string.IsNullOrEmpty(maskedTextBoxДатаОкончТрудДеят.Text))  {  string dateInput = maskedTextBoxДатаОкончТрудДеят.Text;  string dateFormat = "dd.MM.yyyy"; // Замените на нужный формат маски  if (DateTime.TryParseExact(dateInput, dateFormat, CultureInfo.InvariantCulture, DateTimeStyles.None, out DateTime parsedDate))  {  secondDate = parsedDate;  }  }  sqlCommand.Parameters.AddWithValue("РегистрационныйНомерСотрудника", maskedTextBoxРНС1.Text);  sqlCommand.Parameters.AddWithValue("КодРабочегоМеста", maskedTextBoxКодРабМеста.Text);  sqlCommand.Parameters.AddWithValue("КоличествоРабочихСроков", maskedTextBoxКолРабСроков.Text);  sqlCommand.Parameters.AddWithValue("ДатаНачалаТрудовойДеятельности", $"{firstDate.Month}/{firstDate.Day}/{firstDate.Year}");  if (secondDate.HasValue)  {  sqlCommand.Parameters.AddWithValue("ДатаОкончанияТрудовойДеятельности", $"{secondDate.Value.Month}/{secondDate.Value.Day}/{secondDate.Value.Year}");  }  else  {  sqlCommand.Parameters.AddWithValue("ДатаОкончанияТрудовойДеятельности", DBNull.Value);  }  sqlCommand.Parameters.AddWithValue("Оклад", maskedTextBoxОклад.Text);  if (string.IsNullOrEmpty(maskedTextBoxРНС1.Text) || string.IsNullOrEmpty(maskedTextBoxКодРабМеста.Text) || string.IsNullOrEmpty(maskedTextBoxКолРабСроков.Text) || string.IsNullOrEmpty(maskedTextBoxОклад.Text))  {  Notification notification = new Notification();  notification.titleLabel.ForeColor = Color.Red;  notification.Title = "Не все поля заполнены";  notification.Show(3000);  }  else  {  sqlCommand.ExecuteNonQuery();  Notification notification = new Notification();  notification.titleLabel.ForeColor = Color.ForestGreen;  notification.Title = "Добавлена новая строка в таблицу 'Трудовой договор'";  notification.Show(3000);  }  }  catch (Exception ex)  {  MessageBox.Show("Ошибка: " + ex.Message);  }  } |
| Кнопка buttonДобавлениеСотрудника\_Click | private void buttonДобавлениеСотрудника\_Click(object sender, EventArgs e)  {  try  {  SqlCommand sqlCommand = new SqlCommand("INSERT INTO Сотрудники (Фамилия, Имя, Отчество, Дата\_рождения, Место\_рождения, Фото, Примечания) " +  "VALUES(@Фамилия, @Имя, @Отчество, @Дата\_рождения, @Место\_рождения, @Фото, @Примечания)", sqlConnection);  DateTime firstDate = DateTime.Parse(maskedTextBoxДатаРождения.Text);  Notification notification = new Notification();  if (!string.IsNullOrEmpty(imagePath))  {  byte[] imageData = File.ReadAllBytes(imagePath);  sqlCommand.Parameters.AddWithValue("Фамилия", maskedTextBoxФамилия.Text);  sqlCommand.Parameters.AddWithValue("Имя", maskedTextBoxИмя.Text);  sqlCommand.Parameters.AddWithValue("Отчество", maskedTextBoxОтчество.Text);  sqlCommand.Parameters.AddWithValue("Дата\_рождения", $"{firstDate.Month}/{firstDate.Day}/{firstDate.Year}");  sqlCommand.Parameters.AddWithValue("Место\_рождения", maskedTextBoxМестоРождения.Text);  sqlCommand.Parameters.Add("@Фото", SqlDbType.VarBinary, -1).Value = imageData;  sqlCommand.Parameters.AddWithValue("Примечания", maskedTextBoxПримечания.Text);  sqlCommand.ExecuteNonQuery();  notification = new Notification();  notification.titleLabel.ForeColor = Color.ForestGreen;  notification.Title = "Добавлена новая строка в таблицу 'Сотрудники'";  notification.Show(3000);  }  else  {  notification = new Notification();  notification.titleLabel.ForeColor = Color.Red;  notification.Title = "Выберите фотографию сотрудника";  notification.Show(3000);  }  }  catch (Exception ex)  {  MessageBox.Show("Ошибка: " + ex.Message);  }  } |
| Кнопка buttonДобавлениеЛичнойИнформацииСотрудника\_Click | private void buttonДобавлениеЛичнойИнформацииСотрудника\_Click(object sender, EventArgs e)  {  try  {  SqlCommand sqlCommand = new SqlCommand("INSERT INTO ЛичнаяИнформацияСотрудников (РегистрационныйНомерСотрудника, Паспорт, ДатаВыдачиПаспорта, КемВыдан, ДомашнийАдрес, ДомашнийТелефон, АдресЭлектроннойПочты) " +  "VALUES(@РегистрационныйНомерСотрудника, @Паспорт, @ДатаВыдачиПаспорта, @КемВыдан, @ДомашнийАдрес, @ДомашнийТелефон, @АдресЭлектроннойПочты)", sqlConnection);  DateTime firstDate = DateTime.Parse(maskedTextBoxДатаВыдачиПаспорта.Text);  sqlCommand.Parameters.AddWithValue("РегистрационныйНомерСотрудника", maskedTextBoxРНС3.Text);  sqlCommand.Parameters.AddWithValue("Паспорт", maskedTextBoxПаспорт.Text);  sqlCommand.Parameters.AddWithValue("ДатаВыдачиПаспорта", $"{firstDate.Month}/{firstDate.Day}/{firstDate.Year}");  sqlCommand.Parameters.AddWithValue("КемВыдан", maskedTextBoxКемВыдан.Text);  sqlCommand.Parameters.AddWithValue("ДомашнийАдрес", maskedTextBoxДомашнийАдрес.Text);  sqlCommand.Parameters.AddWithValue("ДомашнийТелефон", maskedTextBoxДомашнийТелефон.Text);  sqlCommand.Parameters.AddWithValue("АдресЭлектроннойПочты", maskedTextBoxАдресЭлПочты.Text);  sqlCommand.ExecuteNonQuery();  Notification notification = new Notification();  notification.titleLabel.ForeColor = Color.ForestGreen;  notification.Title = "Добавлена новая строка в таблицу 'Личная информация сотрудников'";  notification.Show(3000);  }  catch (Exception ex)  {  MessageBox.Show("Ошибка: " + ex.Message);  }  } |
| Кнопка buttonДобавлениеРабочегоМестаСотрудника\_Click | private void buttonДобавлениеРабочегоМестаСотрудника\_Click(object sender, EventArgs e)  {  try  {  SqlCommand sqlCommand = new SqlCommand("INSERT INTO ИнформацияОРабочемМестеСотрудников(РегистрационныйНомерСотрудника, НазваниеКафедры, ЗанимаемаяДолжность) " +  "VALUES(@РегистрационныйНомерСотрудника, @НазваниеКафедры, @ЗанимаемаяДолжность)", sqlConnection);  sqlCommand.Parameters.AddWithValue("РегистрационныйНомерСотрудника", maskedTextBoxРНС4.Text);  sqlCommand.Parameters.AddWithValue("НазваниеКафедры", maskedTextBoxНазваниеКафедры.Text);  sqlCommand.Parameters.AddWithValue("ЗанимаемаяДолжность", maskedTextBoxЗанимаемаяДолжность.Text);  sqlCommand.ExecuteNonQuery();  Notification notification = new Notification();  notification.titleLabel.ForeColor = Color.ForestGreen;  notification.Title = "Добавлена новая строка в таблицу 'Информация о рабочем месте'";  notification.Show(3000);  }  catch (Exception ex)  {  MessageBox.Show("Ошибка: " + ex.Message);  }  } |
| Кнопка buttonДобавлениеПредыдущегоРабочегоМестаСотрудника\_Click | private void buttonДобавлениеПредыдущегоРабочегоМестаСотрудника\_Click(object sender, EventArgs e)  {  try  {  SqlCommand sqlCommand = new SqlCommand("INSERT INTO ИнформацияОПредыдущемРабочемМесте (РегистрационныйНомерСотрудника, НазваниеПредприятия, Должность, ДатаУстройства, ДатаУвольнения, ПричинаУвольнения, АдресПредприятия, ТелефонПредприятия) " +  "VALUES(@РегистрационныйНомерСотрудника, @НазваниеПредприятия , @Должность, @ДатаУстройства, @ДатаУвольнения, @ПричинаУвольнения, @АдресПредприятия, @ТелефонПредприятия)", sqlConnection);  DateTime firstDate = DateTime.Parse(maskedTextBoxДатаУстройства.Text);  DateTime secondDate = DateTime.Parse(maskedTextBoxДатаУвольнения.Text);  sqlCommand.Parameters.AddWithValue("РегистрационныйНомерСотрудника", maskedTextBoxРНС5.Text);  sqlCommand.Parameters.AddWithValue("НазваниеПредприятия", maskedTextBoxНазваниеПредприятия.Text);  sqlCommand.Parameters.AddWithValue("Должность", maskedTextBoxДолжность.Text);  sqlCommand.Parameters.AddWithValue("ДатаУстройства", $"{firstDate.Month}/{firstDate.Day}/{firstDate.Year}");  sqlCommand.Parameters.AddWithValue("ДатаУвольнения", $"{secondDate.Month}/{secondDate.Day}/{secondDate.Year}");  sqlCommand.Parameters.AddWithValue("ПричинаУвольнения", maskedTextBoxПричинаУвольнения.Text);  sqlCommand.Parameters.AddWithValue("АдресПредприятия", maskedTextBoxАдресПредприятия.Text);  sqlCommand.Parameters.AddWithValue("ТелефонПредприятия", maskedTextBoxТелефонПредприятия.Text);  sqlCommand.ExecuteNonQuery();  Notification notification = new Notification();  notification.titleLabel.ForeColor = Color.ForestGreen;  notification.Title = "Добавлена новая строка в таблицу 'Информация о предыдущем рабочем месте'";  notification.Show(3000);  }  catch (Exception ex)  {  MessageBox.Show("Ошибка: " + ex.Message);  }  } |
| Кнопка buttonДобавлениеВзысканияНаграждения\_Click | private void buttonДобавлениеВзысканияНаграждения\_Click(object sender, EventArgs e)  {  try  {  SqlCommand sqlCommand = new SqlCommand("INSERT INTO ДисциплинарныеМерыВОтношенииСотрудников(РегистрационныйНомерСотрудника, СведенияОВзысканиях, СведенияОНаграждениях) " +  "VALUES(@РегистрационныйНомерСотрудника, @СведенияОВзысканиях, @СведенияОНаграждениях)", sqlConnection);  sqlCommand.Parameters.AddWithValue("РегистрационныйНомерСотрудника", maskedTextBoxРНС6.Text);  sqlCommand.Parameters.AddWithValue("СведенияОВзысканиях", maskedTextBoxСведенияОВзысканиях.Text);  sqlCommand.Parameters.AddWithValue("СведенияОНаграждениях", maskedTextBoxСведенияОНаграждениях.Text);  sqlCommand.ExecuteNonQuery();  Notification notification = new Notification();  notification.titleLabel.ForeColor = Color.ForestGreen;  notification.Title = "Добавлена новая строка в таблицу 'Дисциплинарные меры в отношении сотрудников'";  notification.Show(3000);  }  catch (Exception ex)  {  MessageBox.Show("Ошибка: " + ex.Message);  }  } |
| Событие maskedTextBoxСведенияОВзысканиях\_TextChanged | private void maskedTextBoxСведенияОВзысканиях\_TextChanged(object sender, EventArgs e)  {  if (!string.IsNullOrWhiteSpace(maskedTextBoxСведенияОНаграждениях.Text))  {  maskedTextBoxСведенияОВзысканиях.Text = "";  Notification notification = new Notification();  notification.titleLabel.ForeColor = Color.Red;  notification.Title = "Поле 'Сведения о награждениях' должно быть пустым";  notification.Show(3000);  }  } |
| Событие maskedTextBoxСведенияОНаграждениях\_TextChanged | private void maskedTextBoxСведенияОНаграждениях\_TextChanged(object sender, EventArgs e)  {  if (!string.IsNullOrWhiteSpace(maskedTextBoxСведенияОВзысканиях.Text))  {  maskedTextBoxСведенияОНаграждениях.Text = "";  Notification notification = new Notification();  notification.titleLabel.ForeColor = Color.Red;  notification.Title = "Поле 'Сведения о взысканиях' должно быть пустым";  notification.Show(3000);  }  } |
| Кнопка buttonДобавлениеОбразованияСотрудника\_Click | private void buttonДобавлениеОбразованияСотрудника\_Click(object sender, EventArgs e)  {  try  {  SqlCommand sqlCommand = new SqlCommand("INSERT INTO ОбразованиеСотрудников (РегистрационныйНомерСотрудника, ВУЗ, ГодОкончанияВУЗа, Специальность, УченаяСтепень, УченоеЗвание) " +  "VALUES(@РегистрационныйНомерСотрудника, @ВУЗ, @ГодОкончанияВУЗа, @Специальность, @УченаяСтепень, @УченоеЗвание)", sqlConnection);  DateTime firstDate = DateTime.Parse(maskedTextBoxГодОкончанияВУЗа.Text);  sqlCommand.Parameters.AddWithValue("РегистрационныйНомерСотрудника", maskedTextBoxРНС7.Text);  sqlCommand.Parameters.AddWithValue("ВУЗ", maskedTextBoxВУЗ.Text);  sqlCommand.Parameters.AddWithValue("ГодОкончанияВУЗа", $"{firstDate.Month}/{firstDate.Day}/{firstDate.Year}");  sqlCommand.Parameters.AddWithValue("Специальность", maskedTextBoxСпециальность.Text);  sqlCommand.Parameters.AddWithValue("УченаяСтепень", maskedTextBoxУченаяСтепень.Text);  sqlCommand.Parameters.AddWithValue("УченоеЗвание", maskedTextBoxУченоеЗвание.Text);  sqlCommand.ExecuteNonQuery();  Notification notification = new Notification();  notification.titleLabel.ForeColor = Color.ForestGreen;  notification.Title = "Добавлена новая строка в таблицу 'Образование сотрудников'";  notification.Show(3000);  }  catch (Exception ex)  {  MessageBox.Show("Ошибка: " + ex.Message);  }  } |
| Кнопка button9\_Click | private void button9\_Click(object sender, EventArgs e)  {  Application.Exit();  } |
| Кнопка button11\_Click | private void button11\_Click(object sender, EventArgs e)  {  Hide();  Authorization UserForm = new Authorization();  UserForm.Show();  } |
| Кнопка button13\_Click | private void button13\_Click(object sender, EventArgs e)  {  openFileDialog.Filter = "Изображения (\*.png;\*.jpg)|\*.png;\*.jpg";  openFileDialog.Multiselect = false;  if (openFileDialog.ShowDialog() == DialogResult.OK)  {  imagePath = openFileDialog.FileName;  }  maskedTextBoxФото.Text = imagePath;  } |
| Событие dataGridView3\_DataError | private void dataGridView3\_DataError(object sender, DataGridViewDataErrorEventArgs e)  {  Image defaultImage = Resource1.Noname;  if (e.ColumnIndex == 6 && e.RowIndex >= 0)  {  e.ThrowException = false;  e.Cancel = true;  dataGridView3.Rows[e.RowIndex].Cells[e.ColumnIndex].Value = defaultImage;  }  } |
| Кнопка button12\_Click | private void button12\_Click(object sender, EventArgs e)  {  string searchText = textBox1.Text;  foreach (DataGridViewRow row in dataGridView3.Rows)  {  bool isMatch = false;  foreach (DataGridViewCell cell in row.Cells)  {  if (cell.Value != null && cell.Value.ToString().Contains(searchText))  {  isMatch = true;  break;  }  }  if (isMatch)  {  row.DefaultCellStyle.BackColor = Color.LightBlue;  }  else  {  row.DefaultCellStyle.BackColor = dataGridView3.DefaultCellStyle.BackColor;  }  }  } |
| Событие Administrator\_Click | private void Administrator\_Click(object sender, EventArgs e)  {  if (!flowLayoutPanel1.ClientRectangle.Contains(flowLayoutPanel1.PointToClient(Cursor.Position)))  {  flowLayoutPanel1.Visible = false;  }  } |
| Кнопка button16\_Click | private void button16\_Click(object sender, EventArgs e)  {  foreach (DataGridViewRow row in dataGridView3.Rows)  {  row.DefaultCellStyle.BackColor = dataGridView3.DefaultCellStyle.BackColor;  }  textBox1.Text = string.Empty;  } |
| Кнопка button10\_Click | private void button10\_Click(object sender, EventArgs e)  {  трудовойДоговорTableAdapter.Update(уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ТрудовойДоговор);  сотрудникиTableAdapter.Update(уП01\_ИСПП\_5\_Гаевский\_ААDataSet.Сотрудники);  образованиеСотрудниковTableAdapter.Update(уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ОбразованиеСотрудников);  личнаяИнформацияСотрудниковTableAdapter.Update(уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ЛичнаяИнформацияСотрудников);  информацияОПредыдущемРабочемМестеTableAdapter.Update(уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ИнформацияОПредыдущемРабочемМесте);  дисциплинарныеМерыВОтношенииСотрудниковTableAdapter.Update(уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ДисциплинарныеМерыВОтношенииСотрудников);  информацияОРабочемМестеСотрудниковTableAdapter.Update(уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ИнформацияОРабочемМестеСотрудников);  пользователиTableAdapter.Update(уП01\_ИСПП\_5\_Гаевский\_ААDataSet.Пользователи);  Notification notification = new Notification();  notification.titleLabel.ForeColor = Color.ForestGreen;  notification.Title = "Запись изменена";  notification.Height = 75;  notification.Show(3000);  } |
| Кнопка buttonОбразованиеСотрудников\_Click | private void buttonОбразованиеСотрудников\_Click(object sender, EventArgs e)  {  button10.Enabled = true;  button27.Enabled = true;  button17.Enabled = true;  indexOfPanel = 7;  indexOfDataGridView = 6;  nameOfTable = "ОбразованиеСотрудников";  OpenPanels.openPanels(tableLayoutPanel1, indexOfPanel);  dataGridView3.DataSource = образованиеСотрудниковbindingSource;  dataGridView3.Columns["КодОбразования"].HeaderText = "Код образования";  dataGridView3.Columns["РегистрационныйНомерСотрудника"].HeaderText = "Регистрационный номер сотрудника";  dataGridView3.Columns["ГодОкончанияВУЗа"].HeaderText = "Год окончания ВУЗа";  dataGridView3.Columns["УченаяСтепень"].HeaderText = "Ученая степень";  dataGridView3.Columns["УченоеЗвание"].HeaderText = "Ученое звание";  } |
| Кнопка button8\_Click | private void button8\_Click(object sender, EventArgs e)  {  maskedTextBoxРНС1.Text = null;  maskedTextBoxКодРабМеста.Text = null;  maskedTextBoxКолРабСроков.Text = null;  maskedTextBoxДатаНачТрудДеят.Text = null;  maskedTextBoxДатаОкончТрудДеят.Text = null;  maskedTextBoxОклад.Text = null;  } |
| Кнопка button15\_Click | private void button15\_Click(object sender, EventArgs e)  {  maskedTextBoxФамилия.Text = null;  maskedTextBoxИмя.Text = null;  maskedTextBoxОтчество.Text = null;  maskedTextBoxДатаРождения.Text = null;  maskedTextBoxМестоРождения.Text = null;  maskedTextBoxФото.Text = null;  maskedTextBoxПримечания.Text = null;  } |
| Кнопка button18\_Click | private void button18\_Click(object sender, EventArgs e)  {  maskedTextBoxРНС3.Text = null;  maskedTextBoxПаспорт.Text = null;  maskedTextBoxДатаВыдачиПаспорта.Text = null;  maskedTextBoxКемВыдан.Text = null;  maskedTextBoxДомашнийАдрес.Text = null;  maskedTextBoxДомашнийТелефон.Text = null;  maskedTextBoxАдресЭлПочты.Text = null;  } |
| Кнопка button1\_Click | private void button1\_Click(object sender, EventArgs e)  {  maskedTextBoxРНС4.Text = null;  maskedTextBoxНазваниеКафедры.Text = null;  maskedTextBoxЗанимаемаяДолжность.Text = null;  } |
| Кнопка button14\_Click | private void button14\_Click(object sender, EventArgs e)  {  maskedTextBoxРНС5.Text = null;  maskedTextBoxНазваниеПредприятия.Text = null;  maskedTextBoxДолжность.Text = null;  maskedTextBoxДатаУстройства.Text = null;  maskedTextBoxДатаУвольнения.Text = null;  maskedTextBoxПричинаУвольнения.Text = null;  maskedTextBoxАдресПредприятия.Text = null;  maskedTextBoxТелефонПредприятия.Text = null;  } |
| Кнопка button19\_Click | private void button19\_Click(object sender, EventArgs e)  {  maskedTextBoxРНС6.Text = null;  maskedTextBoxСведенияОВзысканиях.Text = null;  maskedTextBoxСведенияОНаграждениях.Text= null;  } |
| Кнопка button20\_Click | private void button20\_Click(object sender, EventArgs e)  {  maskedTextBoxРНС7.Text = null;  maskedTextBoxВУЗ.Text = null;  maskedTextBoxГодОкончанияВУЗа.Text = null;  maskedTextBoxСпециальность.Text= null;  maskedTextBoxУченаяСтепень.Text = null;  maskedTextBoxУченоеЗвание.Text = null;  } |
| Событие button6\_MouseHover | private void button6\_MouseHover(object sender, EventArgs e)  {  toolTip1.Show("Трудовые договоры", button6);  } |
| Событие button2\_MouseHover | private void button2\_MouseHover(object sender, EventArgs e)  {  toolTip1.Show("Сотрудники", button2);  } |
| Событие button3\_MouseHover | private void button3\_MouseHover(object sender, EventArgs e)  {  toolTip1.Show("Личная информация сотрудников", button3);  } |
| Событие button4\_MouseHover | private void button4\_MouseHover(object sender, EventArgs e)  {  toolTip1.Show("Рабочие места", button4);  } |
| Событие button5\_MouseHover | private void button5\_MouseHover(object sender, EventArgs e)  {  toolTip1.Show("Дисциплинарные меры", button5);  } |
| Событие button7\_MouseHover | private void button7\_MouseHover(object sender, EventArgs e)  {  toolTip1.Show("Образование сотрудников", button7);  } |
| Событие button10\_MouseHover | private void button10\_MouseHover(object sender, EventArgs e)  {  toolTip1.Show("Сохранить таблицу", button10);  } |
| Событие button11\_MouseHover | private void button11\_MouseHover(object sender, EventArgs e)  {  toolTip1.Show("Сменить пользователя", button11);  } |
| Событие button9\_MouseHover | private void button9\_MouseHover(object sender, EventArgs e)  {  toolTip1.Show("Закрыть", button9);  } |
| Кнопка button17\_Click\_1 | private void button17\_Click\_1(object sender, EventArgs e)  {  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП\_5\_Гаевский\_ААDataSet.Пользователи". При необходимости она может быть перемещена или удалена.  this.пользователиTableAdapter.Fill(this.уП01\_ИСПП\_5\_Гаевский\_ААDataSet.Пользователи);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ОбразованиеСотрудников". При необходимости она может быть перемещена или удалена.  this.образованиеСотрудниковTableAdapter.Fill(this.уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ОбразованиеСотрудников);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ДисциплинарныеМерыВОтношенииСотрудников". При необходимости она может быть перемещена или удалена.  this.дисциплинарныеМерыВОтношенииСотрудниковTableAdapter.Fill(this.уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ДисциплинарныеМерыВОтношенииСотрудников);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ИнформацияОПредыдущемРабочемМесте". При необходимости она может быть перемещена или удалена.  this.информацияОПредыдущемРабочемМестеTableAdapter.Fill(this.уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ИнформацияОПредыдущемРабочемМесте);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ИнформацияОРабочемМестеСотрудников". При необходимости она может быть перемещена или удалена.  this.информацияОРабочемМестеСотрудниковTableAdapter.Fill(this.уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ИнформацияОРабочемМестеСотрудников);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ЛичнаяИнформацияСотрудников". При необходимости она может быть перемещена или удалена.  this.личнаяИнформацияСотрудниковTableAdapter.Fill(this.уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ЛичнаяИнформацияСотрудников);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП\_5\_Гаевский\_ААDataSet.Сотрудники". При необходимости она может быть перемещена или удалена.  this.сотрудникиTableAdapter.Fill(this.уП01\_ИСПП\_5\_Гаевский\_ААDataSet.Сотрудники);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ТрудовойДоговор". При необходимости она может быть перемещена или удалена.  this.трудовойДоговорTableAdapter.Fill(this.уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ТрудовойДоговор);  switch (indexOfDataGridView)  {  case 0:  foreach (DataGridViewColumn column in dataGridView3.Columns)  {  column.AutoSizeMode = DataGridViewAutoSizeColumnMode.ColumnHeader;  };  break;    case 1:  int imageColumnIndex = 6;  foreach (DataGridViewColumn column in dataGridView3.Columns)  {  if (column.Index != imageColumnIndex)  {  column.AutoSizeMode = DataGridViewAutoSizeColumnMode.ColumnHeader;  }  }  dataGridView3.Columns[imageColumnIndex].Width = 200;  foreach (DataGridViewRow row in dataGridView3.Rows)  {  if (row.Cells[imageColumnIndex].Value is Image image)  {  Image resizedImage = new Bitmap(image, 200, 200);  row.Cells[imageColumnIndex].Value = resizedImage;  }  row.Height = 200 + dataGridView3.RowTemplate.Height - dataGridView3.DefaultCellStyle.Padding.Vertical;  };  break;  case 2:  foreach (DataGridViewColumn column in dataGridView3.Columns)  {  column.AutoSizeMode = DataGridViewAutoSizeColumnMode.ColumnHeader;  };  break;  case 3:  foreach (DataGridViewColumn column in dataGridView3.Columns)  {  column.AutoSizeMode = DataGridViewAutoSizeColumnMode.ColumnHeader;  };  break;  case 4:  foreach (DataGridViewColumn column in dataGridView3.Columns)  {  column.AutoSizeMode = DataGridViewAutoSizeColumnMode.ColumnHeader;  };  break;  }  } |
| Событие button17\_MouseHover | private void button17\_MouseHover(object sender, EventArgs e)  {  toolTip1.Show("Обновить таблицу", button17);  } |
| Кнопка button21\_Click | private void button21\_Click(object sender, EventArgs e)  {  flowLayoutPanel1.Width = widthFlowLayoutPanel;  flowLayoutPanel1.Height = heightFlowLayoutPanel;  x = button21.Location.X + 110;  y = button21.Location.Y + 85; ;  flowLayoutPanel1.Controls.Clear();  flowLayoutPanel1.Location = new Point(x, y);  flowLayoutPanel1.Width = 280;  flowLayoutPanel1.Height = 170;  flowLayoutPanel1.BackColor = Color.White;  flowLayoutPanel1.Visible = true;  Button buttonСотрудникиРаботающиеВНастоящееВремя = new Button();  buttonСотрудникиРаботающиеВНастоящееВремя.Text = "Сотрудники работающие в настоящее время";  buttonСотрудникиРаботающиеВНастоящееВремя.Size = new Size(100, 50);  buttonСотрудникиРаботающиеВНастоящееВремя.Width = flowLayoutPanel1.Width - 5;  buttonСотрудникиРаботающиеВНастоящееВремя.Click += buttonСотрудникиРаботающиеВНастоящееВремя\_Click;  buttonСотрудникиРаботающиеВНастоящееВремя.BackColor = Color.White;  buttonСотрудникиРаботающиеВНастоящееВремя.ForeColor = Color.DarkSlateBlue;  buttonСотрудникиРаботающиеВНастоящееВремя.FlatStyle = FlatStyle.Flat;  buttonСотрудникиРаботающиеВНастоящееВремя.Font = new Font("Microsoft Sans Serif", 10);  flowLayoutPanel1.Controls.Add(buttonСотрудникиРаботающиеВНастоящееВремя);  Button buttonСотрудникиРаботающиеВпервые = new Button();  buttonСотрудникиРаботающиеВпервые.Text = "Сотрудники без опыта";  buttonСотрудникиРаботающиеВпервые.Size = new Size(100, 50);  buttonСотрудникиРаботающиеВпервые.Width = flowLayoutPanel1.Width - 5;  buttonСотрудникиРаботающиеВпервые.Click += buttonСотрудникиРаботающиеВпервые\_Click;  buttonСотрудникиРаботающиеВпервые.BackColor = Color.White;  buttonСотрудникиРаботающиеВпервые.ForeColor = Color.DarkSlateBlue;  buttonСотрудникиРаботающиеВпервые.FlatStyle = FlatStyle.Flat;  buttonСотрудникиРаботающиеВпервые.Font = new Font("Microsoft Sans Serif", 10);  flowLayoutPanel1.Controls.Add(buttonСотрудникиРаботающиеВпервые);  Button buttonСотрудникиКоличествоПрошлыхРабот = new Button();  buttonСотрудникиКоличествоПрошлыхРабот.Text = "Сотрудники и количество их прошлых работ";  buttonСотрудникиКоличествоПрошлыхРабот.Size = new Size(100, 50);  buttonСотрудникиКоличествоПрошлыхРабот.Width = flowLayoutPanel1.Width - 5;  buttonСотрудникиКоличествоПрошлыхРабот.Click += buttonСотрудникиКоличествоПрошлыхРабот\_Click;  buttonСотрудникиКоличествоПрошлыхРабот.BackColor = Color.White;  buttonСотрудникиКоличествоПрошлыхРабот.ForeColor = Color.DarkSlateBlue;  buttonСотрудникиКоличествоПрошлыхРабот.FlatStyle = FlatStyle.Flat;  buttonСотрудникиКоличествоПрошлыхРабот.Font = new Font("Microsoft Sans Serif", 10);  flowLayoutPanel1.Controls.Add(buttonСотрудникиКоличествоПрошлыхРабот);  } |
| Кнопка buttonСотрудникиКоличествоПрошлыхРабот\_Click | private void buttonСотрудникиКоличествоПрошлыхРабот\_Click(object sender, EventArgs e)  {  sqlDataAdapter = new SqlDataAdapter("SELECT Сотрудники.РегистрационныйНомерСотрудника, CONCAT(Сотрудники.Фамилия, ' ', Сотрудники.Имя, ' ', Сотрудники.Отчество) " +  "AS ФИО\_Сотрудника, COUNT(ИнформацияОПредыдущемРабочемМесте.КодПредыдущегоРабочегоМеста) AS КоличествоПредыдущихМест FROM Сотрудники LEFT JOIN ИнформацияОПредыдущемРабочемМесте " +  "ON Сотрудники.РегистрационныйНомерСотрудника = ИнформацияОПредыдущемРабочемМесте.РегистрационныйНомерСотрудника GROUP BY Сотрудники.РегистрационныйНомерСотрудника, " +  "Сотрудники.Фамилия, Сотрудники.Имя, Сотрудники.Отчество HAVING COUNT(ИнформацияОПредыдущемРабочемМесте.КодПредыдущегоРабочегоМеста) > 0 ORDER BY КоличествоПредыдущихМест " +  "DESC; ", sqlConnection);  DataTable dataTable = new DataTable();  sqlDataAdapter.Fill(dataTable);  dataGridView3.DataSource = dataTable;  dataGridView3.Columns["РегистрационныйНомерСотрудника"].HeaderText = "Регистрационный номер сотрудника";  dataGridView3.Columns["ФИО\_Сотрудника"].HeaderText = "ФИО сотрудника";  dataGridView3.Columns["КоличествоПредыдущихМест"].HeaderText = "Количество предыдущих мест";  } |
| Кнопка buttonСотрудникиРаботающиеВпервые\_Click | private void buttonСотрудникиРаботающиеВпервые\_Click(object sender, EventArgs e)  {  sqlDataAdapter = new SqlDataAdapter("SELECT Сотрудники.РегистрационныйНомерСотрудника AS [Регистрационный номер сотрудника], CONCAT(Сотрудники.Фамилия, ' ', Сотрудники.Имя, ' ', Сотрудники.Отчество) " +  "AS [ФИО сотрудника], Дата\_рождения AS [Дата рождения], Место\_рождения AS [Место рождения] FROM Сотрудники WHERE РегистрационныйНомерСотрудника " +  "NOT IN(SELECT РегистрационныйНомерСотрудника FROM ИнформацияОПредыдущемРабочемМесте)", sqlConnection);  DataTable dataTable = new DataTable();  sqlDataAdapter.Fill(dataTable);  dataGridView3.DataSource = dataTable;  } |
| Кнопка buttonСотрудникиРаботающиеВНастоящееВремя\_Click | private void buttonСотрудникиРаботающиеВНастоящееВремя\_Click(object sender, EventArgs e)  {  sqlDataAdapter = new SqlDataAdapter("SELECT Сотрудники.РегистрационныйНомерСотрудника AS [Регистрационный номер сотрудника], CONCAT(Сотрудники.Фамилия, ' ', Сотрудники.Имя, ' ', " +  "Сотрудники.Отчество) AS [ФИО сотрудника], Сотрудники.Дата\_рождения AS [Дата рождения], Сотрудники.Место\_рождения AS [Место рождения]" +  "FROM Сотрудники " +  "WHERE Сотрудники.РегистрационныйНомерСотрудника NOT IN " +  "(SELECT ТрудовойДоговор.РегистрационныйНомерСотрудника FROM ТрудовойДоговор WHERE ТрудовойДоговор.ДатаОкончанияТрудовойДеятельности IS NOT NULL)", sqlConnection);  DataTable dataTable = new DataTable();  sqlDataAdapter.Fill(dataTable);  dataGridView3.DataSource = dataTable;  } |
| Кнопка button22\_Click | private void button22\_Click(object sender, EventArgs e)  {  flowLayoutPanel1.Width = widthFlowLayoutPanel;  flowLayoutPanel1.Height = heightFlowLayoutPanel;  x = button21.Location.X + 110;  y = button21.Location.Y + 85; ;  flowLayoutPanel1.Controls.Clear();  flowLayoutPanel1.Location = new Point(x, y);  flowLayoutPanel1.Width = 280;  flowLayoutPanel1.Height = 70;  flowLayoutPanel1.BackColor = Color.White;  flowLayoutPanel1.Visible = true;  Button buttonСотрудникиОкладВышеСреднего = new Button();  buttonСотрудникиОкладВышеСреднего.Text = "Сотрудники с окладом выше среднего";  buttonСотрудникиОкладВышеСреднего.Size = new Size(100, 50);  buttonСотрудникиОкладВышеСреднего.Width = flowLayoutPanel1.Width - 5;  buttonСотрудникиОкладВышеСреднего.Click += buttonСотрудникиОкладВышеСреднего\_Click;  buttonСотрудникиОкладВышеСреднего.BackColor = Color.White;  buttonСотрудникиОкладВышеСреднего.ForeColor = Color.DarkOrchid;  buttonСотрудникиОкладВышеСреднего.FlatStyle = FlatStyle.Flat;  buttonСотрудникиОкладВышеСреднего.Font = new Font("Microsoft Sans Serif", 10);  flowLayoutPanel1.Controls.Add(buttonСотрудникиОкладВышеСреднего);  } |
| Кнопка buttonСотрудникиОкладВышеСреднего\_Click | private void buttonСотрудникиОкладВышеСреднего\_Click(object sender, EventArgs e)  {  sqlDataAdapter = new SqlDataAdapter("SELECT Сотрудники.РегистрационныйНомерСотрудника AS [Регистрационный номер сотрудника], CONCAT(Сотрудники.Фамилия, ' ', Сотрудники.Имя, ' ', " +  "Сотрудники.Отчество) AS [ФИО сотрудника], ТрудовойДоговор.Оклад FROM Сотрудники INNER JOIN ТрудовойДоговор ON Сотрудники.РегистрационныйНомерСотрудника = " +  "ТрудовойДоговор.РегистрационныйНомерСотрудника WHERE ТрудовойДоговор.Оклад > (SELECT AVG(Оклад) FROM ТрудовойДоговор); ", sqlConnection);  DataTable dataTable = new DataTable();  sqlDataAdapter.Fill(dataTable);  dataGridView3.DataSource = dataTable;  } |
| Кнопка button23\_Click | private void button23\_Click(object sender, EventArgs e)  {  flowLayoutPanel1.Width = widthFlowLayoutPanel;  flowLayoutPanel1.Height = heightFlowLayoutPanel;  x = button21.Location.X + 110;  y = button21.Location.Y + 85; ;  flowLayoutPanel1.Controls.Clear();  flowLayoutPanel1.Location = new Point(x, y);  flowLayoutPanel1.Width = 280;  flowLayoutPanel1.Height = 70;  flowLayoutPanel1.BackColor = Color.White;  flowLayoutPanel1.Visible = true;  Button buttonСотрудникиКоличествоНаграждений = new Button();  buttonСотрудникиКоличествоНаграждений.Text = "Количество награждений";  buttonСотрудникиКоличествоНаграждений.Size = new Size(100, 50);  buttonСотрудникиКоличествоНаграждений.Width = flowLayoutPanel1.Width - 5;  buttonСотрудникиКоличествоНаграждений.Click += buttonСотрудникиКоличествоНаграждений\_Click;  buttonСотрудникиКоличествоНаграждений.BackColor = Color.White;  buttonСотрудникиКоличествоНаграждений.ForeColor = Color.Indigo;  buttonСотрудникиКоличествоНаграждений.FlatStyle = FlatStyle.Flat;  buttonСотрудникиКоличествоНаграждений.Font = new Font("Microsoft Sans Serif", 10);  flowLayoutPanel1.Controls.Add(buttonСотрудникиКоличествоНаграждений);  } |
| Кнопка buttonСотрудникиКоличествоНаграждений\_Click | private void buttonСотрудникиКоличествоНаграждений\_Click(object sender, EventArgs e)  {  button10.Enabled = false;  button27.Enabled = false;  button17.Enabled = false;  sqlDataAdapter = new SqlDataAdapter("SELECT Сотрудники.РегистрационныйНомерСотрудника AS [Регистрационный номер сотрудника], CONCAT(Сотрудники.Фамилия, ' ', Сотрудники.Имя, ' ', " +  "Сотрудники.Отчество) AS [ФИО сотрудника], ИнформацияОРабочемМестеСотрудников.КодРабочегоМеста AS [Код рабочего места], Сотрудники.Фото, " +  "COUNT(ДисциплинарныеМерыВОтношенииСотрудников.КодДисциплинарнойМеры) " +  "AS [Количество награждений] FROM Сотрудники LEFT JOIN ДисциплинарныеМерыВОтношенииСотрудников ON Сотрудники.РегистрационныйНомерСотрудника = " +  "ДисциплинарныеМерыВОтношенииСотрудников.РегистрационныйНомерСотрудника INNER JOIN ИнформацияОРабочемМестеСотрудников ON Сотрудники.РегистрационныйНомерСотрудника = " +  "ИнформацияОРабочемМестеСотрудников.РегистрационныйНомерСотрудника WHERE ДисциплинарныеМерыВОтношенииСотрудников.СведенияОНаграждениях IS NOT NULL GROUP BY " +  "Сотрудники.РегистрационныйНомерСотрудника, Сотрудники.Фамилия, Сотрудники.Имя, Сотрудники.Отчество, Сотрудники.Фото, ИнформацияОРабочемМестеСотрудников.КодРабочегоМеста; ", sqlConnection);  DataTable dataTable = new DataTable();  sqlDataAdapter.Fill(dataTable);  dataGridView3.DataSource = dataTable;  int imageColumnIndex = 3;  int desiredColumnWidth = 200;  int desiredColumnHeight = 200;  foreach (DataGridViewColumn column in dataGridView3.Columns)  {  if (column.Index != imageColumnIndex)  {  column.AutoSizeMode = DataGridViewAutoSizeColumnMode.ColumnHeader;  }  }  dataGridView3.Columns[imageColumnIndex].Width = desiredColumnWidth;  foreach (DataGridViewRow row in dataGridView3.Rows)  {  if (row.Cells[imageColumnIndex].Value is Image image)  {  Image resizedImage = new Bitmap(image, desiredColumnWidth, desiredColumnHeight);  row.Cells[imageColumnIndex].Value = resizedImage;  }  row.Height = desiredColumnHeight + dataGridView3.RowTemplate.Height - dataGridView3.DefaultCellStyle.Padding.Vertical;  }  } |
| Кнопка button24\_Click | private void button24\_Click(object sender, EventArgs e)  {  flowLayoutPanel1.Width = widthFlowLayoutPanel;  flowLayoutPanel1.Height = heightFlowLayoutPanel;  x = button24.Location.X + button24.Size.Width;  y = button24.Location.Y;  flowLayoutPanel1.Controls.Clear();  flowLayoutPanel1.Location = new Point(x, y);  flowLayoutPanel1.Width = 200;  flowLayoutPanel1.BackColor = Color.FromArgb(192, 192, 255);  flowLayoutPanel1.Visible = true;  Button buttonПользователи = new Button();  buttonПользователи.Text = "Пользователи";  buttonПользователи.Width = flowLayoutPanel1.Width - 5;  buttonПользователи.Click += buttonПользователи\_Click;  flowLayoutPanel1.Controls.Add(buttonПользователи);  Button buttonДобавитьПользователя = new Button();  buttonДобавитьПользователя.Text = "Добавить пользователя";  buttonДобавитьПользователя.Width = flowLayoutPanel1.Width - 5;  buttonДобавитьПользователя.Click += buttonДобавитьПользователя\_Click;  flowLayoutPanel1.Controls.Add(buttonДобавитьПользователя);  } |
| Кнопка buttonДобавитьПользователя\_Click | private void buttonДобавитьПользователя\_Click(object sender, EventArgs e)  {  button10.Enabled = true;  button27.Enabled = true;  button17.Enabled = true;  indexOfPanel = 8;  OpenPanels.openPanels(tableLayoutPanel1, indexOfPanel);  } |
| Кнопка buttonПользователи\_Click | private void buttonПользователи\_Click(object sender, EventArgs e)  {  button10.Enabled = true;  button27.Enabled = true;  button17.Enabled = true;  indexOfPanel = 7;  indexOfDataGridView = 7;  nameOfTable = "Пользователи";  OpenPanels.openPanels(tableLayoutPanel1, indexOfPanel);  dataGridView3.DataSource = пользователиbindingSource;  dataGridView3.Columns["ПраваДоступа"].HeaderText = "Права доступа";  dataGridView3.Columns["КодПользователя"].HeaderText = "Код пользователя";  foreach (DataGridViewColumn column in dataGridView3.Columns)  {  column.AutoSizeMode = DataGridViewAutoSizeColumnMode.ColumnHeader;  }  } |
| Кнопка button26\_Click | private void button26\_Click(object sender, EventArgs e)  {  try  {  SqlCommand sqlCommand = new SqlCommand("INSERT INTO Пользователи (Логин, Пароль, ПраваДоступа) " +  "VALUES(@Логин, @Пароль, @ПраваДоступа)", sqlConnection);  sqlCommand.Parameters.AddWithValue("Логин", maskedTextBoxЛогин.Text);  sqlCommand.Parameters.AddWithValue("Пароль", maskedTextBoxПароль.Text);  sqlCommand.Parameters.AddWithValue("ПраваДоступа", comboBox1.SelectedItem.ToString());  sqlCommand.ExecuteNonQuery();  Notification notification = new Notification();  notification.titleLabel.ForeColor = Color.Orange;  notification.Title = "Добавлен новый пользователь";  notification.Show(3000);  }  catch (Exception ex)  {  MessageBox.Show("Ошибка: " + ex.Message);  }  } |
| Кнопка button27\_Click | private void button27\_Click(object sender, EventArgs e)  {  try  {  if (dataGridView3.SelectedRows.Count > 0)  {  var id = dataGridView3.SelectedRows[0].Cells[0].Value.ToString();  string Message;  Message = "Вы действительно хотите удалить выделенную запись?";  if (MessageBox.Show(Message, "Внимание", MessageBoxButtons.YesNo, MessageBoxIcon.Warning, MessageBoxDefaultButton.Button2) == DialogResult.No)  {  return;  }  var deleQuery = $"DELETE FROM {nameOfTable} WHERE {dataGridView3.Columns[0].Name} = {id}";  SqlCommand sqlCommand = new SqlCommand(deleQuery, sqlConnection);  sqlCommand.ExecuteNonQuery();  Notification notification = new Notification();  notification.titleLabel.ForeColor = Color.Orange;  notification.Title = "Запись удалена, обновите таблицу";  notification.Show(3000);  }  else  {  Notification notification = new Notification();  notification.titleLabel.ForeColor = Color.DarkRed;  notification.Title = "Выберите запись, которую следует удалить";  notification.Show(3000);  }  }  catch (Exception ex)  {  MessageBox.Show($"Произошла ошибка: {ex.Message}", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  } |
| Событие button27\_MouseHover | private void button27\_MouseHover(object sender, EventArgs e)  {  toolTip1.Show("Удалить запись в таблице", button10);  } |
| Событие dataGridView3\_CellContentClick | private void dataGridView3\_CellContentClick(object sender, DataGridViewCellEventArgs e)  {  if (e.ColumnIndex == 6)  {  if (e.RowIndex >= 0 && e.RowIndex < dataGridView3.Rows.Count - 1)  {  var openFileDialog = new OpenFileDialog();  openFileDialog.Filter = "JPEG Files (\*.jpeg;\*.jpg)|\*.jpeg;\*.jpg|PNG Files (\*.png)|\*.png";  if (openFileDialog.ShowDialog() == DialogResult.OK)  {  var selectedFiles = openFileDialog.FileNames;  foreach (var file in selectedFiles)  {  var imageBytes = File.ReadAllBytes(file);  dataGridView3.Rows[e.RowIndex].Cells[e.ColumnIndex].Value = imageBytes;  }  }  }  }  } |
| Событие HideFlowLayoutPanel | private void HideFlowLayoutPanel(object sender, EventArgs e)  {  flowLayoutPanel1.Visible = false;  } |

Свойства элементов формы Начальник представлены в таблице 5.

Таблица 5 – Свойства элементов формы Начальник

|  |  |  |
| --- | --- | --- |
| Объект | Свойство | Значение |
| Начальник | BackColor | 192; 192; 255 |
| FormBorderStyle | None |
| Icon | (Значок) |
| MaximizeBox | False |
| MinimizeBox | False |
| StartPosition | CenterScreen |
| Size | 1270; 1114 |
| Text | Мои сотрудники |

Событийно-управляемые процедуры в форме Начальник представлены в таблице 6.

Таблица 6 – Событийно-управляемые процедуры в форме «Начальник»

| Объект | Событийно-управляемые процедуры |
| --- | --- |
| Класс Chief | public partial class Chief: Form  {  public Chief()  {  InitializeComponent();  }  private SqlConnection sqlConnection = null;  private SqlDataAdapter sqlDataAdapter = null;  private OpenFileDialog openFileDialog = new OpenFileDialog();  private int x, y;  private int indexOfPanel = 7;  private int indexOfDataGridView = 0;  private string imagePath = null;  private string nameOfTable = null;  private const int widthFlowLayoutPanel = 95;  private const int heightFlowLayoutPanel = 60;  } |
| Событие Chief\_Load | private void Chief\_Load(object sender, EventArgs e)  {  sqlConnection = new SqlConnection(ConfigurationManager.ConnectionStrings["Отдел\_кадров\_университета.Properties.Settings.УП01\_ИСПП\_5\_Гаевский\_ААConnectionString"].ConnectionString);  sqlConnection.Open();  OpenPanels.openPanels(tableLayoutPanel1, indexOfPanel);  dataGridView3.Click += HideFlowLayoutPanel;  panel1.Click += HideFlowLayoutPanel;  panel2.Click += HideFlowLayoutPanel;  panel3.Click += HideFlowLayoutPanel;  panel4.Click += HideFlowLayoutPanel;  panel5.Click += HideFlowLayoutPanel;  panel6.Click += HideFlowLayoutPanel;  panel7.Click += HideFlowLayoutPanel;  panel8.Click += HideFlowLayoutPanel;  panel9.Click += HideFlowLayoutPanel;  dataGridView3.ColumnHeadersDefaultCellStyle.Font = new Font("Microsoft Sans Serif", 12);  dataGridView3.DefaultCellStyle.Font = new Font("Microsoft Sans Serif", 11);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП\_5\_Гаевский\_ААDataSet.Пользователи". При необходимости она может быть перемещена или удалена.  this.пользователиTableAdapter.Fill(this.уП01\_ИСПП\_5\_Гаевский\_ААDataSet.Пользователи);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ОбразованиеСотрудников". При необходимости она может быть перемещена или удалена.  this.образованиеСотрудниковTableAdapter.Fill(this.уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ОбразованиеСотрудников);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ДисциплинарныеМерыВОтношенииСотрудников". При необходимости она может быть перемещена или удалена.  this.дисциплинарныеМерыВОтношенииСотрудниковTableAdapter.Fill(this.уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ДисциплинарныеМерыВОтношенииСотрудников);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ИнформацияОПредыдущемРабочемМесте". При необходимости она может быть перемещена или удалена.  this.информацияОПредыдущемРабочемМестеTableAdapter.Fill(this.уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ИнформацияОПредыдущемРабочемМесте);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ИнформацияОРабочемМестеСотрудников". При необходимости она может быть перемещена или удалена.  this.информацияОРабочемМестеСотрудниковTableAdapter.Fill(this.уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ИнформацияОРабочемМестеСотрудников);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ЛичнаяИнформацияСотрудников". При необходимости она может быть перемещена или удалена.  this.личнаяИнформацияСотрудниковTableAdapter.Fill(this.уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ЛичнаяИнформацияСотрудников);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП\_5\_Гаевский\_ААDataSet.Сотрудники". При необходимости она может быть перемещена или удалена.  this.сотрудникиTableAdapter.Fill(this.уП01\_ИСПП\_5\_Гаевский\_ААDataSet.Сотрудники);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ТрудовойДоговор". При необходимости она может быть перемещена или удалена.  this.трудовойДоговорTableAdapter.Fill(this.уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ТрудовойДоговор);  } |
| Кнопка button6\_click | private void button6\_Click(object sender, EventArgs e)  {  flowLayoutPanel1.Width = widthFlowLayoutPanel;  flowLayoutPanel1.Height = heightFlowLayoutPanel;  x = button6.Location.X + button6.Size.Width;  y = button6.Location.Y;  flowLayoutPanel1.Controls.Clear();  flowLayoutPanel1.Location = new Point(x, y);  flowLayoutPanel1.Width = 200;  flowLayoutPanel1.BackColor = Color.FromArgb(192, 192, 255);  flowLayoutPanel1.Visible = true;  Button buttonТрудовыеДоговоры = new Button();  buttonТрудовыеДоговоры.Text = "Трудовые договоры";  buttonТрудовыеДоговоры.Width = flowLayoutPanel1.Width - 5;  buttonТрудовыеДоговоры.Click += buttonТрудовыеДоговоры\_Click;  flowLayoutPanel1.Controls.Add(buttonТрудовыеДоговоры);  Button buttonСоздатьТрудовойДоговор = new Button();  buttonСоздатьТрудовойДоговор.Text = "Добавить трудовой договор";  buttonСоздатьТрудовойДоговор.Width = flowLayoutPanel1.Width - 5;  buttonСоздатьТрудовойДоговор.Click += buttonСоздатьТрудовойДоговор\_Click;  flowLayoutPanel1.Controls.Add(buttonСоздатьТрудовойДоговор);  } |
| Кнопка buttonСоздатьТрудовойДоговор\_Click | private void buttonСоздатьТрудовойДоговор\_Click(object sender, EventArgs e)  {  button10.Enabled = true;  button27.Enabled = true;  button17.Enabled = true;  indexOfPanel = 0;  OpenPanels.openPanels(tableLayoutPanel1, indexOfPanel);  } |
| Кнопка buttonТрудовыеДоговоры\_Click | private void buttonТрудовыеДоговоры\_Click(object sender, EventArgs e)  {  button10.Enabled = true;  button27.Enabled = true;  button17.Enabled = true;  indexOfPanel = 7;  indexOfDataGridView = 0;  nameOfTable = "ТрудовойДоговор";  OpenPanels.openPanels(tableLayoutPanel1, indexOfPanel);  dataGridView3.DataSource = трудовойДоговорbindingSource;  dataGridView3.Columns["КодТрудовогоДоговора"].HeaderText = "Код трудового договора";  dataGridView3.Columns["РегистрационныйНомерСотрудника"].HeaderText = "Регистрационный номер сотрудника";  dataGridView3.Columns["КодРабочегоМеста"].HeaderText = "Код рабочего места";  dataGridView3.Columns["КоличествоРабочихСроков"].HeaderText = "Количество рабочих сроков";  dataGridView3.Columns["ДатаНачалаТрудовойДеятельности"].HeaderText = "Дата начала трудовой деятельности";  dataGridView3.Columns["ДатаОкончанияТрудовойДеятельности"].HeaderText = "Дата окончания трудовой деятельности";  foreach (DataGridViewColumn column in dataGridView3.Columns)  {  column.AutoSizeMode = DataGridViewAutoSizeColumnMode.ColumnHeader;  }  typeof(DataGridView).InvokeMember("DoubleBuffered", BindingFlags.NonPublic | BindingFlags.Instance | BindingFlags.SetProperty, null, dataGridView3, new object[] { true });  dataGridView3.Scroll += (scrollSender, scrollEventArgs) =>  {  if (scrollEventArgs.ScrollOrientation == ScrollOrientation.VerticalScroll)  {  dataGridView3.FirstDisplayedScrollingRowIndex = scrollEventArgs.NewValue;  }  else if (scrollEventArgs.ScrollOrientation == ScrollOrientation.HorizontalScroll)  {  dataGridView3.FirstDisplayedScrollingColumnIndex = scrollEventArgs.NewValue;  }  };  } |
| Кнопка button2\_Click | private void button2\_Click(object sender, EventArgs e)  {  flowLayoutPanel1.Width = widthFlowLayoutPanel;  flowLayoutPanel1.Height = heightFlowLayoutPanel;  x = button2.Location.X + button2.Size.Width;  y = button2.Location.Y;  flowLayoutPanel1.Controls.Clear();  flowLayoutPanel1.Location = new Point(x, y);  flowLayoutPanel1.Width = 150;  flowLayoutPanel1.BackColor = Color.FromArgb(192, 192, 255);  flowLayoutPanel1.Visible = true;  Button buttonСотрудники = new Button();  buttonСотрудники.Text = "Сотрудники";  buttonСотрудники.Width = flowLayoutPanel1.Width - 5;  buttonСотрудники.BackColor = button2.BackColor;  buttonСотрудники.Click += buttonСотрудники\_Click;  flowLayoutPanel1.Controls.Add(buttonСотрудники);  Button buttonСоздатьСотрудника = new Button();  buttonСоздатьСотрудника.Text = "Добавить сотрудника";  buttonСоздатьСотрудника.Width = flowLayoutPanel1.Width - 5;  buttonСотрудники.BackColor = button2.BackColor;  buttonСоздатьСотрудника.Click += buttonСоздатьСотрудника\_Click;  flowLayoutPanel1.Controls.Add(buttonСоздатьСотрудника);  } |
| Кнопка buttonСоздатьСотрудника\_Click | private void buttonСоздатьСотрудника\_Click(object sender, EventArgs e)  {  button10.Enabled = true;  button27.Enabled = true;  button17.Enabled = true;  indexOfPanel = 1;  OpenPanels.openPanels(tableLayoutPanel1, indexOfPanel);  } |
| Кнопка buttonСотрудники\_Click | private void buttonСотрудники\_Click(object sender, EventArgs e)  {  button10.Enabled = true;  button27.Enabled = true;  button17.Enabled = true;  indexOfPanel = 7;  indexOfDataGridView = 1;  nameOfTable = "Сотрудники";  OpenPanels.openPanels(tableLayoutPanel1, indexOfPanel);  dataGridView3.DataSource = сотрудникиbindingSource;  int imageColumnIndex = 6;  int desiredColumnWidth = 200;  int desiredColumnHeight = 200;  foreach (DataGridViewColumn column in dataGridView3.Columns)  {  if (column.Index != imageColumnIndex)  {  column.AutoSizeMode = DataGridViewAutoSizeColumnMode.ColumnHeader;  }  }  dataGridView3.Columns[imageColumnIndex].Width = desiredColumnWidth;  foreach (DataGridViewRow row in dataGridView3.Rows)  {  if (row.Cells[imageColumnIndex].Value is Image image)  {  Image resizedImage = new Bitmap(image, desiredColumnWidth, desiredColumnHeight);  row.Cells[imageColumnIndex].Value = resizedImage;  }  row.Height = desiredColumnHeight + dataGridView3.RowTemplate.Height - dataGridView3.DefaultCellStyle.Padding.Vertical;  }  dataGridView3.Columns["РегистрационныйНомерСотрудника"].HeaderText = "Регистрационный номер сотрудника";  dataGridView3.Columns["Дата\_рождения"].HeaderText = "Дата рождения";  dataGridView3.Columns["Место\_рождения"].HeaderText = "Место рождения";  } |
| Кнопка button3\_Click | private void button3\_Click(object sender, EventArgs e)  {  flowLayoutPanel1.Width = widthFlowLayoutPanel;  flowLayoutPanel1.Height = heightFlowLayoutPanel;  x = button3.Location.X + button3.Size.Width;  y = button3.Location.Y;  flowLayoutPanel1.Controls.Clear();  flowLayoutPanel1.Location = new Point(x, y);  flowLayoutPanel1.Width = 200;  flowLayoutPanel1.BackColor = Color.FromArgb(192, 192, 255);  flowLayoutPanel1.Visible = true;  Button buttonЛичнаяИнформацияСотрудников = new Button();  buttonЛичнаяИнформацияСотрудников.Text = "Личная ифнормация сотрудников";  buttonЛичнаяИнформацияСотрудников.Width = flowLayoutPanel1.Width - 5;  buttonЛичнаяИнформацияСотрудников.Click += buttonЛичнаяИнформацияСотрудников\_Click;  flowLayoutPanel1.Controls.Add(buttonЛичнаяИнформацияСотрудников);  Button buttonСоздатьЛичнуюИнформациюСотрудника = new Button();  buttonСоздатьЛичнуюИнформациюСотрудника.Text = "Добавить личную информацию";  buttonСоздатьЛичнуюИнформациюСотрудника.Width = flowLayoutPanel1.Width - 5;  buttonСоздатьЛичнуюИнформациюСотрудника.Click += buttonСоздатьЛичнуюИнформациюСотрудника\_Click;  flowLayoutPanel1.Controls.Add(buttonСоздатьЛичнуюИнформациюСотрудника);  } |
| Кнопка buttonСоздатьЛичнуюИнформациюСотрудника\_Click | private void buttonСоздатьЛичнуюИнформациюСотрудника\_Click(object sender, EventArgs e)  {  button10.Enabled = true;  button27.Enabled = true;  button17.Enabled = true;  indexOfPanel = 2;  OpenPanels.openPanels(tableLayoutPanel1, indexOfPanel);  } |
| Кнопка buttonЛичнаяИнформацияСотрудников\_Click | private void buttonЛичнаяИнформацияСотрудников\_Click(object sender, EventArgs e)  {  button10.Enabled = true;  button27.Enabled = true;  button17.Enabled = true;  indexOfPanel = 7;  indexOfDataGridView = 2;  nameOfTable = "ЛичнаяИнформацияСотрудников";  OpenPanels.openPanels(tableLayoutPanel1, indexOfPanel);  dataGridView3.DataSource = личнаяИнформацияСотрудниковbindingSource;  dataGridView3.Columns["КодЛичнойИнформацииСотрудника"].HeaderText = "Код личной информации сотрудника";  dataGridView3.Columns["РегистрационныйНомерСотрудника"].HeaderText = "Регистрационный номер сотрудника";  dataGridView3.Columns["ДатаВыдачиПаспорта"].HeaderText = "Дата выдачи паспорта";  dataGridView3.Columns["КемВыдан"].HeaderText = "Кем выдан";  dataGridView3.Columns["ДомашнийАдрес"].HeaderText = "Домашний адрес";  dataGridView3.Columns["ДомашнийТелефон"].HeaderText = "Домашний телефон";  dataGridView3.Columns["АдресЭлектроннойПочты"].HeaderText = "Адрес электронной почты";  foreach (DataGridViewColumn column in dataGridView3.Columns)  {  column.AutoSizeMode = DataGridViewAutoSizeColumnMode.ColumnHeader;  }  } |
| Кнопка button4\_Click | private void button4\_Click(object sender, EventArgs e)  {  flowLayoutPanel1.Width = widthFlowLayoutPanel;  flowLayoutPanel1.Height = heightFlowLayoutPanel;  x = button4.Location.X + button4.Size.Width;  y = button4.Location.Y;  flowLayoutPanel1.Controls.Clear();  flowLayoutPanel1.Location = new Point(x, y);  flowLayoutPanel1.Width = 200;  flowLayoutPanel1.Height = 120;  flowLayoutPanel1.BackColor = Color.FromArgb(192, 192, 255);  flowLayoutPanel1.Visible = true;  Button buttonРабочееМестоСотрудника = new Button();  buttonРабочееМестоСотрудника.Text = "Рабочие места";  buttonРабочееМестоСотрудника.Width = flowLayoutPanel1.Width - 5;  buttonРабочееМестоСотрудника.Click += buttonРабочееМестоСотрудника\_Click;  flowLayoutPanel1.Controls.Add(buttonРабочееМестоСотрудника);  Button buttonБывшееРабочееМестоСотрудника = new Button();  buttonБывшееРабочееМестоСотрудника.Text = "Предыдущие рабочие места";  buttonБывшееРабочееМестоСотрудника.Width = flowLayoutPanel1.Width - 5;  buttonБывшееРабочееМестоСотрудника.Click += buttonБывшееРабочееМестоСотрудника\_Click;  flowLayoutPanel1.Controls.Add(buttonБывшееРабочееМестоСотрудника);  Button buttonДобавитьРабочееМесто = new Button();  buttonДобавитьРабочееМесто.Text = "Добавить рабочее место";  buttonДобавитьРабочееМесто.Width = flowLayoutPanel1.Width - 5;  buttonДобавитьРабочееМесто.Click += buttonДобавитьРабочееМесто\_Click;  flowLayoutPanel1.Controls.Add(buttonДобавитьРабочееМесто);  Button buttonДобавитьБывшееРабочееМесто = new Button();  buttonДобавитьБывшееРабочееМесто.Text = "Добавить предыдущее место";  buttonДобавитьБывшееРабочееМесто.Width = flowLayoutPanel1.Width - 5;  buttonДобавитьБывшееРабочееМесто.Click += buttonДобавитьБывшееРабочееМесто\_Click;  flowLayoutPanel1.Controls.Add(buttonДобавитьБывшееРабочееМесто);  } |
| Кнопка buttonДобавитьБывшееРабочееМесто\_Click | private void buttonДобавитьБывшееРабочееМесто\_Click(object sender, EventArgs e)  {  button10.Enabled = true;  button27.Enabled = true;  button17.Enabled = true;  indexOfPanel = 4;  OpenPanels.openPanels(tableLayoutPanel1, indexOfPanel);  } |
| Кнопка buttonДобавитьРабочееМесто\_Click | private void buttonДобавитьРабочееМесто\_Click(object sender, EventArgs e)  {  button10.Enabled = true;  button27.Enabled = true;  button17.Enabled = true;  indexOfPanel = 3;  OpenPanels.openPanels(tableLayoutPanel1, indexOfPanel);  } |
| Кнопка buttonБывшееРабочееМестоСотрудника\_Click | private void buttonБывшееРабочееМестоСотрудника\_Click(object sender, EventArgs e)  {  button10.Enabled = true;  button27.Enabled = true;  button17.Enabled = true;  indexOfPanel = 7;  indexOfDataGridView = 3;  nameOfTable = "ИнформацияОПредыдущемРабочемМесте";  OpenPanels.openPanels(tableLayoutPanel1, indexOfPanel);  dataGridView3.DataSource = информацияОПредыдущемРабочемМестеbindingSource;  dataGridView3.Columns["КодПредыдущегоРабочегоМеста"].HeaderText = "Код предыдущего рабочего места";  dataGridView3.Columns["РегистрационныйНомерСотрудника"].HeaderText = "Регистрационный номер сотрудника";  dataGridView3.Columns["НазваниеПредприятия"].HeaderText = "Название предприятия";  dataGridView3.Columns["ДатаУстройства"].HeaderText = "Дата устройства";  dataGridView3.Columns["ДатаУвольнения"].HeaderText = "Дата увольнения";  dataGridView3.Columns["ПричинаУвольнения"].HeaderText = "Причина увольнения";  dataGridView3.Columns["АдресПредприятия"].HeaderText = "Адрес предприятия";  dataGridView3.Columns["ТелефонПредприятия"].HeaderText = "Телефон предприятия";  foreach (DataGridViewColumn column in dataGridView3.Columns)  {  column.AutoSizeMode = DataGridViewAutoSizeColumnMode.ColumnHeader;  }  } |
| Кнопка buttonРабочееМестоСотрудника\_Click | private void buttonРабочееМестоСотрудника\_Click(object sender, EventArgs e)  {  button10.Enabled = true;  button27.Enabled = true;  button17.Enabled = true;  indexOfPanel = 7;  indexOfDataGridView = 4;  nameOfTable = "ИнформацияОРабочемМестеСотрудников";  OpenPanels.openPanels(tableLayoutPanel1, indexOfPanel);  dataGridView3.DataSource = информацияОРабочемМестеСотрудниковbindingSource;  dataGridView3.Columns["КодРабочегоМеста"].HeaderText = "Код рабочего места";  dataGridView3.Columns["РегистрационныйНомерСотрудника"].HeaderText = "Регистрационный номер сотрудника";  dataGridView3.Columns["НазваниеКафедры"].HeaderText = "Название кафедры";  dataGridView3.Columns["ЗанимаемаяДолжность"].HeaderText = "Занимаемая должность";  foreach (DataGridViewColumn column in dataGridView3.Columns)  {  column.AutoSizeMode = DataGridViewAutoSizeColumnMode.ColumnHeader;  }  } |
| Кнопка button5\_Click | private void button5\_Click(object sender, EventArgs e)  {  flowLayoutPanel1.Width = widthFlowLayoutPanel;  flowLayoutPanel1.Height = heightFlowLayoutPanel;  x = button5.Location.X + button5.Size.Width;  y = button5.Location.Y;  flowLayoutPanel1.Controls.Clear();  flowLayoutPanel1.Location = new Point(x, y);  flowLayoutPanel1.Width = 200;  flowLayoutPanel1.BackColor = Color.FromArgb(192, 192, 255);  flowLayoutPanel1.Visible = true;  Button buttonДисциплинарныеМеры = new Button();  buttonДисциплинарныеМеры.Text = "Дисциплинарные меры";  buttonДисциплинарныеМеры.Width = flowLayoutPanel1.Width - 5;  buttonДисциплинарныеМеры.Click += buttonДисциплинарныеМеры\_Click;  flowLayoutPanel1.Controls.Add(buttonДисциплинарныеМеры);  Button buttonДобавитьВзыскание = new Button();  buttonДобавитьВзыскание.Text = "Вознаграждения и поощрения";  buttonДобавитьВзыскание.Width = flowLayoutPanel1.Width - 5;  buttonДобавитьВзыскание.Click += buttonДобавитьВзыскание\_Click;  flowLayoutPanel1.Controls.Add(buttonДобавитьВзыскание);  } |
| Кнопка buttonДобавитьВзыскание\_Click | private void buttonДобавитьВзыскание\_Click(object sender, EventArgs e)  {  button10.Enabled = true;  button27.Enabled = true;  button17.Enabled = true;  indexOfPanel = 5;  OpenPanels.openPanels(tableLayoutPanel1, indexOfPanel);  } |
| Кнопка buttonДисциплинарныеМеры\_Click | private void buttonДисциплинарныеМеры\_Click(object sender, EventArgs e)  {  button10.Enabled = true;  button27.Enabled = true;  button17.Enabled = true;  indexOfPanel = 7;  indexOfDataGridView = 5;  nameOfTable = "ДисциплинарныеМерыВОтношенииСотрудников";  OpenPanels.openPanels(tableLayoutPanel1, indexOfPanel);  dataGridView3.DataSource = дисциплинарныеМерыВОтношенииСотрудниковbindingSource;  dataGridView3.Columns["КодДисциплинарнойМеры"].HeaderText = "Код дисциплинарной меры";  dataGridView3.Columns["РегистрационныйНомерСотрудника"].HeaderText = "Регистрационный номер сотрудника";  dataGridView3.Columns["СведенияОВзысканиях"].HeaderText = "Сведения о взысканиях";  dataGridView3.Columns["СведенияОНаграждениях"].HeaderText = "Сведения о награждениях";  } |
| Кнопка button7\_Click | private void button7\_Click(object sender, EventArgs e)  {  flowLayoutPanel1.Width = widthFlowLayoutPanel;  flowLayoutPanel1.Height = heightFlowLayoutPanel;  x = button7.Location.X + button7.Size.Width;  y = button7.Location.Y;  flowLayoutPanel1.Controls.Clear();  flowLayoutPanel1.Location = new Point(x, y);  flowLayoutPanel1.Width = 200;  flowLayoutPanel1.BackColor = Color.FromArgb(192, 192, 255);  flowLayoutPanel1.Visible = true;  Button buttonОбразованиеСотрудников = new Button();  buttonОбразованиеСотрудников.Text = "Образование сотрудников";  buttonОбразованиеСотрудников.Width = flowLayoutPanel1.Width - 5;  buttonОбразованиеСотрудников.Click += buttonОбразованиеСотрудников\_Click;  flowLayoutPanel1.Controls.Add(buttonОбразованиеСотрудников);  Button buttonДобавитьОбразованиеСотрудника = new Button();  buttonДобавитьОбразованиеСотрудника.Text = "Добавить образование";  buttonДобавитьОбразованиеСотрудника.Width = flowLayoutPanel1.Width - 5;  buttonДобавитьОбразованиеСотрудника.Click += buttonДобавитьОбразованиеСотрудника\_Click;  flowLayoutPanel1.Controls.Add(buttonДобавитьОбразованиеСотрудника);  } |
| Кнопка buttonДобавитьОбразованиеСотрудника\_Click | private void buttonДобавитьОбразованиеСотрудника\_Click(object sender, EventArgs e)  {  button10.Enabled = true;  button27.Enabled = true;  button17.Enabled = true;  indexOfPanel = 6;  OpenPanels.openPanels(tableLayoutPanel1, indexOfPanel);  } |
| Кнопка buttonДобавлениеТрудовогоДоговора\_Click | private void buttonДобавлениеТрудовогоДоговора\_Click(object sender, EventArgs e)  {  try  {  SqlCommand sqlCommand = new SqlCommand("INSERT INTO ТрудовойДоговор(РегистрационныйНомерСотрудника, КодРабочегоМеста, КоличествоРабочихСроков, ДатаНачалаТрудовойДеятельности, ДатаОкончанияТрудовойДеятельности, Оклад) " +  "VALUES (@РегистрационныйНомерСотрудника, @КодРабочегоМеста, @КоличествоРабочихСроков, @ДатаНачалаТрудовойДеятельности, @ДатаОкончанияТрудовойДеятельности, @Оклад)", sqlConnection);  DateTime firstDate = DateTime.Parse(maskedTextBoxДатаНачТрудДеят.Text);  DateTime? secondDate = null;  if (!string.IsNullOrEmpty(maskedTextBoxДатаОкончТрудДеят.Text))  {  string dateInput = maskedTextBoxДатаОкончТрудДеят.Text;  string dateFormat = "dd.MM.yyyy"; // Замените на нужный формат маски  if (DateTime.TryParseExact(dateInput, dateFormat, CultureInfo.InvariantCulture, DateTimeStyles.None, out DateTime parsedDate))  {  secondDate = parsedDate;  }  }  sqlCommand.Parameters.AddWithValue("РегистрационныйНомерСотрудника", maskedTextBoxРНС1.Text);  sqlCommand.Parameters.AddWithValue("КодРабочегоМеста", maskedTextBoxКодРабМеста.Text);  sqlCommand.Parameters.AddWithValue("КоличествоРабочихСроков", maskedTextBoxКолРабСроков.Text);  sqlCommand.Parameters.AddWithValue("ДатаНачалаТрудовойДеятельности", $"{firstDate.Month}/{firstDate.Day}/{firstDate.Year}");  if (secondDate.HasValue)  {  sqlCommand.Parameters.AddWithValue("ДатаОкончанияТрудовойДеятельности", $"{secondDate.Value.Month}/{secondDate.Value.Day}/{secondDate.Value.Year}");  }  else  {  sqlCommand.Parameters.AddWithValue("ДатаОкончанияТрудовойДеятельности", DBNull.Value);  }  sqlCommand.Parameters.AddWithValue("Оклад", maskedTextBoxОклад.Text);  if (string.IsNullOrEmpty(maskedTextBoxРНС1.Text) || string.IsNullOrEmpty(maskedTextBoxКодРабМеста.Text) || string.IsNullOrEmpty(maskedTextBoxКолРабСроков.Text) || string.IsNullOrEmpty(maskedTextBoxОклад.Text))  {  Notification notification = new Notification();  notification.titleLabel.ForeColor = Color.Red;  notification.Title = "Не все поля заполнены";  notification.Show(3000);  }  else  {  sqlCommand.ExecuteNonQuery();  Notification notification = new Notification();  notification.titleLabel.ForeColor = Color.ForestGreen;  notification.Title = "Добавлена новая строка в таблицу 'Трудовой договор'";  notification.Show(3000);  }  }  catch (Exception ex)  {  MessageBox.Show("Ошибка: " + ex.Message);  }  } |
| Кнопка buttonДобавлениеСотрудника\_Click | private void buttonДобавлениеСотрудника\_Click(object sender, EventArgs e)  {  try  {  SqlCommand sqlCommand = new SqlCommand("INSERT INTO Сотрудники (Фамилия, Имя, Отчество, Дата\_рождения, Место\_рождения, Фото, Примечания) " +  "VALUES(@Фамилия, @Имя, @Отчество, @Дата\_рождения, @Место\_рождения, @Фото, @Примечания)", sqlConnection);  DateTime firstDate = DateTime.Parse(maskedTextBoxДатаРождения.Text);  Notification notification = new Notification();  if (!string.IsNullOrEmpty(imagePath))  {  byte[] imageData = File.ReadAllBytes(imagePath);  sqlCommand.Parameters.AddWithValue("Фамилия", maskedTextBoxФамилия.Text);  sqlCommand.Parameters.AddWithValue("Имя", maskedTextBoxИмя.Text);  sqlCommand.Parameters.AddWithValue("Отчество", maskedTextBoxОтчество.Text);  sqlCommand.Parameters.AddWithValue("Дата\_рождения", $"{firstDate.Month}/{firstDate.Day}/{firstDate.Year}");  sqlCommand.Parameters.AddWithValue("Место\_рождения", maskedTextBoxМестоРождения.Text);  sqlCommand.Parameters.Add("@Фото", SqlDbType.VarBinary, -1).Value = imageData;  sqlCommand.Parameters.AddWithValue("Примечания", maskedTextBoxПримечания.Text);  sqlCommand.ExecuteNonQuery();  notification = new Notification();  notification.titleLabel.ForeColor = Color.ForestGreen;  notification.Title = "Добавлена новая строка в таблицу 'Сотрудники'";  notification.Show(3000);  }  else  {  notification = new Notification();  notification.titleLabel.ForeColor = Color.Red;  notification.Title = "Выберите фотографию сотрудника";  notification.Show(3000);  }  }  catch (Exception ex)  {  MessageBox.Show("Ошибка: " + ex.Message);  }  } |
| Кнопка buttonДобавлениеЛичнойИнформацииСотрудника\_Click | private void buttonДобавлениеЛичнойИнформацииСотрудника\_Click(object sender, EventArgs e)  {  try  {  SqlCommand sqlCommand = new SqlCommand("INSERT INTO ЛичнаяИнформацияСотрудников (РегистрационныйНомерСотрудника, Паспорт, ДатаВыдачиПаспорта, КемВыдан, ДомашнийАдрес, ДомашнийТелефон, АдресЭлектроннойПочты) " +  "VALUES(@РегистрационныйНомерСотрудника, @Паспорт, @ДатаВыдачиПаспорта, @КемВыдан, @ДомашнийАдрес, @ДомашнийТелефон, @АдресЭлектроннойПочты)", sqlConnection);  DateTime firstDate = DateTime.Parse(maskedTextBoxДатаВыдачиПаспорта.Text);  sqlCommand.Parameters.AddWithValue("РегистрационныйНомерСотрудника", maskedTextBoxРНС3.Text);  sqlCommand.Parameters.AddWithValue("Паспорт", maskedTextBoxПаспорт.Text);  sqlCommand.Parameters.AddWithValue("ДатаВыдачиПаспорта", $"{firstDate.Month}/{firstDate.Day}/{firstDate.Year}");  sqlCommand.Parameters.AddWithValue("КемВыдан", maskedTextBoxКемВыдан.Text);  sqlCommand.Parameters.AddWithValue("ДомашнийАдрес", maskedTextBoxДомашнийАдрес.Text);  sqlCommand.Parameters.AddWithValue("ДомашнийТелефон", maskedTextBoxДомашнийТелефон.Text);  sqlCommand.Parameters.AddWithValue("АдресЭлектроннойПочты", maskedTextBoxАдресЭлПочты.Text);  sqlCommand.ExecuteNonQuery();  Notification notification = new Notification();  notification.titleLabel.ForeColor = Color.ForestGreen;  notification.Title = "Добавлена новая строка в таблицу 'Личная информация сотрудников'";  notification.Show(3000);  }  catch (Exception ex)  {  MessageBox.Show("Ошибка: " + ex.Message);  }  } |
| Кнопка buttonДобавлениеРабочегоМестаСотрудника\_Click | private void buttonДобавлениеРабочегоМестаСотрудника\_Click(object sender, EventArgs e)  {  try  {  SqlCommand sqlCommand = new SqlCommand("INSERT INTO ИнформацияОРабочемМестеСотрудников(РегистрационныйНомерСотрудника, НазваниеКафедры, ЗанимаемаяДолжность) " +  "VALUES(@РегистрационныйНомерСотрудника, @НазваниеКафедры, @ЗанимаемаяДолжность)", sqlConnection);  sqlCommand.Parameters.AddWithValue("РегистрационныйНомерСотрудника", maskedTextBoxРНС4.Text);  sqlCommand.Parameters.AddWithValue("НазваниеКафедры", maskedTextBoxНазваниеКафедры.Text);  sqlCommand.Parameters.AddWithValue("ЗанимаемаяДолжность", maskedTextBoxЗанимаемаяДолжность.Text);  sqlCommand.ExecuteNonQuery();  Notification notification = new Notification();  notification.titleLabel.ForeColor = Color.ForestGreen;  notification.Title = "Добавлена новая строка в таблицу 'Информация о рабочем месте'";  notification.Show(3000);  }  catch (Exception ex)  {  MessageBox.Show("Ошибка: " + ex.Message);  }  } |
| Кнопка buttonДобавлениеПредыдущегоРабочегоМестаСотрудника\_Click | private void buttonДобавлениеПредыдущегоРабочегоМестаСотрудника\_Click(object sender, EventArgs e)  {  try  {  SqlCommand sqlCommand = new SqlCommand("INSERT INTO ИнформацияОПредыдущемРабочемМесте (РегистрационныйНомерСотрудника, НазваниеПредприятия, Должность, ДатаУстройства, ДатаУвольнения, ПричинаУвольнения, АдресПредприятия, ТелефонПредприятия) " +  "VALUES(@РегистрационныйНомерСотрудника, @НазваниеПредприятия , @Должность, @ДатаУстройства, @ДатаУвольнения, @ПричинаУвольнения, @АдресПредприятия, @ТелефонПредприятия)", sqlConnection);  DateTime firstDate = DateTime.Parse(maskedTextBoxДатаУстройства.Text);  DateTime secondDate = DateTime.Parse(maskedTextBoxДатаУвольнения.Text);  sqlCommand.Parameters.AddWithValue("РегистрационныйНомерСотрудника", maskedTextBoxРНС5.Text);  sqlCommand.Parameters.AddWithValue("НазваниеПредприятия", maskedTextBoxНазваниеПредприятия.Text);  sqlCommand.Parameters.AddWithValue("Должность", maskedTextBoxДолжность.Text);  sqlCommand.Parameters.AddWithValue("ДатаУстройства", $"{firstDate.Month}/{firstDate.Day}/{firstDate.Year}");  sqlCommand.Parameters.AddWithValue("ДатаУвольнения", $"{secondDate.Month}/{secondDate.Day}/{secondDate.Year}");  sqlCommand.Parameters.AddWithValue("ПричинаУвольнения", maskedTextBoxПричинаУвольнения.Text);  sqlCommand.Parameters.AddWithValue("АдресПредприятия", maskedTextBoxАдресПредприятия.Text);  sqlCommand.Parameters.AddWithValue("ТелефонПредприятия", maskedTextBoxТелефонПредприятия.Text);  sqlCommand.ExecuteNonQuery();  Notification notification = new Notification();  notification.titleLabel.ForeColor = Color.ForestGreen;  notification.Title = "Добавлена новая строка в таблицу 'Информация о предыдущем рабочем месте'";  notification.Show(3000);  }  catch (Exception ex)  {  MessageBox.Show("Ошибка: " + ex.Message);  }  } |
| Кнопка buttonДобавлениеВзысканияНаграждения\_Click | private void buttonДобавлениеВзысканияНаграждения\_Click(object sender, EventArgs e)  {  try  {  SqlCommand sqlCommand = new SqlCommand("INSERT INTO ДисциплинарныеМерыВОтношенииСотрудников(РегистрационныйНомерСотрудника, СведенияОВзысканиях, СведенияОНаграждениях) " +  "VALUES(@РегистрационныйНомерСотрудника, @СведенияОВзысканиях, @СведенияОНаграждениях)", sqlConnection);  sqlCommand.Parameters.AddWithValue("РегистрационныйНомерСотрудника", maskedTextBoxРНС6.Text);  sqlCommand.Parameters.AddWithValue("СведенияОВзысканиях", maskedTextBoxСведенияОВзысканиях.Text);  sqlCommand.Parameters.AddWithValue("СведенияОНаграждениях", maskedTextBoxСведенияОНаграждениях.Text);  sqlCommand.ExecuteNonQuery();  Notification notification = new Notification();  notification.titleLabel.ForeColor = Color.ForestGreen;  notification.Title = "Добавлена новая строка в таблицу 'Дисциплинарные меры в отношении сотрудников'";  notification.Show(3000);  }  catch (Exception ex)  {  MessageBox.Show("Ошибка: " + ex.Message);  }  } |
| Событие maskedTextBoxСведенияОВзысканиях\_TextChanged | private void maskedTextBoxСведенияОВзысканиях\_TextChanged(object sender, EventArgs e)  {  if (!string.IsNullOrWhiteSpace(maskedTextBoxСведенияОНаграждениях.Text))  {  maskedTextBoxСведенияОВзысканиях.Text = "";  Notification notification = new Notification();  notification.titleLabel.ForeColor = Color.Red;  notification.Title = "Поле 'Сведения о награждениях' должно быть пустым";  notification.Show(3000);  }  } |
| Событие maskedTextBoxСведенияОНаграждениях\_TextChanged | private void maskedTextBoxСведенияОНаграждениях\_TextChanged(object sender, EventArgs e)  {  if (!string.IsNullOrWhiteSpace(maskedTextBoxСведенияОВзысканиях.Text))  {  maskedTextBoxСведенияОНаграждениях.Text = "";  Notification notification = new Notification();  notification.titleLabel.ForeColor = Color.Red;  notification.Title = "Поле 'Сведения о взысканиях' должно быть пустым";  notification.Show(3000);  }  } |
| Кнопка buttonДобавлениеОбразованияСотрудника\_Click | private void buttonДобавлениеОбразованияСотрудника\_Click(object sender, EventArgs e)  {  try  {  SqlCommand sqlCommand = new SqlCommand("INSERT INTO ОбразованиеСотрудников (РегистрационныйНомерСотрудника, ВУЗ, ГодОкончанияВУЗа, Специальность, УченаяСтепень, УченоеЗвание) " +  "VALUES(@РегистрационныйНомерСотрудника, @ВУЗ, @ГодОкончанияВУЗа, @Специальность, @УченаяСтепень, @УченоеЗвание)", sqlConnection);  DateTime firstDate = DateTime.Parse(maskedTextBoxГодОкончанияВУЗа.Text);  sqlCommand.Parameters.AddWithValue("РегистрационныйНомерСотрудника", maskedTextBoxРНС7.Text);  sqlCommand.Parameters.AddWithValue("ВУЗ", maskedTextBoxВУЗ.Text);  sqlCommand.Parameters.AddWithValue("ГодОкончанияВУЗа", $"{firstDate.Month}/{firstDate.Day}/{firstDate.Year}");  sqlCommand.Parameters.AddWithValue("Специальность", maskedTextBoxСпециальность.Text);  sqlCommand.Parameters.AddWithValue("УченаяСтепень", maskedTextBoxУченаяСтепень.Text);  sqlCommand.Parameters.AddWithValue("УченоеЗвание", maskedTextBoxУченоеЗвание.Text);  sqlCommand.ExecuteNonQuery();  Notification notification = new Notification();  notification.titleLabel.ForeColor = Color.ForestGreen;  notification.Title = "Добавлена новая строка в таблицу 'Образование сотрудников'";  notification.Show(3000);  }  catch (Exception ex)  {  MessageBox.Show("Ошибка: " + ex.Message);  }  } |
| Кнопка button9\_Click | private void button9\_Click(object sender, EventArgs e)  {  Application.Exit();  } |
| Кнопка button11\_Click | private void button11\_Click(object sender, EventArgs e)  {  Hide();  Authorization UserForm = new Authorization();  UserForm.Show();  } |
| Кнопка button13\_Click | private void button13\_Click(object sender, EventArgs e)  {  openFileDialog.Filter = "Изображения (\*.png;\*.jpg)|\*.png;\*.jpg";  openFileDialog.Multiselect = false;  if (openFileDialog.ShowDialog() == DialogResult.OK)  {  imagePath = openFileDialog.FileName;  }  maskedTextBoxФото.Text = imagePath;  } |
| Событие dataGridView3\_DataError | private void dataGridView3\_DataError(object sender, DataGridViewDataErrorEventArgs e)  {  Image defaultImage = Resource1.Noname;  if (e.ColumnIndex == 6 && e.RowIndex >= 0)  {  e.ThrowException = false;  e.Cancel = true;  dataGridView3.Rows[e.RowIndex].Cells[e.ColumnIndex].Value = defaultImage;  }  } |
| Кнопка button12\_Click | private void button12\_Click(object sender, EventArgs e)  {  string searchText = textBox1.Text;  foreach (DataGridViewRow row in dataGridView3.Rows)  {  bool isMatch = false;  foreach (DataGridViewCell cell in row.Cells)  {  if (cell.Value != null && cell.Value.ToString().Contains(searchText))  {  isMatch = true;  break;  }  }  if (isMatch)  {  row.DefaultCellStyle.BackColor = Color.LightBlue;  }  else  {  row.DefaultCellStyle.BackColor = dataGridView3.DefaultCellStyle.BackColor;  }  }  } |
| Событие Administrator\_Click | private void Administrator\_Click(object sender, EventArgs e)  {  if (!flowLayoutPanel1.ClientRectangle.Contains(flowLayoutPanel1.PointToClient(Cursor.Position)))  {  flowLayoutPanel1.Visible = false;  }  } |
| Кнопка button16\_Click | private void button16\_Click(object sender, EventArgs e)  {  foreach (DataGridViewRow row in dataGridView3.Rows)  {  row.DefaultCellStyle.BackColor = dataGridView3.DefaultCellStyle.BackColor;  }  textBox1.Text = string.Empty;  } |
| Кнопка button10\_Click | private void button10\_Click(object sender, EventArgs e)  {  трудовойДоговорTableAdapter.Update(уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ТрудовойДоговор);  сотрудникиTableAdapter.Update(уП01\_ИСПП\_5\_Гаевский\_ААDataSet.Сотрудники);  образованиеСотрудниковTableAdapter.Update(уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ОбразованиеСотрудников);  личнаяИнформацияСотрудниковTableAdapter.Update(уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ЛичнаяИнформацияСотрудников);  информацияОПредыдущемРабочемМестеTableAdapter.Update(уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ИнформацияОПредыдущемРабочемМесте);  дисциплинарныеМерыВОтношенииСотрудниковTableAdapter.Update(уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ДисциплинарныеМерыВОтношенииСотрудников);  информацияОРабочемМестеСотрудниковTableAdapter.Update(уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ИнформацияОРабочемМестеСотрудников);  пользователиTableAdapter.Update(уП01\_ИСПП\_5\_Гаевский\_ААDataSet.Пользователи);  Notification notification = new Notification();  notification.titleLabel.ForeColor = Color.ForestGreen;  notification.Title = "Запись изменена";  notification.Height = 75;  notification.Show(3000);  } |
| Кнопка buttonОбразованиеСотрудников\_Click | private void buttonОбразованиеСотрудников\_Click(object sender, EventArgs e)  {  button10.Enabled = true;  button27.Enabled = true;  button17.Enabled = true;  indexOfPanel = 7;  indexOfDataGridView = 6;  nameOfTable = "ОбразованиеСотрудников";  OpenPanels.openPanels(tableLayoutPanel1, indexOfPanel);  dataGridView3.DataSource = образованиеСотрудниковbindingSource;  dataGridView3.Columns["КодОбразования"].HeaderText = "Код образования";  dataGridView3.Columns["РегистрационныйНомерСотрудника"].HeaderText = "Регистрационный номер сотрудника";  dataGridView3.Columns["ГодОкончанияВУЗа"].HeaderText = "Год окончания ВУЗа";  dataGridView3.Columns["УченаяСтепень"].HeaderText = "Ученая степень";  dataGridView3.Columns["УченоеЗвание"].HeaderText = "Ученое звание";  } |
| Кнопка button8\_Click | private void button8\_Click(object sender, EventArgs e)  {  maskedTextBoxРНС1.Text = null;  maskedTextBoxКодРабМеста.Text = null;  maskedTextBoxКолРабСроков.Text = null;  maskedTextBoxДатаНачТрудДеят.Text = null;  maskedTextBoxДатаОкончТрудДеят.Text = null;  maskedTextBoxОклад.Text = null;  } |
| Кнопка button15\_Click | private void button15\_Click(object sender, EventArgs e)  {  maskedTextBoxФамилия.Text = null;  maskedTextBoxИмя.Text = null;  maskedTextBoxОтчество.Text = null;  maskedTextBoxДатаРождения.Text = null;  maskedTextBoxМестоРождения.Text = null;  maskedTextBoxФото.Text = null;  maskedTextBoxПримечания.Text = null;  } |
| Кнопка button18\_Click | private void button18\_Click(object sender, EventArgs e)  {  maskedTextBoxРНС3.Text = null;  maskedTextBoxПаспорт.Text = null;  maskedTextBoxДатаВыдачиПаспорта.Text = null;  maskedTextBoxКемВыдан.Text = null;  maskedTextBoxДомашнийАдрес.Text = null;  maskedTextBoxДомашнийТелефон.Text = null;  maskedTextBoxАдресЭлПочты.Text = null;  } |
| Кнопка button1\_Click | private void button1\_Click(object sender, EventArgs e)  {  maskedTextBoxРНС4.Text = null;  maskedTextBoxНазваниеКафедры.Text = null;  maskedTextBoxЗанимаемаяДолжность.Text = null;  } |
| Кнопка button14\_Click | private void button14\_Click(object sender, EventArgs e)  {  maskedTextBoxРНС5.Text = null;  maskedTextBoxНазваниеПредприятия.Text = null;  maskedTextBoxДолжность.Text = null;  maskedTextBoxДатаУстройства.Text = null;  maskedTextBoxДатаУвольнения.Text = null;  maskedTextBoxПричинаУвольнения.Text = null;  maskedTextBoxАдресПредприятия.Text = null;  maskedTextBoxТелефонПредприятия.Text = null;  } |
| Кнопка button19\_Click | private void button19\_Click(object sender, EventArgs e)  {  maskedTextBoxРНС6.Text = null;  maskedTextBoxСведенияОВзысканиях.Text = null;  maskedTextBoxСведенияОНаграждениях.Text= null;  } |
| Кнопка button20\_Click | private void button20\_Click(object sender, EventArgs e)  {  maskedTextBoxРНС7.Text = null;  maskedTextBoxВУЗ.Text = null;  maskedTextBoxГодОкончанияВУЗа.Text = null;  maskedTextBoxСпециальность.Text= null;  maskedTextBoxУченаяСтепень.Text = null;  maskedTextBoxУченоеЗвание.Text = null;  } |
| Событие button6\_MouseHover | private void button6\_MouseHover(object sender, EventArgs e)  {  toolTip1.Show("Трудовые договоры", button6);  } |
| Событие button2\_MouseHover | private void button2\_MouseHover(object sender, EventArgs e)  {  toolTip1.Show("Сотрудники", button2);  } |
| Событие button3\_MouseHover | private void button3\_MouseHover(object sender, EventArgs e)  {  toolTip1.Show("Личная информация сотрудников", button3);  } |
| Событие button4\_MouseHover | private void button4\_MouseHover(object sender, EventArgs e)  {  toolTip1.Show("Рабочие места", button4);  } |
| Событие button5\_MouseHover | private void button5\_MouseHover(object sender, EventArgs e)  {  toolTip1.Show("Дисциплинарные меры", button5);  } |
| Событие button7\_MouseHover | private void button7\_MouseHover(object sender, EventArgs e)  {  toolTip1.Show("Образование сотрудников", button7);  } |
| Событие button10\_MouseHover | private void button10\_MouseHover(object sender, EventArgs e)  {  toolTip1.Show("Сохранить таблицу", button10);  } |
| Событие button11\_MouseHover | private void button11\_MouseHover(object sender, EventArgs e)  {  toolTip1.Show("Сменить пользователя", button11);  } |
| Событие button9\_MouseHover | private void button9\_MouseHover(object sender, EventArgs e)  {  toolTip1.Show("Закрыть", button9);  } |
| Кнопка button17\_Click\_1 | private void button17\_Click\_1(object sender, EventArgs e)  {  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП\_5\_Гаевский\_ААDataSet.Пользователи". При необходимости она может быть перемещена или удалена.  this.пользователиTableAdapter.Fill(this.уП01\_ИСПП\_5\_Гаевский\_ААDataSet.Пользователи);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ОбразованиеСотрудников". При необходимости она может быть перемещена или удалена.  this.образованиеСотрудниковTableAdapter.Fill(this.уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ОбразованиеСотрудников);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ДисциплинарныеМерыВОтношенииСотрудников". При необходимости она может быть перемещена или удалена.  this.дисциплинарныеМерыВОтношенииСотрудниковTableAdapter.Fill(this.уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ДисциплинарныеМерыВОтношенииСотрудников);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ИнформацияОПредыдущемРабочемМесте". При необходимости она может быть перемещена или удалена.  this.информацияОПредыдущемРабочемМестеTableAdapter.Fill(this.уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ИнформацияОПредыдущемРабочемМесте);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ИнформацияОРабочемМестеСотрудников". При необходимости она может быть перемещена или удалена.  this.информацияОРабочемМестеСотрудниковTableAdapter.Fill(this.уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ИнформацияОРабочемМестеСотрудников);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ЛичнаяИнформацияСотрудников". При необходимости она может быть перемещена или удалена.  this.личнаяИнформацияСотрудниковTableAdapter.Fill(this.уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ЛичнаяИнформацияСотрудников);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП\_5\_Гаевский\_ААDataSet.Сотрудники". При необходимости она может быть перемещена или удалена.  this.сотрудникиTableAdapter.Fill(this.уП01\_ИСПП\_5\_Гаевский\_ААDataSet.Сотрудники);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ТрудовойДоговор". При необходимости она может быть перемещена или удалена.  this.трудовойДоговорTableAdapter.Fill(this.уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ТрудовойДоговор);  switch (indexOfDataGridView)  {  case 0:  foreach (DataGridViewColumn column in dataGridView3.Columns)  {  column.AutoSizeMode = DataGridViewAutoSizeColumnMode.ColumnHeader;  };  break;    case 1:  int imageColumnIndex = 6;  foreach (DataGridViewColumn column in dataGridView3.Columns)  {  if (column.Index != imageColumnIndex)  {  column.AutoSizeMode = DataGridViewAutoSizeColumnMode.ColumnHeader;  }  }  dataGridView3.Columns[imageColumnIndex].Width = 200;  foreach (DataGridViewRow row in dataGridView3.Rows)  {  if (row.Cells[imageColumnIndex].Value is Image image)  {  Image resizedImage = new Bitmap(image, 200, 200);  row.Cells[imageColumnIndex].Value = resizedImage;  }  row.Height = 200 + dataGridView3.RowTemplate.Height - dataGridView3.DefaultCellStyle.Padding.Vertical;  };  break;  case 2:  foreach (DataGridViewColumn column in dataGridView3.Columns)  {  column.AutoSizeMode = DataGridViewAutoSizeColumnMode.ColumnHeader;  };  break;  case 3:  foreach (DataGridViewColumn column in dataGridView3.Columns)  {  column.AutoSizeMode = DataGridViewAutoSizeColumnMode.ColumnHeader;  };  break;  case 4:  foreach (DataGridViewColumn column in dataGridView3.Columns)  {  column.AutoSizeMode = DataGridViewAutoSizeColumnMode.ColumnHeader;  };  break;  }  } |
| Событие button17\_MouseHover | private void button17\_MouseHover(object sender, EventArgs e)  {  toolTip1.Show("Обновить таблицу", button17);  } |
| Кнопка button21\_Click | private void button21\_Click(object sender, EventArgs e)  {  flowLayoutPanel1.Width = widthFlowLayoutPanel;  flowLayoutPanel1.Height = heightFlowLayoutPanel;  x = button21.Location.X + 110;  y = button21.Location.Y + 85; ;  flowLayoutPanel1.Controls.Clear();  flowLayoutPanel1.Location = new Point(x, y);  flowLayoutPanel1.Width = 280;  flowLayoutPanel1.Height = 170;  flowLayoutPanel1.BackColor = Color.White;  flowLayoutPanel1.Visible = true;  Button buttonСотрудникиРаботающиеВНастоящееВремя = new Button();  buttonСотрудникиРаботающиеВНастоящееВремя.Text = "Сотрудники работающие в настоящее время";  buttonСотрудникиРаботающиеВНастоящееВремя.Size = new Size(100, 50);  buttonСотрудникиРаботающиеВНастоящееВремя.Width = flowLayoutPanel1.Width - 5;  buttonСотрудникиРаботающиеВНастоящееВремя.Click += buttonСотрудникиРаботающиеВНастоящееВремя\_Click;  buttonСотрудникиРаботающиеВНастоящееВремя.BackColor = Color.White;  buttonСотрудникиРаботающиеВНастоящееВремя.ForeColor = Color.DarkSlateBlue;  buttonСотрудникиРаботающиеВНастоящееВремя.FlatStyle = FlatStyle.Flat;  buttonСотрудникиРаботающиеВНастоящееВремя.Font = new Font("Microsoft Sans Serif", 10);  flowLayoutPanel1.Controls.Add(buttonСотрудникиРаботающиеВНастоящееВремя);  Button buttonСотрудникиРаботающиеВпервые = new Button();  buttonСотрудникиРаботающиеВпервые.Text = "Сотрудники без опыта";  buttonСотрудникиРаботающиеВпервые.Size = new Size(100, 50);  buttonСотрудникиРаботающиеВпервые.Width = flowLayoutPanel1.Width - 5;  buttonСотрудникиРаботающиеВпервые.Click += buttonСотрудникиРаботающиеВпервые\_Click;  buttonСотрудникиРаботающиеВпервые.BackColor = Color.White;  buttonСотрудникиРаботающиеВпервые.ForeColor = Color.DarkSlateBlue;  buttonСотрудникиРаботающиеВпервые.FlatStyle = FlatStyle.Flat;  buttonСотрудникиРаботающиеВпервые.Font = new Font("Microsoft Sans Serif", 10);  flowLayoutPanel1.Controls.Add(buttonСотрудникиРаботающиеВпервые);  Button buttonСотрудникиКоличествоПрошлыхРабот = new Button();  buttonСотрудникиКоличествоПрошлыхРабот.Text = "Сотрудники и количество их прошлых работ";  buttonСотрудникиКоличествоПрошлыхРабот.Size = new Size(100, 50);  buttonСотрудникиКоличествоПрошлыхРабот.Width = flowLayoutPanel1.Width - 5;  buttonСотрудникиКоличествоПрошлыхРабот.Click += buttonСотрудникиКоличествоПрошлыхРабот\_Click;  buttonСотрудникиКоличествоПрошлыхРабот.BackColor = Color.White;  buttonСотрудникиКоличествоПрошлыхРабот.ForeColor = Color.DarkSlateBlue;  buttonСотрудникиКоличествоПрошлыхРабот.FlatStyle = FlatStyle.Flat;  buttonСотрудникиКоличествоПрошлыхРабот.Font = new Font("Microsoft Sans Serif", 10);  flowLayoutPanel1.Controls.Add(buttonСотрудникиКоличествоПрошлыхРабот);  } |
| Кнопка buttonСотрудникиКоличествоПрошлыхРабот\_Click | private void buttonСотрудникиКоличествоПрошлыхРабот\_Click(object sender, EventArgs e)  {  sqlDataAdapter = new SqlDataAdapter("SELECT Сотрудники.РегистрационныйНомерСотрудника, CONCAT(Сотрудники.Фамилия, ' ', Сотрудники.Имя, ' ', Сотрудники.Отчество) " +  "AS ФИО\_Сотрудника, COUNT(ИнформацияОПредыдущемРабочемМесте.КодПредыдущегоРабочегоМеста) AS КоличествоПредыдущихМест FROM Сотрудники LEFT JOIN ИнформацияОПредыдущемРабочемМесте " +  "ON Сотрудники.РегистрационныйНомерСотрудника = ИнформацияОПредыдущемРабочемМесте.РегистрационныйНомерСотрудника GROUP BY Сотрудники.РегистрационныйНомерСотрудника, " +  "Сотрудники.Фамилия, Сотрудники.Имя, Сотрудники.Отчество HAVING COUNT(ИнформацияОПредыдущемРабочемМесте.КодПредыдущегоРабочегоМеста) > 0 ORDER BY КоличествоПредыдущихМест " +  "DESC; ", sqlConnection);  DataTable dataTable = new DataTable();  sqlDataAdapter.Fill(dataTable);  dataGridView3.DataSource = dataTable;  dataGridView3.Columns["РегистрационныйНомерСотрудника"].HeaderText = "Регистрационный номер сотрудника";  dataGridView3.Columns["ФИО\_Сотрудника"].HeaderText = "ФИО сотрудника";  dataGridView3.Columns["КоличествоПредыдущихМест"].HeaderText = "Количество предыдущих мест";  } |
| Кнопка buttonСотрудникиРаботающиеВпервые\_Click | private void buttonСотрудникиРаботающиеВпервые\_Click(object sender, EventArgs e)  {  sqlDataAdapter = new SqlDataAdapter("SELECT Сотрудники.РегистрационныйНомерСотрудника AS [Регистрационный номер сотрудника], CONCAT(Сотрудники.Фамилия, ' ', Сотрудники.Имя, ' ', Сотрудники.Отчество) " +  "AS [ФИО сотрудника], Дата\_рождения AS [Дата рождения], Место\_рождения AS [Место рождения] FROM Сотрудники WHERE РегистрационныйНомерСотрудника " +  "NOT IN(SELECT РегистрационныйНомерСотрудника FROM ИнформацияОПредыдущемРабочемМесте)", sqlConnection);  DataTable dataTable = new DataTable();  sqlDataAdapter.Fill(dataTable);  dataGridView3.DataSource = dataTable;  } |
| Кнопка buttonСотрудникиРаботающиеВНастоящееВремя\_Click | private void buttonСотрудникиРаботающиеВНастоящееВремя\_Click(object sender, EventArgs e)  {  sqlDataAdapter = new SqlDataAdapter("SELECT Сотрудники.РегистрационныйНомерСотрудника AS [Регистрационный номер сотрудника], CONCAT(Сотрудники.Фамилия, ' ', Сотрудники.Имя, ' ', " +  "Сотрудники.Отчество) AS [ФИО сотрудника], Сотрудники.Дата\_рождения AS [Дата рождения], Сотрудники.Место\_рождения AS [Место рождения]" +  "FROM Сотрудники " +  "WHERE Сотрудники.РегистрационныйНомерСотрудника NOT IN " +  "(SELECT ТрудовойДоговор.РегистрационныйНомерСотрудника FROM ТрудовойДоговор WHERE ТрудовойДоговор.ДатаОкончанияТрудовойДеятельности IS NOT NULL)", sqlConnection);  DataTable dataTable = new DataTable();  sqlDataAdapter.Fill(dataTable);  dataGridView3.DataSource = dataTable;  } |
| Кнопка button22\_Click | private void button22\_Click(object sender, EventArgs e)  {  flowLayoutPanel1.Width = widthFlowLayoutPanel;  flowLayoutPanel1.Height = heightFlowLayoutPanel;  x = button21.Location.X + 110;  y = button21.Location.Y + 85; ;  flowLayoutPanel1.Controls.Clear();  flowLayoutPanel1.Location = new Point(x, y);  flowLayoutPanel1.Width = 280;  flowLayoutPanel1.Height = 70;  flowLayoutPanel1.BackColor = Color.White;  flowLayoutPanel1.Visible = true;  Button buttonСотрудникиОкладВышеСреднего = new Button();  buttonСотрудникиОкладВышеСреднего.Text = "Сотрудники с окладом выше среднего";  buttonСотрудникиОкладВышеСреднего.Size = new Size(100, 50);  buttonСотрудникиОкладВышеСреднего.Width = flowLayoutPanel1.Width - 5;  buttonСотрудникиОкладВышеСреднего.Click += buttonСотрудникиОкладВышеСреднего\_Click;  buttonСотрудникиОкладВышеСреднего.BackColor = Color.White;  buttonСотрудникиОкладВышеСреднего.ForeColor = Color.DarkOrchid;  buttonСотрудникиОкладВышеСреднего.FlatStyle = FlatStyle.Flat;  buttonСотрудникиОкладВышеСреднего.Font = new Font("Microsoft Sans Serif", 10);  flowLayoutPanel1.Controls.Add(buttonСотрудникиОкладВышеСреднего);  } |
| Кнопка buttonСотрудникиОкладВышеСреднего\_Click | private void buttonСотрудникиОкладВышеСреднего\_Click(object sender, EventArgs e)  {  sqlDataAdapter = new SqlDataAdapter("SELECT Сотрудники.РегистрационныйНомерСотрудника AS [Регистрационный номер сотрудника], CONCAT(Сотрудники.Фамилия, ' ', Сотрудники.Имя, ' ', " +  "Сотрудники.Отчество) AS [ФИО сотрудника], ТрудовойДоговор.Оклад FROM Сотрудники INNER JOIN ТрудовойДоговор ON Сотрудники.РегистрационныйНомерСотрудника = " +  "ТрудовойДоговор.РегистрационныйНомерСотрудника WHERE ТрудовойДоговор.Оклад > (SELECT AVG(Оклад) FROM ТрудовойДоговор); ", sqlConnection);  DataTable dataTable = new DataTable();  sqlDataAdapter.Fill(dataTable);  dataGridView3.DataSource = dataTable;  } |
| Кнопка button23\_Click | private void button23\_Click(object sender, EventArgs e)  {  flowLayoutPanel1.Width = widthFlowLayoutPanel;  flowLayoutPanel1.Height = heightFlowLayoutPanel;  x = button21.Location.X + 110;  y = button21.Location.Y + 85; ;  flowLayoutPanel1.Controls.Clear();  flowLayoutPanel1.Location = new Point(x, y);  flowLayoutPanel1.Width = 280;  flowLayoutPanel1.Height = 70;  flowLayoutPanel1.BackColor = Color.White;  flowLayoutPanel1.Visible = true;  Button buttonСотрудникиКоличествоНаграждений = new Button();  buttonСотрудникиКоличествоНаграждений.Text = "Количество награждений";  buttonСотрудникиКоличествоНаграждений.Size = new Size(100, 50);  buttonСотрудникиКоличествоНаграждений.Width = flowLayoutPanel1.Width - 5;  buttonСотрудникиКоличествоНаграждений.Click += buttonСотрудникиКоличествоНаграждений\_Click;  buttonСотрудникиКоличествоНаграждений.BackColor = Color.White;  buttonСотрудникиКоличествоНаграждений.ForeColor = Color.Indigo;  buttonСотрудникиКоличествоНаграждений.FlatStyle = FlatStyle.Flat;  buttonСотрудникиКоличествоНаграждений.Font = new Font("Microsoft Sans Serif", 10);  flowLayoutPanel1.Controls.Add(buttonСотрудникиКоличествоНаграждений);  } |
| Кнопка buttonСотрудникиКоличествоНаграждений\_Click | private void buttonСотрудникиКоличествоНаграждений\_Click(object sender, EventArgs e)  {  button10.Enabled = false;  button27.Enabled = false;  button17.Enabled = false;  sqlDataAdapter = new SqlDataAdapter("SELECT Сотрудники.РегистрационныйНомерСотрудника AS [Регистрационный номер сотрудника], CONCAT(Сотрудники.Фамилия, ' ', Сотрудники.Имя, ' ', " +  "Сотрудники.Отчество) AS [ФИО сотрудника], ИнформацияОРабочемМестеСотрудников.КодРабочегоМеста AS [Код рабочего места], Сотрудники.Фото, " +  "COUNT(ДисциплинарныеМерыВОтношенииСотрудников.КодДисциплинарнойМеры) " +  "AS [Количество награждений] FROM Сотрудники LEFT JOIN ДисциплинарныеМерыВОтношенииСотрудников ON Сотрудники.РегистрационныйНомерСотрудника = " +  "ДисциплинарныеМерыВОтношенииСотрудников.РегистрационныйНомерСотрудника INNER JOIN ИнформацияОРабочемМестеСотрудников ON Сотрудники.РегистрационныйНомерСотрудника = " +  "ИнформацияОРабочемМестеСотрудников.РегистрационныйНомерСотрудника WHERE ДисциплинарныеМерыВОтношенииСотрудников.СведенияОНаграждениях IS NOT NULL GROUP BY " +  "Сотрудники.РегистрационныйНомерСотрудника, Сотрудники.Фамилия, Сотрудники.Имя, Сотрудники.Отчество, Сотрудники.Фото, ИнформацияОРабочемМестеСотрудников.КодРабочегоМеста; ", sqlConnection);  DataTable dataTable = new DataTable();  sqlDataAdapter.Fill(dataTable);  dataGridView3.DataSource = dataTable;  int imageColumnIndex = 3;  int desiredColumnWidth = 200;  int desiredColumnHeight = 200;  foreach (DataGridViewColumn column in dataGridView3.Columns)  {  if (column.Index != imageColumnIndex)  {  column.AutoSizeMode = DataGridViewAutoSizeColumnMode.ColumnHeader;  }  }  dataGridView3.Columns[imageColumnIndex].Width = desiredColumnWidth;  foreach (DataGridViewRow row in dataGridView3.Rows)  {  if (row.Cells[imageColumnIndex].Value is Image image)  {  Image resizedImage = new Bitmap(image, desiredColumnWidth, desiredColumnHeight);  row.Cells[imageColumnIndex].Value = resizedImage;  }  row.Height = desiredColumnHeight + dataGridView3.RowTemplate.Height - dataGridView3.DefaultCellStyle.Padding.Vertical;  }  } |
| Кнопка button26\_Click | private void button26\_Click(object sender, EventArgs e)  {  try  {  SqlCommand sqlCommand = new SqlCommand("INSERT INTO Пользователи (Логин, Пароль, ПраваДоступа) " +  "VALUES(@Логин, @Пароль, @ПраваДоступа)", sqlConnection);  sqlCommand.Parameters.AddWithValue("Логин", maskedTextBoxЛогин.Text);  sqlCommand.Parameters.AddWithValue("Пароль", maskedTextBoxПароль.Text);  sqlCommand.Parameters.AddWithValue("ПраваДоступа", comboBox1.SelectedItem.ToString());  sqlCommand.ExecuteNonQuery();  Notification notification = new Notification();  notification.titleLabel.ForeColor = Color.Orange;  notification.Title = "Добавлен новый пользователь";  notification.Show(3000);  }  catch (Exception ex)  {  MessageBox.Show("Ошибка: " + ex.Message);  }  } |
| Кнопка button27\_Click | private void button27\_Click(object sender, EventArgs e)  {  try  {  if (dataGridView3.SelectedRows.Count > 0)  {  var id = dataGridView3.SelectedRows[0].Cells[0].Value.ToString();  string Message;  Message = "Вы действительно хотите удалить выделенную запись?";  if (MessageBox.Show(Message, "Внимание", MessageBoxButtons.YesNo, MessageBoxIcon.Warning, MessageBoxDefaultButton.Button2) == DialogResult.No)  {  return;  }  var deleQuery = $"DELETE FROM {nameOfTable} WHERE {dataGridView3.Columns[0].Name} = {id}";  SqlCommand sqlCommand = new SqlCommand(deleQuery, sqlConnection);  sqlCommand.ExecuteNonQuery();  Notification notification = new Notification();  notification.titleLabel.ForeColor = Color.Orange;  notification.Title = "Запись удалена, обновите таблицу";  notification.Show(3000);  }  else  {  Notification notification = new Notification();  notification.titleLabel.ForeColor = Color.DarkRed;  notification.Title = "Выберите запись, которую следует удалить";  notification.Show(3000);  }  }  catch (Exception ex)  {  MessageBox.Show($"Произошла ошибка: {ex.Message}", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  } |
| Событие button27\_MouseHover | private void button27\_MouseHover(object sender, EventArgs e)  {  toolTip1.Show("Удалить запись в таблице", button10);  } |
| Событие dataGridView3\_CellContentClick | private void dataGridView3\_CellContentClick(object sender, DataGridViewCellEventArgs e)  {  if (e.ColumnIndex == 6)  {  if (e.RowIndex >= 0 && e.RowIndex < dataGridView3.Rows.Count - 1)  {  var openFileDialog = new OpenFileDialog();  openFileDialog.Filter = "JPEG Files (\*.jpeg;\*.jpg)|\*.jpeg;\*.jpg|PNG Files (\*.png)|\*.png";  if (openFileDialog.ShowDialog() == DialogResult.OK)  {  var selectedFiles = openFileDialog.FileNames;  foreach (var file in selectedFiles)  {  var imageBytes = File.ReadAllBytes(file);  dataGridView3.Rows[e.RowIndex].Cells[e.ColumnIndex].Value = imageBytes;  }  }  }  }  } |
| Событие HideFlowLayoutPanel | private void HideFlowLayoutPanel(object sender, EventArgs e)  {  flowLayoutPanel1.Visible = false;  } |

Свойства элементов формы Рекрутер представлены в таблице 7.

Таблица 7 – Свойства элементов формы «Рекрутер»

|  |  |  |
| --- | --- | --- |
| Объект | Свойство | Значение |
| Рекрутер | BackColor | 192; 192; 255 |
| FormBorderStyle | None |
| Icon | (Значок) |
| MaximizeBox | False |
| MinimizeBox | False |
| StartPosition | CenterScreen |
| Size | 1270; 1114 |
| Text | Мои сотрудники |

Событийно-управляемые процедуры в форме Рекрутер представлены в таблице 8.

Таблица 8 – Событийно-управляемые процедуры в форме «Рекрутер»

| Объект | Событийно-управляемые процедуры |
| --- | --- |
| Класс Recruter | public partial class Recruter: Form  {  public Recruter()  {  InitializeComponent();  }  private SqlConnection sqlConnection = null;  private SqlDataAdapter sqlDataAdapter = null;  private OpenFileDialog openFileDialog = new OpenFileDialog();  private int x, y;  private int indexOfPanel = 7;  private int indexOfDataGridView = 0;  private string imagePath = null;  private string nameOfTable = null;  private const int widthFlowLayoutPanel = 95;  private const int heightFlowLayoutPanel = 60;  } |
| Событие Recruter\_Load | private void Recruter\_Load(object sender, EventArgs e)  {  sqlConnection = new SqlConnection(ConfigurationManager.ConnectionStrings["Отдел\_кадров\_университета.Properties.Settings.УП01\_ИСПП\_5\_Гаевский\_ААConnectionString"].ConnectionString);  sqlConnection.Open();  OpenPanels.openPanels(tableLayoutPanel1, indexOfPanel);  dataGridView3.Click += HideFlowLayoutPanel;  panel1.Click += HideFlowLayoutPanel;  panel2.Click += HideFlowLayoutPanel;  panel3.Click += HideFlowLayoutPanel;  panel4.Click += HideFlowLayoutPanel;  panel5.Click += HideFlowLayoutPanel;  panel6.Click += HideFlowLayoutPanel;  panel7.Click += HideFlowLayoutPanel;  panel8.Click += HideFlowLayoutPanel;  panel9.Click += HideFlowLayoutPanel;  dataGridView3.ColumnHeadersDefaultCellStyle.Font = new Font("Microsoft Sans Serif", 12);  dataGridView3.DefaultCellStyle.Font = new Font("Microsoft Sans Serif", 11);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП\_5\_Гаевский\_ААDataSet.Пользователи". При необходимости она может быть перемещена или удалена.  this.пользователиTableAdapter.Fill(this.уП01\_ИСПП\_5\_Гаевский\_ААDataSet.Пользователи);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ОбразованиеСотрудников". При необходимости она может быть перемещена или удалена.  this.образованиеСотрудниковTableAdapter.Fill(this.уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ОбразованиеСотрудников);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ДисциплинарныеМерыВОтношенииСотрудников". При необходимости она может быть перемещена или удалена.  this.дисциплинарныеМерыВОтношенииСотрудниковTableAdapter.Fill(this.уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ДисциплинарныеМерыВОтношенииСотрудников);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ИнформацияОПредыдущемРабочемМесте". При необходимости она может быть перемещена или удалена.  this.информацияОПредыдущемРабочемМестеTableAdapter.Fill(this.уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ИнформацияОПредыдущемРабочемМесте);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ИнформацияОРабочемМестеСотрудников". При необходимости она может быть перемещена или удалена.  this.информацияОРабочемМестеСотрудниковTableAdapter.Fill(this.уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ИнформацияОРабочемМестеСотрудников);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ЛичнаяИнформацияСотрудников". При необходимости она может быть перемещена или удалена.  this.личнаяИнформацияСотрудниковTableAdapter.Fill(this.уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ЛичнаяИнформацияСотрудников);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП\_5\_Гаевский\_ААDataSet.Сотрудники". При необходимости она может быть перемещена или удалена.  this.сотрудникиTableAdapter.Fill(this.уП01\_ИСПП\_5\_Гаевский\_ААDataSet.Сотрудники);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ТрудовойДоговор". При необходимости она может быть перемещена или удалена.  this.трудовойДоговорTableAdapter.Fill(this.уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ТрудовойДоговор);  } |
| Кнопка button6\_click | private void button6\_Click(object sender, EventArgs e)  {  flowLayoutPanel1.Width = widthFlowLayoutPanel;  flowLayoutPanel1.Height = heightFlowLayoutPanel;  x = button6.Location.X + button6.Size.Width;  y = button6.Location.Y;  flowLayoutPanel1.Controls.Clear();  flowLayoutPanel1.Location = new Point(x, y);  flowLayoutPanel1.Width = 200;  flowLayoutPanel1.BackColor = Color.FromArgb(192, 192, 255);  flowLayoutPanel1.Visible = true;  Button buttonТрудовыеДоговоры = new Button();  buttonТрудовыеДоговоры.Text = "Трудовые договоры";  buttonТрудовыеДоговоры.Width = flowLayoutPanel1.Width - 5;  buttonТрудовыеДоговоры.Click += buttonТрудовыеДоговоры\_Click;  flowLayoutPanel1.Controls.Add(buttonТрудовыеДоговоры);  } |
| Кнопка buttonТрудовыеДоговоры\_Click | private void buttonТрудовыеДоговоры\_Click(object sender, EventArgs e)  {  button10.Enabled = true;  button27.Enabled = true;  button17.Enabled = true;  indexOfPanel = 7;  indexOfDataGridView = 0;  nameOfTable = "ТрудовойДоговор";  OpenPanels.openPanels(tableLayoutPanel1, indexOfPanel);  dataGridView3.DataSource = трудовойДоговорbindingSource;  dataGridView3.Columns["КодТрудовогоДоговора"].HeaderText = "Код трудового договора";  dataGridView3.Columns["РегистрационныйНомерСотрудника"].HeaderText = "Регистрационный номер сотрудника";  dataGridView3.Columns["КодРабочегоМеста"].HeaderText = "Код рабочего места";  dataGridView3.Columns["КоличествоРабочихСроков"].HeaderText = "Количество рабочих сроков";  dataGridView3.Columns["ДатаНачалаТрудовойДеятельности"].HeaderText = "Дата начала трудовой деятельности";  dataGridView3.Columns["ДатаОкончанияТрудовойДеятельности"].HeaderText = "Дата окончания трудовой деятельности";  foreach (DataGridViewColumn column in dataGridView3.Columns)  {  column.AutoSizeMode = DataGridViewAutoSizeColumnMode.ColumnHeader;  }  typeof(DataGridView).InvokeMember("DoubleBuffered", BindingFlags.NonPublic | BindingFlags.Instance | BindingFlags.SetProperty, null, dataGridView3, new object[] { true });  dataGridView3.Scroll += (scrollSender, scrollEventArgs) =>  {  if (scrollEventArgs.ScrollOrientation == ScrollOrientation.VerticalScroll)  {  dataGridView3.FirstDisplayedScrollingRowIndex = scrollEventArgs.NewValue;  }  else if (scrollEventArgs.ScrollOrientation == ScrollOrientation.HorizontalScroll)  {  dataGridView3.FirstDisplayedScrollingColumnIndex = scrollEventArgs.NewValue;  }  };  } |
| Кнопка button2\_Click | private void button2\_Click(object sender, EventArgs e)  {  flowLayoutPanel1.Width = widthFlowLayoutPanel;  flowLayoutPanel1.Height = heightFlowLayoutPanel;  x = button2.Location.X + button2.Size.Width;  y = button2.Location.Y;  flowLayoutPanel1.Controls.Clear();  flowLayoutPanel1.Location = new Point(x, y);  flowLayoutPanel1.Width = 150;  flowLayoutPanel1.BackColor = Color.FromArgb(192, 192, 255);  flowLayoutPanel1.Visible = true;  Button buttonСотрудники = new Button();  buttonСотрудники.Text = "Сотрудники";  buttonСотрудники.Width = flowLayoutPanel1.Width - 5;  buttonСотрудники.BackColor = button2.BackColor;  buttonСотрудники.Click += buttonСотрудники\_Click;  flowLayoutPanel1.Controls.Add(buttonСотрудники);  } |
| Кнопка buttonСотрудники\_Click | private void buttonСотрудники\_Click(object sender, EventArgs e)  {  button10.Enabled = true;  button27.Enabled = true;  button17.Enabled = true;  indexOfPanel = 7;  indexOfDataGridView = 1;  nameOfTable = "Сотрудники";  OpenPanels.openPanels(tableLayoutPanel1, indexOfPanel);  dataGridView3.DataSource = сотрудникиbindingSource;  int imageColumnIndex = 6;  int desiredColumnWidth = 200;  int desiredColumnHeight = 200;  foreach (DataGridViewColumn column in dataGridView3.Columns)  {  if (column.Index != imageColumnIndex)  {  column.AutoSizeMode = DataGridViewAutoSizeColumnMode.ColumnHeader;  }  }  dataGridView3.Columns[imageColumnIndex].Width = desiredColumnWidth;  foreach (DataGridViewRow row in dataGridView3.Rows)  {  if (row.Cells[imageColumnIndex].Value is Image image)  {  Image resizedImage = new Bitmap(image, desiredColumnWidth, desiredColumnHeight);  row.Cells[imageColumnIndex].Value = resizedImage;  }  row.Height = desiredColumnHeight + dataGridView3.RowTemplate.Height - dataGridView3.DefaultCellStyle.Padding.Vertical;  }  dataGridView3.Columns["РегистрационныйНомерСотрудника"].HeaderText = "Регистрационный номер сотрудника";  dataGridView3.Columns["Дата\_рождения"].HeaderText = "Дата рождения";  dataGridView3.Columns["Место\_рождения"].HeaderText = "Место рождения";  } |
| Кнопка button3\_Click | private void button3\_Click(object sender, EventArgs e)  {  flowLayoutPanel1.Width = widthFlowLayoutPanel;  flowLayoutPanel1.Height = heightFlowLayoutPanel;  x = button3.Location.X + button3.Size.Width;  y = button3.Location.Y;  flowLayoutPanel1.Controls.Clear();  flowLayoutPanel1.Location = new Point(x, y);  flowLayoutPanel1.Width = 200;  flowLayoutPanel1.BackColor = Color.FromArgb(192, 192, 255);  flowLayoutPanel1.Visible = true;  Button buttonЛичнаяИнформацияСотрудников = new Button();  buttonЛичнаяИнформацияСотрудников.Text = "Личная ифнормация сотрудников";  buttonЛичнаяИнформацияСотрудников.Width = flowLayoutPanel1.Width - 5;  buttonЛичнаяИнформацияСотрудников.Click += buttonЛичнаяИнформацияСотрудников\_Click;  flowLayoutPanel1.Controls.Add(buttonЛичнаяИнформацияСотрудников);  } |
| Кнопка buttonЛичнаяИнформацияСотрудников\_Click | private void buttonЛичнаяИнформацияСотрудников\_Click(object sender, EventArgs e)  {  button10.Enabled = true;  button27.Enabled = true;  button17.Enabled = true;  indexOfPanel = 7;  indexOfDataGridView = 2;  nameOfTable = "ЛичнаяИнформацияСотрудников";  OpenPanels.openPanels(tableLayoutPanel1, indexOfPanel);  dataGridView3.DataSource = личнаяИнформацияСотрудниковbindingSource;  dataGridView3.Columns["КодЛичнойИнформацииСотрудника"].HeaderText = "Код личной информации сотрудника";  dataGridView3.Columns["РегистрационныйНомерСотрудника"].HeaderText = "Регистрационный номер сотрудника";  dataGridView3.Columns["ДатаВыдачиПаспорта"].HeaderText = "Дата выдачи паспорта";  dataGridView3.Columns["КемВыдан"].HeaderText = "Кем выдан";  dataGridView3.Columns["ДомашнийАдрес"].HeaderText = "Домашний адрес";  dataGridView3.Columns["ДомашнийТелефон"].HeaderText = "Домашний телефон";  dataGridView3.Columns["АдресЭлектроннойПочты"].HeaderText = "Адрес электронной почты";  foreach (DataGridViewColumn column in dataGridView3.Columns)  {  column.AutoSizeMode = DataGridViewAutoSizeColumnMode.ColumnHeader;  }  } |
| Кнопка button4\_Click | private void button4\_Click(object sender, EventArgs e)  {  flowLayoutPanel1.Width = widthFlowLayoutPanel;  flowLayoutPanel1.Height = heightFlowLayoutPanel;  x = button4.Location.X + button4.Size.Width;  y = button4.Location.Y;  flowLayoutPanel1.Controls.Clear();  flowLayoutPanel1.Location = new Point(x, y);  flowLayoutPanel1.Width = 200;  flowLayoutPanel1.Height = 120;  flowLayoutPanel1.BackColor = Color.FromArgb(192, 192, 255);  flowLayoutPanel1.Visible = true;  Button buttonРабочееМестоСотрудника = new Button();  buttonРабочееМестоСотрудника.Text = "Рабочие места";  buttonРабочееМестоСотрудника.Width = flowLayoutPanel1.Width - 5;  buttonРабочееМестоСотрудника.Click += buttonРабочееМестоСотрудника\_Click;  flowLayoutPanel1.Controls.Add(buttonРабочееМестоСотрудника);  Button buttonБывшееРабочееМестоСотрудника = new Button();  buttonБывшееРабочееМестоСотрудника.Text = "Предыдущие рабочие места";  buttonБывшееРабочееМестоСотрудника.Width = flowLayoutPanel1.Width - 5;  buttonБывшееРабочееМестоСотрудника.Click += buttonБывшееРабочееМестоСотрудника\_Click;  flowLayoutPanel1.Controls.Add(buttonБывшееРабочееМестоСотрудника);  } |
| Кнопка buttonБывшееРабочееМестоСотрудника\_Click | private void buttonБывшееРабочееМестоСотрудника\_Click(object sender, EventArgs e)  {  button10.Enabled = true;  button27.Enabled = true;  button17.Enabled = true;  indexOfPanel = 7;  indexOfDataGridView = 3;  nameOfTable = "ИнформацияОПредыдущемРабочемМесте";  OpenPanels.openPanels(tableLayoutPanel1, indexOfPanel);  dataGridView3.DataSource = информацияОПредыдущемРабочемМестеbindingSource;  dataGridView3.Columns["КодПредыдущегоРабочегоМеста"].HeaderText = "Код предыдущего рабочего места";  dataGridView3.Columns["РегистрационныйНомерСотрудника"].HeaderText = "Регистрационный номер сотрудника";  dataGridView3.Columns["НазваниеПредприятия"].HeaderText = "Название предприятия";  dataGridView3.Columns["ДатаУстройства"].HeaderText = "Дата устройства";  dataGridView3.Columns["ДатаУвольнения"].HeaderText = "Дата увольнения";  dataGridView3.Columns["ПричинаУвольнения"].HeaderText = "Причина увольнения";  dataGridView3.Columns["АдресПредприятия"].HeaderText = "Адрес предприятия";  dataGridView3.Columns["ТелефонПредприятия"].HeaderText = "Телефон предприятия";  foreach (DataGridViewColumn column in dataGridView3.Columns)  {  column.AutoSizeMode = DataGridViewAutoSizeColumnMode.ColumnHeader;  }  } |
| Кнопка buttonРабочееМестоСотрудника\_Click | private void buttonРабочееМестоСотрудника\_Click(object sender, EventArgs e)  {  button10.Enabled = true;  button27.Enabled = true;  button17.Enabled = true;  indexOfPanel = 7;  indexOfDataGridView = 4;  nameOfTable = "ИнформацияОРабочемМестеСотрудников";  OpenPanels.openPanels(tableLayoutPanel1, indexOfPanel);  dataGridView3.DataSource = информацияОРабочемМестеСотрудниковbindingSource;  dataGridView3.Columns["КодРабочегоМеста"].HeaderText = "Код рабочего места";  dataGridView3.Columns["РегистрационныйНомерСотрудника"].HeaderText = "Регистрационный номер сотрудника";  dataGridView3.Columns["НазваниеКафедры"].HeaderText = "Название кафедры";  dataGridView3.Columns["ЗанимаемаяДолжность"].HeaderText = "Занимаемая должность";  foreach (DataGridViewColumn column in dataGridView3.Columns)  {  column.AutoSizeMode = DataGridViewAutoSizeColumnMode.ColumnHeader;  }  } |
| Кнопка button5\_Click | private void button5\_Click(object sender, EventArgs e)  {  flowLayoutPanel1.Width = widthFlowLayoutPanel;  flowLayoutPanel1.Height = heightFlowLayoutPanel;  x = button5.Location.X + button5.Size.Width;  y = button5.Location.Y;  flowLayoutPanel1.Controls.Clear();  flowLayoutPanel1.Location = new Point(x, y);  flowLayoutPanel1.Width = 200;  flowLayoutPanel1.BackColor = Color.FromArgb(192, 192, 255);  flowLayoutPanel1.Visible = true;  Button buttonДисциплинарныеМеры = new Button();  buttonДисциплинарныеМеры.Text = "Дисциплинарные меры";  buttonДисциплинарныеМеры.Width = flowLayoutPanel1.Width - 5;  buttonДисциплинарныеМеры.Click += buttonДисциплинарныеМеры\_Click;  flowLayoutPanel1.Controls.Add(buttonДисциплинарныеМеры);  } |
| Кнопка buttonДисциплинарныеМеры\_Click | private void buttonДисциплинарныеМеры\_Click(object sender, EventArgs e)  {  button10.Enabled = true;  button27.Enabled = true;  button17.Enabled = true;  indexOfPanel = 7;  indexOfDataGridView = 5;  nameOfTable = "ДисциплинарныеМерыВОтношенииСотрудников";  OpenPanels.openPanels(tableLayoutPanel1, indexOfPanel);  dataGridView3.DataSource = дисциплинарныеМерыВОтношенииСотрудниковbindingSource;  dataGridView3.Columns["КодДисциплинарнойМеры"].HeaderText = "Код дисциплинарной меры";  dataGridView3.Columns["РегистрационныйНомерСотрудника"].HeaderText = "Регистрационный номер сотрудника";  dataGridView3.Columns["СведенияОВзысканиях"].HeaderText = "Сведения о взысканиях";  dataGridView3.Columns["СведенияОНаграждениях"].HeaderText = "Сведения о награждениях";  } |
| Кнопка button7\_Click | private void button7\_Click(object sender, EventArgs e)  {  flowLayoutPanel1.Width = widthFlowLayoutPanel;  flowLayoutPanel1.Height = heightFlowLayoutPanel;  x = button7.Location.X + button7.Size.Width;  y = button7.Location.Y;  flowLayoutPanel1.Controls.Clear();  flowLayoutPanel1.Location = new Point(x, y);  flowLayoutPanel1.Width = 200;  flowLayoutPanel1.BackColor = Color.FromArgb(192, 192, 255);  flowLayoutPanel1.Visible = true;  Button buttonОбразованиеСотрудников = new Button();  buttonОбразованиеСотрудников.Text = "Образование сотрудников";  buttonОбразованиеСотрудников.Width = flowLayoutPanel1.Width - 5;  buttonОбразованиеСотрудников.Click += buttonОбразованиеСотрудников\_Click;  flowLayoutPanel1.Controls.Add(buttonОбразованиеСотрудников);  } |
| Кнопка button9\_Click | private void button9\_Click(object sender, EventArgs e)  {  Application.Exit();  } |
| Кнопка button11\_Click | private void button11\_Click(object sender, EventArgs e)  {  Hide();  Authorization UserForm = new Authorization();  UserForm.Show();  } |
| Кнопка button13\_Click | private void button13\_Click(object sender, EventArgs e)  {  openFileDialog.Filter = "Изображения (\*.png;\*.jpg)|\*.png;\*.jpg";  openFileDialog.Multiselect = false;  if (openFileDialog.ShowDialog() == DialogResult.OK)  {  imagePath = openFileDialog.FileName;  }  maskedTextBoxФото.Text = imagePath;  } |
| Событие dataGridView3\_DataError | private void dataGridView3\_DataError(object sender, DataGridViewDataErrorEventArgs e)  {  Image defaultImage = Resource1.Noname;  if (e.ColumnIndex == 6 && e.RowIndex >= 0)  {  e.ThrowException = false;  e.Cancel = true;  dataGridView3.Rows[e.RowIndex].Cells[e.ColumnIndex].Value = defaultImage;  }  } |
| Кнопка button12\_Click | private void button12\_Click(object sender, EventArgs e)  {  string searchText = textBox1.Text;  foreach (DataGridViewRow row in dataGridView3.Rows)  {  bool isMatch = false;  foreach (DataGridViewCell cell in row.Cells)  {  if (cell.Value != null && cell.Value.ToString().Contains(searchText))  {  isMatch = true;  break;  }  }  if (isMatch)  {  row.DefaultCellStyle.BackColor = Color.LightBlue;  }  else  {  row.DefaultCellStyle.BackColor = dataGridView3.DefaultCellStyle.BackColor;  }  }  } |
| Событие Recruter\_Click | private void Administrator\_Click(object sender, EventArgs e)  {  if (!flowLayoutPanel1.ClientRectangle.Contains(flowLayoutPanel1.PointToClient(Cursor.Position)))  {  flowLayoutPanel1.Visible = false;  }  } |
| Кнопка button16\_Click | private void button16\_Click(object sender, EventArgs e)  {  foreach (DataGridViewRow row in dataGridView3.Rows)  {  row.DefaultCellStyle.BackColor = dataGridView3.DefaultCellStyle.BackColor;  }  textBox1.Text = string.Empty;  } |
| Кнопка buttonОбразованиеСотрудников\_Click | private void buttonОбразованиеСотрудников\_Click(object sender, EventArgs e)  {  button17.Enabled = true;  indexOfPanel = 7;  indexOfDataGridView = 6;  nameOfTable = "ОбразованиеСотрудников";  OpenPanels.openPanels(tableLayoutPanel1, indexOfPanel);  dataGridView3.DataSource = образованиеСотрудниковbindingSource;  dataGridView3.Columns["КодОбразования"].HeaderText = "Код образования";  dataGridView3.Columns["РегистрационныйНомерСотрудника"].HeaderText = "Регистрационный номер сотрудника";  dataGridView3.Columns["ГодОкончанияВУЗа"].HeaderText = "Год окончания ВУЗа";  dataGridView3.Columns["УченаяСтепень"].HeaderText = "Ученая степень";  dataGridView3.Columns["УченоеЗвание"].HeaderText = "Ученое звание";  } |
| Событие button6\_MouseHover | private void button6\_MouseHover(object sender, EventArgs e)  {  toolTip1.Show("Трудовые договоры", button6);  } |
| Событие button2\_MouseHover | private void button2\_MouseHover(object sender, EventArgs e)  {  toolTip1.Show("Сотрудники", button2);  } |
| Событие button3\_MouseHover | private void button3\_MouseHover(object sender, EventArgs e)  {  toolTip1.Show("Личная информация сотрудников", button3);  } |
| Событие button4\_MouseHover | private void button4\_MouseHover(object sender, EventArgs e)  {  toolTip1.Show("Рабочие места", button4);  } |
| Событие button5\_MouseHover | private void button5\_MouseHover(object sender, EventArgs e)  {  toolTip1.Show("Дисциплинарные меры", button5);  } |
| Событие button7\_MouseHover | private void button7\_MouseHover(object sender, EventArgs e)  {  toolTip1.Show("Образование сотрудников", button7);  } |
| Событие button10\_MouseHover | private void button10\_MouseHover(object sender, EventArgs e)  {  toolTip1.Show("Сохранить таблицу", button10);  } |
| Событие button11\_MouseHover | private void button11\_MouseHover(object sender, EventArgs e)  {  toolTip1.Show("Сменить пользователя", button11);  } |
| Событие button9\_MouseHover | private void button9\_MouseHover(object sender, EventArgs e)  {  toolTip1.Show("Закрыть", button9);  } |
| Кнопка button17\_Click\_1 | private void button17\_Click\_1(object sender, EventArgs e)  {  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП\_5\_Гаевский\_ААDataSet.Пользователи". При необходимости она может быть перемещена или удалена.  this.пользователиTableAdapter.Fill(this.уП01\_ИСПП\_5\_Гаевский\_ААDataSet.Пользователи);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ОбразованиеСотрудников". При необходимости она может быть перемещена или удалена.  this.образованиеСотрудниковTableAdapter.Fill(this.уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ОбразованиеСотрудников);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ДисциплинарныеМерыВОтношенииСотрудников". При необходимости она может быть перемещена или удалена.  this.дисциплинарныеМерыВОтношенииСотрудниковTableAdapter.Fill(this.уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ДисциплинарныеМерыВОтношенииСотрудников);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ИнформацияОПредыдущемРабочемМесте". При необходимости она может быть перемещена или удалена.  this.информацияОПредыдущемРабочемМестеTableAdapter.Fill(this.уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ИнформацияОПредыдущемРабочемМесте);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ИнформацияОРабочемМестеСотрудников". При необходимости она может быть перемещена или удалена.  this.информацияОРабочемМестеСотрудниковTableAdapter.Fill(this.уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ИнформацияОРабочемМестеСотрудников);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ЛичнаяИнформацияСотрудников". При необходимости она может быть перемещена или удалена.  this.личнаяИнформацияСотрудниковTableAdapter.Fill(this.уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ЛичнаяИнформацияСотрудников);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП\_5\_Гаевский\_ААDataSet.Сотрудники". При необходимости она может быть перемещена или удалена.  this.сотрудникиTableAdapter.Fill(this.уП01\_ИСПП\_5\_Гаевский\_ААDataSet.Сотрудники);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ТрудовойДоговор". При необходимости она может быть перемещена или удалена.  this.трудовойДоговорTableAdapter.Fill(this.уП01\_ИСПП\_5\_Гаевский\_ААDataSet.ТрудовойДоговор);  switch (indexOfDataGridView)  {  case 0:  foreach (DataGridViewColumn column in dataGridView3.Columns)  {  column.AutoSizeMode = DataGridViewAutoSizeColumnMode.ColumnHeader;  };  break;    case 1:  int imageColumnIndex = 6;  foreach (DataGridViewColumn column in dataGridView3.Columns)  {  if (column.Index != imageColumnIndex)  {  column.AutoSizeMode = DataGridViewAutoSizeColumnMode.ColumnHeader;  }  }  dataGridView3.Columns[imageColumnIndex].Width = 200;  foreach (DataGridViewRow row in dataGridView3.Rows)  {  if (row.Cells[imageColumnIndex].Value is Image image)  {  Image resizedImage = new Bitmap(image, 200, 200);  row.Cells[imageColumnIndex].Value = resizedImage;  }  row.Height = 200 + dataGridView3.RowTemplate.Height - dataGridView3.DefaultCellStyle.Padding.Vertical;  };  break;  case 2:  foreach (DataGridViewColumn column in dataGridView3.Columns)  {  column.AutoSizeMode = DataGridViewAutoSizeColumnMode.ColumnHeader;  };  break;  case 3:  foreach (DataGridViewColumn column in dataGridView3.Columns)  {  column.AutoSizeMode = DataGridViewAutoSizeColumnMode.ColumnHeader;  };  break;  case 4:  foreach (DataGridViewColumn column in dataGridView3.Columns)  {  column.AutoSizeMode = DataGridViewAutoSizeColumnMode.ColumnHeader;  };  break;  }  } |
| Событие button17\_MouseHover | private void button17\_MouseHover(object sender, EventArgs e)  {  toolTip1.Show("Обновить таблицу", button17);  } |
| Кнопка button21\_Click | private void button21\_Click(object sender, EventArgs e)  {  flowLayoutPanel1.Width = widthFlowLayoutPanel;  flowLayoutPanel1.Height = heightFlowLayoutPanel;  x = button21.Location.X + 110;  y = button21.Location.Y + 85; ;  flowLayoutPanel1.Controls.Clear();  flowLayoutPanel1.Location = new Point(x, y);  flowLayoutPanel1.Width = 280;  flowLayoutPanel1.Height = 170;  flowLayoutPanel1.BackColor = Color.White;  flowLayoutPanel1.Visible = true;  Button buttonСотрудникиРаботающиеВНастоящееВремя = new Button();  buttonСотрудникиРаботающиеВНастоящееВремя.Text = "Сотрудники работающие в настоящее время";  buttonСотрудникиРаботающиеВНастоящееВремя.Size = new Size(100, 50);  buttonСотрудникиРаботающиеВНастоящееВремя.Width = flowLayoutPanel1.Width - 5;  buttonСотрудникиРаботающиеВНастоящееВремя.Click += buttonСотрудникиРаботающиеВНастоящееВремя\_Click;  buttonСотрудникиРаботающиеВНастоящееВремя.BackColor = Color.White;  buttonСотрудникиРаботающиеВНастоящееВремя.ForeColor = Color.DarkSlateBlue;  buttonСотрудникиРаботающиеВНастоящееВремя.FlatStyle = FlatStyle.Flat;  buttonСотрудникиРаботающиеВНастоящееВремя.Font = new Font("Microsoft Sans Serif", 10);  flowLayoutPanel1.Controls.Add(buttonСотрудникиРаботающиеВНастоящееВремя);  Button buttonСотрудникиРаботающиеВпервые = new Button();  buttonСотрудникиРаботающиеВпервые.Text = "Сотрудники без опыта";  buttonСотрудникиРаботающиеВпервые.Size = new Size(100, 50);  buttonСотрудникиРаботающиеВпервые.Width = flowLayoutPanel1.Width - 5;  buttonСотрудникиРаботающиеВпервые.Click += buttonСотрудникиРаботающиеВпервые\_Click;  buttonСотрудникиРаботающиеВпервые.BackColor = Color.White;  buttonСотрудникиРаботающиеВпервые.ForeColor = Color.DarkSlateBlue;  buttonСотрудникиРаботающиеВпервые.FlatStyle = FlatStyle.Flat;  buttonСотрудникиРаботающиеВпервые.Font = new Font("Microsoft Sans Serif", 10);  flowLayoutPanel1.Controls.Add(buttonСотрудникиРаботающиеВпервые);  Button buttonСотрудникиКоличествоПрошлыхРабот = new Button();  buttonСотрудникиКоличествоПрошлыхРабот.Text = "Сотрудники и количество их прошлых работ";  buttonСотрудникиКоличествоПрошлыхРабот.Size = new Size(100, 50);  buttonСотрудникиКоличествоПрошлыхРабот.Width = flowLayoutPanel1.Width - 5;  buttonСотрудникиКоличествоПрошлыхРабот.Click += buttonСотрудникиКоличествоПрошлыхРабот\_Click;  buttonСотрудникиКоличествоПрошлыхРабот.BackColor = Color.White;  buttonСотрудникиКоличествоПрошлыхРабот.ForeColor = Color.DarkSlateBlue;  buttonСотрудникиКоличествоПрошлыхРабот.FlatStyle = FlatStyle.Flat;  buttonСотрудникиКоличествоПрошлыхРабот.Font = new Font("Microsoft Sans Serif", 10);  flowLayoutPanel1.Controls.Add(buttonСотрудникиКоличествоПрошлыхРабот);  } |
| Кнопка buttonСотрудникиКоличествоПрошлыхРабот\_Click | private void buttonСотрудникиКоличествоПрошлыхРабот\_Click(object sender, EventArgs e)  {  sqlDataAdapter = new SqlDataAdapter("SELECT Сотрудники.РегистрационныйНомерСотрудника, CONCAT(Сотрудники.Фамилия, ' ', Сотрудники.Имя, ' ', Сотрудники.Отчество) " +  "AS ФИО\_Сотрудника, COUNT(ИнформацияОПредыдущемРабочемМесте.КодПредыдущегоРабочегоМеста) AS КоличествоПредыдущихМест FROM Сотрудники LEFT JOIN ИнформацияОПредыдущемРабочемМесте " +  "ON Сотрудники.РегистрационныйНомерСотрудника = ИнформацияОПредыдущемРабочемМесте.РегистрационныйНомерСотрудника GROUP BY Сотрудники.РегистрационныйНомерСотрудника, " +  "Сотрудники.Фамилия, Сотрудники.Имя, Сотрудники.Отчество HAVING COUNT(ИнформацияОПредыдущемРабочемМесте.КодПредыдущегоРабочегоМеста) > 0 ORDER BY КоличествоПредыдущихМест " +  "DESC; ", sqlConnection);  DataTable dataTable = new DataTable();  sqlDataAdapter.Fill(dataTable);  dataGridView3.DataSource = dataTable;  dataGridView3.Columns["РегистрационныйНомерСотрудника"].HeaderText = "Регистрационный номер сотрудника";  dataGridView3.Columns["ФИО\_Сотрудника"].HeaderText = "ФИО сотрудника";  dataGridView3.Columns["КоличествоПредыдущихМест"].HeaderText = "Количество предыдущих мест";  } |
| Кнопка buttonСотрудникиРаботающиеВпервые\_Click | private void buttonСотрудникиРаботающиеВпервые\_Click(object sender, EventArgs e)  {  sqlDataAdapter = new SqlDataAdapter("SELECT Сотрудники.РегистрационныйНомерСотрудника AS [Регистрационный номер сотрудника], CONCAT(Сотрудники.Фамилия, ' ', Сотрудники.Имя, ' ', Сотрудники.Отчество) " +  "AS [ФИО сотрудника], Дата\_рождения AS [Дата рождения], Место\_рождения AS [Место рождения] FROM Сотрудники WHERE РегистрационныйНомерСотрудника " +  "NOT IN(SELECT РегистрационныйНомерСотрудника FROM ИнформацияОПредыдущемРабочемМесте)", sqlConnection);  DataTable dataTable = new DataTable();  sqlDataAdapter.Fill(dataTable);  dataGridView3.DataSource = dataTable;  } |
| Кнопка buttonСотрудникиРаботающиеВНастоящееВремя\_Click | private void buttonСотрудникиРаботающиеВНастоящееВремя\_Click(object sender, EventArgs e)  {  sqlDataAdapter = new SqlDataAdapter("SELECT Сотрудники.РегистрационныйНомерСотрудника AS [Регистрационный номер сотрудника], CONCAT(Сотрудники.Фамилия, ' ', Сотрудники.Имя, ' ', " +  "Сотрудники.Отчество) AS [ФИО сотрудника], Сотрудники.Дата\_рождения AS [Дата рождения], Сотрудники.Место\_рождения AS [Место рождения]" +  "FROM Сотрудники " +  "WHERE Сотрудники.РегистрационныйНомерСотрудника NOT IN " +  "(SELECT ТрудовойДоговор.РегистрационныйНомерСотрудника FROM ТрудовойДоговор WHERE ТрудовойДоговор.ДатаОкончанияТрудовойДеятельности IS NOT NULL)", sqlConnection);  DataTable dataTable = new DataTable();  sqlDataAdapter.Fill(dataTable);  dataGridView3.DataSource = dataTable;  } |
| Кнопка button22\_Click | private void button22\_Click(object sender, EventArgs e)  {  flowLayoutPanel1.Width = widthFlowLayoutPanel;  flowLayoutPanel1.Height = heightFlowLayoutPanel;  x = button21.Location.X + 110;  y = button21.Location.Y + 85; ;  flowLayoutPanel1.Controls.Clear();  flowLayoutPanel1.Location = new Point(x, y);  flowLayoutPanel1.Width = 280;  flowLayoutPanel1.Height = 70;  flowLayoutPanel1.BackColor = Color.White;  flowLayoutPanel1.Visible = true;  Button buttonСотрудникиОкладВышеСреднего = new Button();  buttonСотрудникиОкладВышеСреднего.Text = "Сотрудники с окладом выше среднего";  buttonСотрудникиОкладВышеСреднего.Size = new Size(100, 50);  buttonСотрудникиОкладВышеСреднего.Width = flowLayoutPanel1.Width - 5;  buttonСотрудникиОкладВышеСреднего.Click += buttonСотрудникиОкладВышеСреднего\_Click;  buttonСотрудникиОкладВышеСреднего.BackColor = Color.White;  buttonСотрудникиОкладВышеСреднего.ForeColor = Color.DarkOrchid;  buttonСотрудникиОкладВышеСреднего.FlatStyle = FlatStyle.Flat;  buttonСотрудникиОкладВышеСреднего.Font = new Font("Microsoft Sans Serif", 10);  flowLayoutPanel1.Controls.Add(buttonСотрудникиОкладВышеСреднего);  } |
| Кнопка buttonСотрудникиОкладВышеСреднего\_Click | private void buttonСотрудникиОкладВышеСреднего\_Click(object sender, EventArgs e)  {  sqlDataAdapter = new SqlDataAdapter("SELECT Сотрудники.РегистрационныйНомерСотрудника AS [Регистрационный номер сотрудника], CONCAT(Сотрудники.Фамилия, ' ', Сотрудники.Имя, ' ', " +  "Сотрудники.Отчество) AS [ФИО сотрудника], ТрудовойДоговор.Оклад FROM Сотрудники INNER JOIN ТрудовойДоговор ON Сотрудники.РегистрационныйНомерСотрудника = " +  "ТрудовойДоговор.РегистрационныйНомерСотрудника WHERE ТрудовойДоговор.Оклад > (SELECT AVG(Оклад) FROM ТрудовойДоговор); ", sqlConnection);  DataTable dataTable = new DataTable();  sqlDataAdapter.Fill(dataTable);  dataGridView3.DataSource = dataTable;  } |
| Кнопка button23\_Click | private void button23\_Click(object sender, EventArgs e)  {  flowLayoutPanel1.Width = widthFlowLayoutPanel;  flowLayoutPanel1.Height = heightFlowLayoutPanel;  x = button21.Location.X + 110;  y = button21.Location.Y + 85; ;  flowLayoutPanel1.Controls.Clear();  flowLayoutPanel1.Location = new Point(x, y);  flowLayoutPanel1.Width = 280;  flowLayoutPanel1.Height = 70;  flowLayoutPanel1.BackColor = Color.White;  flowLayoutPanel1.Visible = true;  Button buttonСотрудникиКоличествоНаграждений = new Button();  buttonСотрудникиКоличествоНаграждений.Text = "Количество награждений";  buttonСотрудникиКоличествоНаграждений.Size = new Size(100, 50);  buttonСотрудникиКоличествоНаграждений.Width = flowLayoutPanel1.Width - 5;  buttonСотрудникиКоличествоНаграждений.Click += buttonСотрудникиКоличествоНаграждений\_Click;  buttonСотрудникиКоличествоНаграждений.BackColor = Color.White;  buttonСотрудникиКоличествоНаграждений.ForeColor = Color.Indigo;  buttonСотрудникиКоличествоНаграждений.FlatStyle = FlatStyle.Flat;  buttonСотрудникиКоличествоНаграждений.Font = new Font("Microsoft Sans Serif", 10);  flowLayoutPanel1.Controls.Add(buttonСотрудникиКоличествоНаграждений);  } |
| Кнопка buttonСотрудникиКоличествоНаграждений\_Click | private void buttonСотрудникиКоличествоНаграждений\_Click(object sender, EventArgs e)  {  button10.Enabled = false;  button27.Enabled = false;  button17.Enabled = false;  sqlDataAdapter = new SqlDataAdapter("SELECT Сотрудники.РегистрационныйНомерСотрудника AS [Регистрационный номер сотрудника], CONCAT(Сотрудники.Фамилия, ' ', Сотрудники.Имя, ' ', " +  "Сотрудники.Отчество) AS [ФИО сотрудника], ИнформацияОРабочемМестеСотрудников.КодРабочегоМеста AS [Код рабочего места], Сотрудники.Фото, " +  "COUNT(ДисциплинарныеМерыВОтношенииСотрудников.КодДисциплинарнойМеры) " +  "AS [Количество награждений] FROM Сотрудники LEFT JOIN ДисциплинарныеМерыВОтношенииСотрудников ON Сотрудники.РегистрационныйНомерСотрудника = " +  "ДисциплинарныеМерыВОтношенииСотрудников.РегистрационныйНомерСотрудника INNER JOIN ИнформацияОРабочемМестеСотрудников ON Сотрудники.РегистрационныйНомерСотрудника = " +  "ИнформацияОРабочемМестеСотрудников.РегистрационныйНомерСотрудника WHERE ДисциплинарныеМерыВОтношенииСотрудников.СведенияОНаграждениях IS NOT NULL GROUP BY " +  "Сотрудники.РегистрационныйНомерСотрудника, Сотрудники.Фамилия, Сотрудники.Имя, Сотрудники.Отчество, Сотрудники.Фото, ИнформацияОРабочемМестеСотрудников.КодРабочегоМеста; ", sqlConnection);  DataTable dataTable = new DataTable();  sqlDataAdapter.Fill(dataTable);  dataGridView3.DataSource = dataTable;  int imageColumnIndex = 3;  int desiredColumnWidth = 200;  int desiredColumnHeight = 200;  foreach (DataGridViewColumn column in dataGridView3.Columns)  {  if (column.Index != imageColumnIndex)  {  column.AutoSizeMode = DataGridViewAutoSizeColumnMode.ColumnHeader;  }  }  dataGridView3.Columns[imageColumnIndex].Width = desiredColumnWidth;  foreach (DataGridViewRow row in dataGridView3.Rows)  {  if (row.Cells[imageColumnIndex].Value is Image image)  {  Image resizedImage = new Bitmap(image, desiredColumnWidth, desiredColumnHeight);  row.Cells[imageColumnIndex].Value = resizedImage;  }  row.Height = desiredColumnHeight + dataGridView3.RowTemplate.Height - dataGridView3.DefaultCellStyle.Padding.Vertical;  }  } |
| Событие dataGridView3\_CellContentClick | private void dataGridView3\_CellContentClick(object sender, DataGridViewCellEventArgs e)  {  if (e.ColumnIndex == 6)  {  if (e.RowIndex >= 0 && e.RowIndex < dataGridView3.Rows.Count - 1)  {  var openFileDialog = new OpenFileDialog();  openFileDialog.Filter = "JPEG Files (\*.jpeg;\*.jpg)|\*.jpeg;\*.jpg|PNG Files (\*.png)|\*.png";  if (openFileDialog.ShowDialog() == DialogResult.OK)  {  var selectedFiles = openFileDialog.FileNames;  foreach (var file in selectedFiles)  {  var imageBytes = File.ReadAllBytes(file);  dataGridView3.Rows[e.RowIndex].Cells[e.ColumnIndex].Value = imageBytes;  }  }  }  }  } |
| Событие HideFlowLayoutPanel | private void HideFlowLayoutPanel(object sender, EventArgs e)  {  flowLayoutPanel1.Visible = false;  } |

Форма Notification создана для расширения функционала приложения, этот класс выводит форму с уведомлениями.

Свойства элементов формы «Notification» представлены в таблице 9.

Таблица 9 – Свойства элементов формы «Notification»

|  |  |  |
| --- | --- | --- |
| Объект | Свойство | Значение |
| Notification | FormBorderStyle | FixedToolWindow |
| MaximizeBox | False |
| MinimizeBox | False |
| StartPosition | CenterScreen |
| Size | 334; 118 |
| Text | Уведомление |

Событийно-управляемые процедуры в форме «Notification» представлены в таблице 10.

Таблица 10 – Событийно-управляемые процедуры в форме «Notification»

| Объект | Событийно-управляемые процедуры |
| --- | --- |
| Класс Notification | public class Notification : Form  {  public Label titleLabel;  public string Title  {  get { return titleLabel.Text; }  set { titleLabel.Text = value; }  }  public Notification()  {  InitializeComponent();  }  } |
| Событие InitializeComponent | private void InitializeComponent()  {  this.titleLabel = new System.Windows.Forms.Label();  this.SuspendLayout();  //  // titleLabel  //  this.titleLabel.Font = new System.Drawing.Font("Segoe UI", 12F, System.Drawing.FontStyle.Bold);  this.titleLabel.Location = new System.Drawing.Point(10, 8);  this.titleLabel.Name = "titleLabel";  this.titleLabel.Size = new System.Drawing.Size(296, 71);  this.titleLabel.TabIndex = 0;  this.titleLabel.Text = "Заголовок уведомления";  this.titleLabel.Click += new System.EventHandler(this.titleLabel\_Click);  //  // Notification  //  this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);  this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;  this.ClientSize = new System.Drawing.Size(318, 79);  this.Controls.Add(this.titleLabel);  this.FormBorderStyle = System.Windows.Forms.FormBorderStyle.FixedToolWindow;  this.MaximizeBox = false;  this.MinimizeBox = false;  this.Name = "Notification";  this.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen;  this.Text = "Уведомление";  this.ResumeLayout(false);  } |
| Метод Show | public void Show(int duration)  {  // Отображаем форму уведомления  Show();  // Запускаем таймер для закрытия формы через указанное время  Timer timer = new Timer();  timer.Interval = duration;  timer.Tick += (s, e) =>  {  Close();  timer.Stop();  timer.Dispose();  };  timer.Start();  } |

Класс OpenPanels создан для упрощения кода в формах Администратор, Начальник и Рекрутер. Он позволяет выводить элемент управления panel по индексу.

Код класса OpenPanels представлен в приложении Б в листинге кода класса OpenPanels.

Вывод по главе: была полностью спроектирована и разработана система базы данных, были полностью спроектированы и разработаны все программные модули информационной системы “Мои Сотрудники”.

# РАЗРАБОТКА ТЕСТОВЫХ НАБОРОВ И ТЕСТОВЫХ СЦЕНАРИЕВ

# ЗАКЛЮЧЕНИЕ

В результате учебной практики был проведен анализ предметной области, спроектирована и разработана информационная система «Мои Сотрудники», серверная часть которой была реализована в MS SQL, а клиентская часть в MS Visual Studio.

В MS SQL была создана база данных для хранения информации о сотрудниках, их личной информации, образовании, предыдущем месте работы, текущем рабочем месте, выговорах и трудовых договоров.

Клиентская часть, реализованная в MS Visual Studio, отражает все хранящиеся в базе данных таблицы. Также разработаны некоторые запросы, позволяющие редактировать и удалять существующие данные, добавлять новые, осуществлять поиск по таблицам и выводить данные согласно условиям.

Цели учебной практики были достигнуты путем проведения анализа предметной области, проектирования информационной системы и разработки программных модулей.

Это позволяет сделать вывод, что поставленные цели и задачи выполнены в полном объеме.

В дальнейшем данная информационная система может быть модернизирована путем добавления панели личного кабинета, создания версии для слабовидящих людей, улучшения и изменения дизайна, добавления большего количества фильтров вывода информации или изменение панелей добавления записей.

# БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. ГОСТ 20886-85. Организация данных в системах обработки данных. Термины и определения [Электронный доступ]. — Режим доступа: [ГОСТ 20886-85. Организация данных в системах обработки данных. Термины и определения (rostest.info)](https://rostest.info/gost/001.001.040.035/gost-20886-85/#:~:text=%D0%93%D0%9E%D0%A1%D0%A2%2020886%2D85.%20%D0%9E%D1%80%D0%B3%D0%B0%D0%BD%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F%20%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85%20%D0%B2,%D0%BD%D0%B0%D1%83%D1%87%D0%BD%D0%BE%2D%D1%82%D0%B5%D1%85%D0%BD%D0%B8%D1%87%D0%B5%D1%81%D0%) (Дата обращения: 22.05.2013 г.).
2. ГОСТ Р. 59793–2021. Информационные технологии. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания [Электронный доступ]. — Режим доступа: [ГОСТ Р. 59793-2021. Информационные технологии. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания (internet-law.ru)](https://internet-law.ru/gosts/gost/77858/?ysclid=lifap277mg519765664) (Дата обращения: 30.04.2022 г.).
3. ГОСТ 19.201-78. ЕСПД. Общие положения. [Электронный доступ]. — Режим доступа: [ГОСТ 19.201-78.ЕСПД. Техническое задание. Требования к содержанию и оформлению (swrit.ru)](https://www.swrit.ru/doc/espd/19.201-78.pdf?ysclid=lpi5u6naom706376064) (Дата обращения: 01.10.1981 г.).
4. Роберт Э. Уолтерс, Майкл Коулс. SQL Server 2008: ускоренный курс для профессионалов = Accelerated SQL Server 2008. — М: [«Вильямс»](https://ru.wikipedia.org/w/index.php?title=%D0%92%D0%B8%D0%BB%D1%8C%D1%8F%D0%BC%D1%81_(%D0%B8%D0%B7%D0%B4%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D1%82%D0%B2%D0%BE)&action=edit&redlink=1), 2008. — С. 768.
5. Microsoft SQL Server [Электронный доступ]. — Режим доступа: [Microsoft SQL Server — Википедия (wikipedia.org)](https://ru.wikipedia.org/wiki/Microsoft_SQL_Server) (Дата обращения: 10.09.2023 г.).
6. Дебби Валковски. Microsoft Office Visio 2003 для «чайников» = Visio 2003 For Dummies. — М: [«Диалектика»](https://ru.wikipedia.org/w/index.php?title=%D0%94%D0%B8%D0%B0%D0%BB%D0%B5%D0%BA%D1%82%D0%B8%D0%BA%D0%B0_(%D0%B8%D0%B7%D0%B4%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D1%82%D0%B2%D0%BE)&action=edit&redlink=1), 2006. — С. 336.
7. Microsoft Visio [Электронный доступ]. — Режим доступа: [Microsoft Visio — Википедия (wikipedia.org)](https://ru.wikipedia.org/wiki/Microsoft_Visio) (Дата обращения: 07.08.2023 г.).
8. Майо Д. Самоучитель Microsoft Visual Studio 2010 = Microsoft Visual Studio 2010: A Beginner's Guide (A Beginners Guide). — C: [«БХВ-Петербург»](https://ru.wikipedia.org/w/index.php?title=%D0%91%D0%A5%D0%92-%D0%9F%D0%B5%D1%82%D0%B5%D1%80%D0%B1%D1%83%D1%80%D0%B3_(%D0%B8%D0%B7%D0%B4%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D1%82%D0%B2%D0%BE)&action=edit&redlink=1), 2010. — С. 464.
9. Microsoft Visual Studio [Электронный доступ]. — Режим доступа: [Руководство по разработке модулей расширений на C# — Павел Еремеев (pvs-studio.ru)](https://pvs-studio.ru/ru/blog/posts/csharp/a0082/) (Дата обращения: 29.08.2013 г.).

# ПРИЛОЖЕНИЯ А

# 

Рисунок 1 – Диаграмма вариантов использования

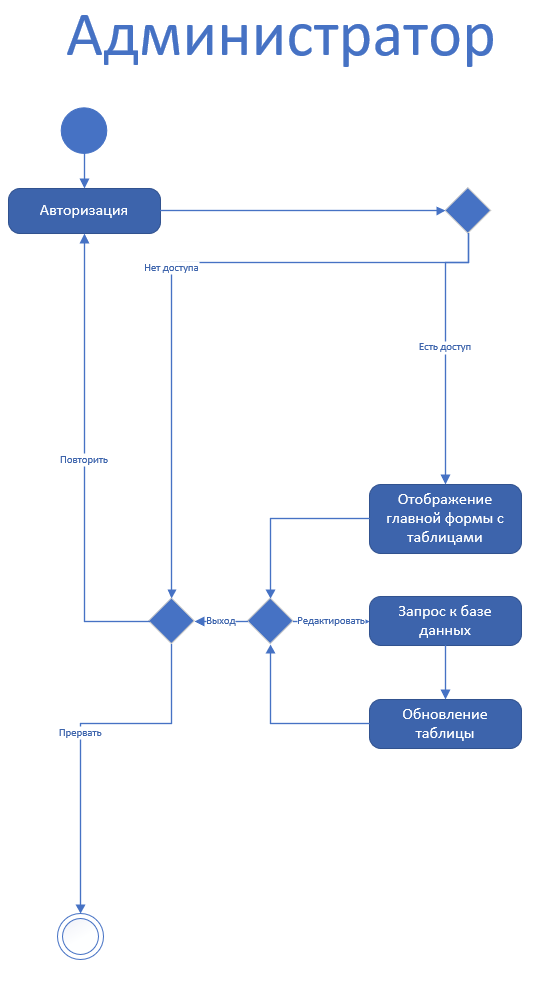


Рисунок 2 – Диаграмма деятельности (Администратор)

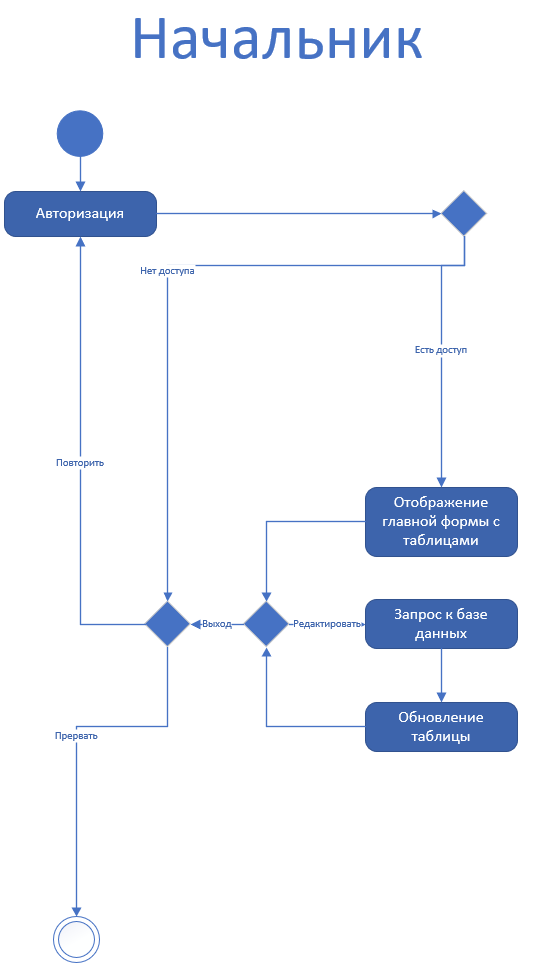


Рисунок 3 – Диаграмма деятельности (Начальник)

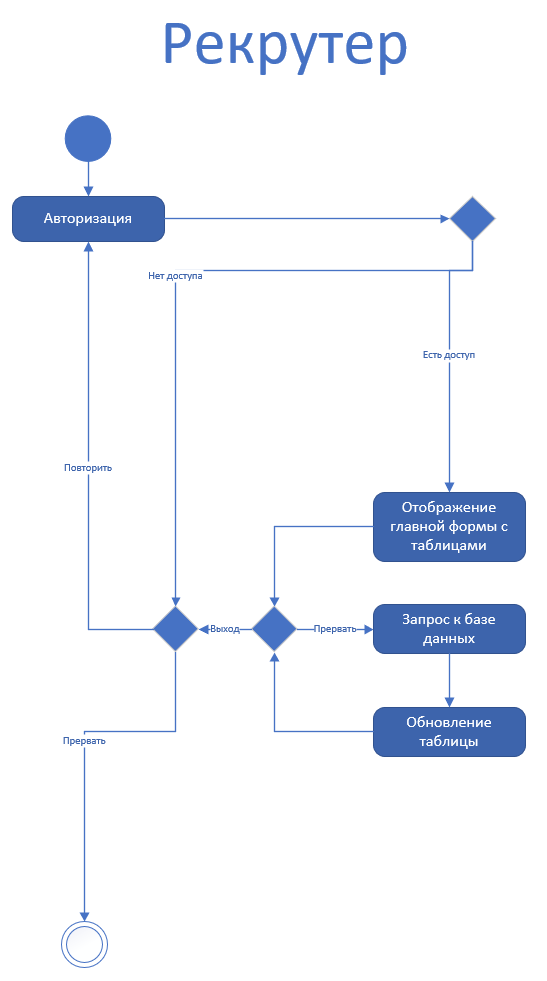


Рисунок 4 – Диаграмма деятельности (Рекрутер)

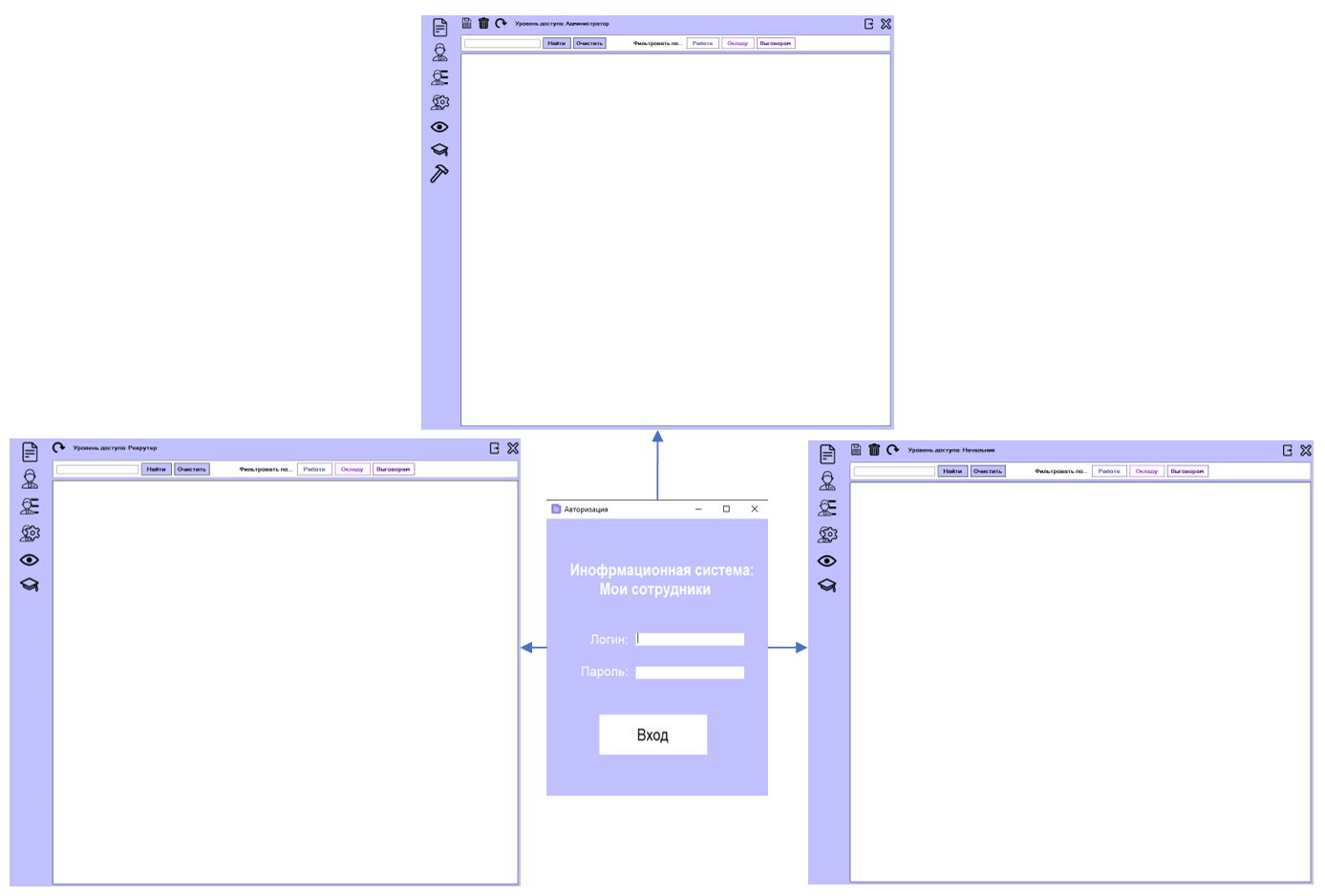


Рисунок 5 – Карта переходов

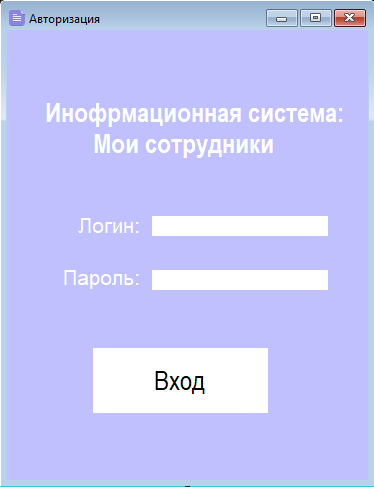


Рисунок 6 – Макет Авторизации

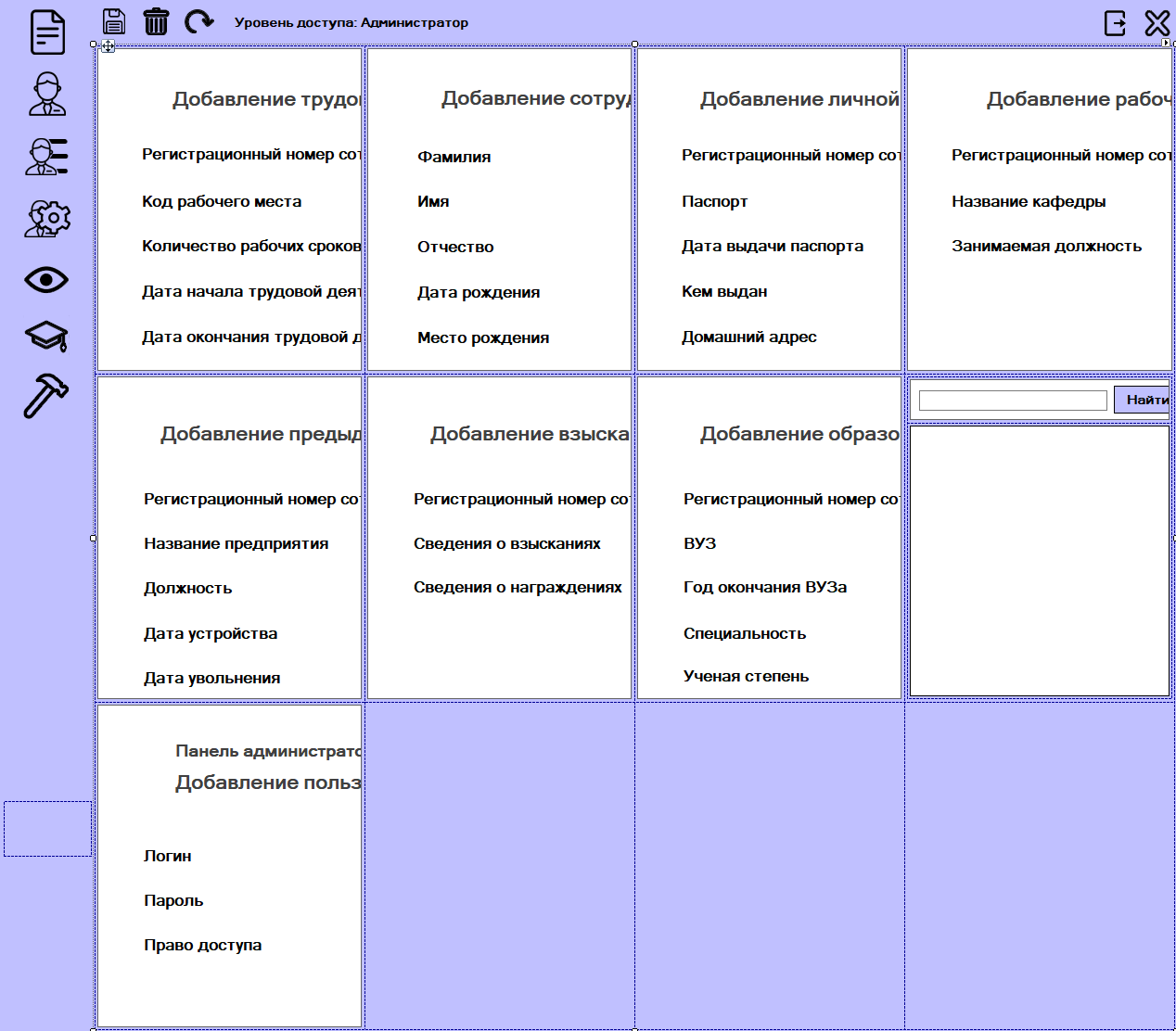


Рисунок 7 – Макет Администратора

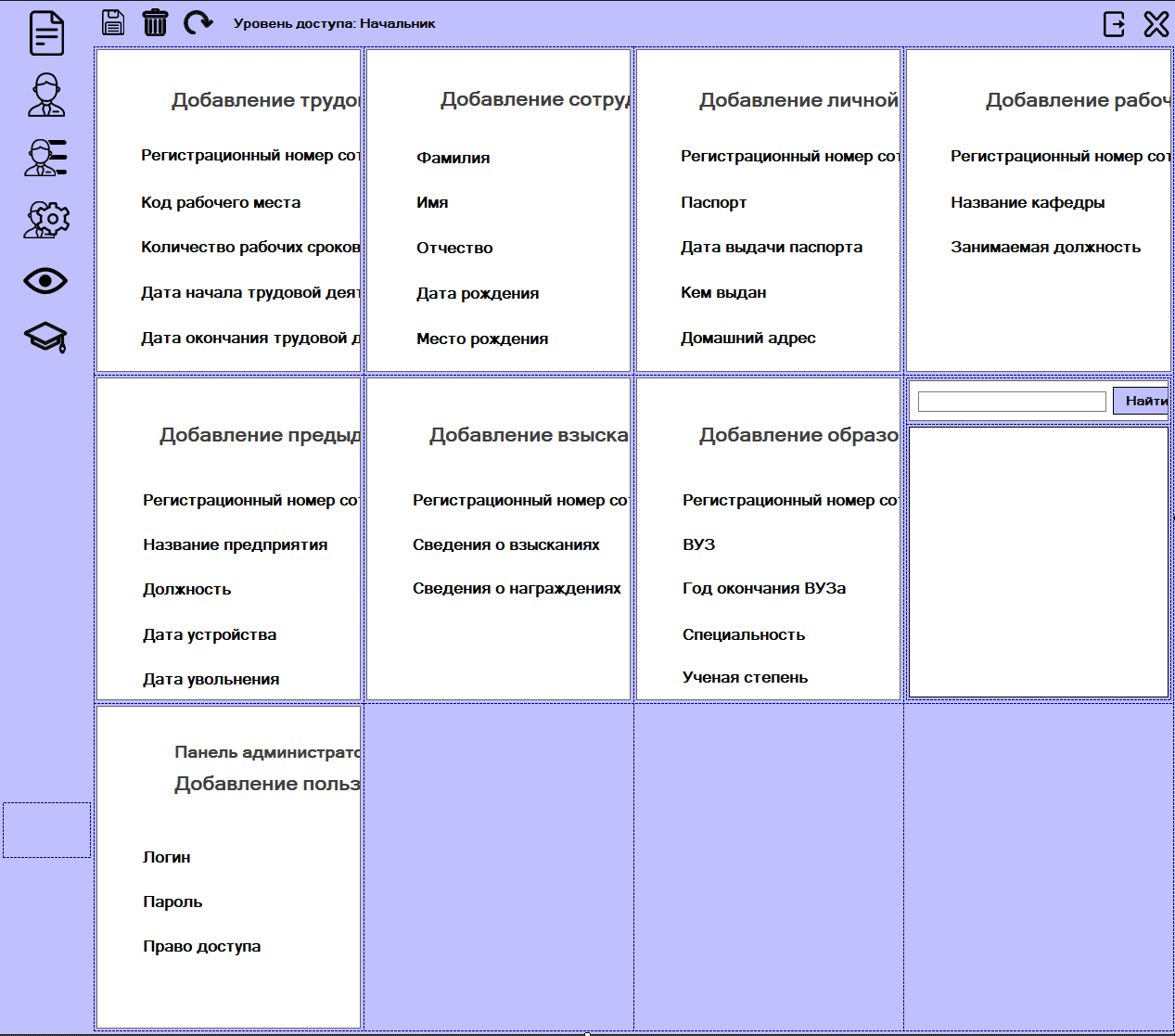


Рисунок 8 – Макет Начальника

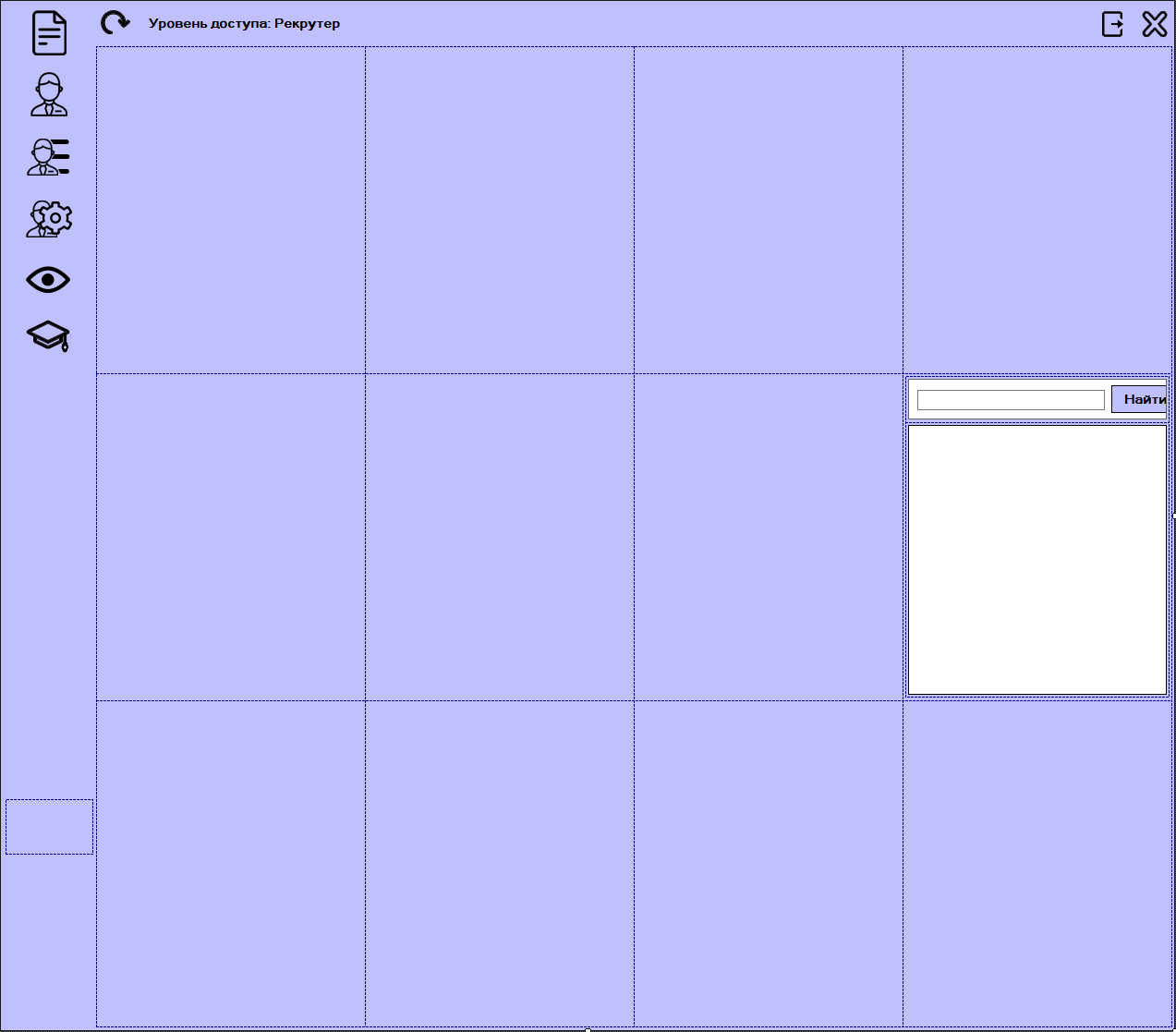


Рисунок 9 – Макет Рекрутера

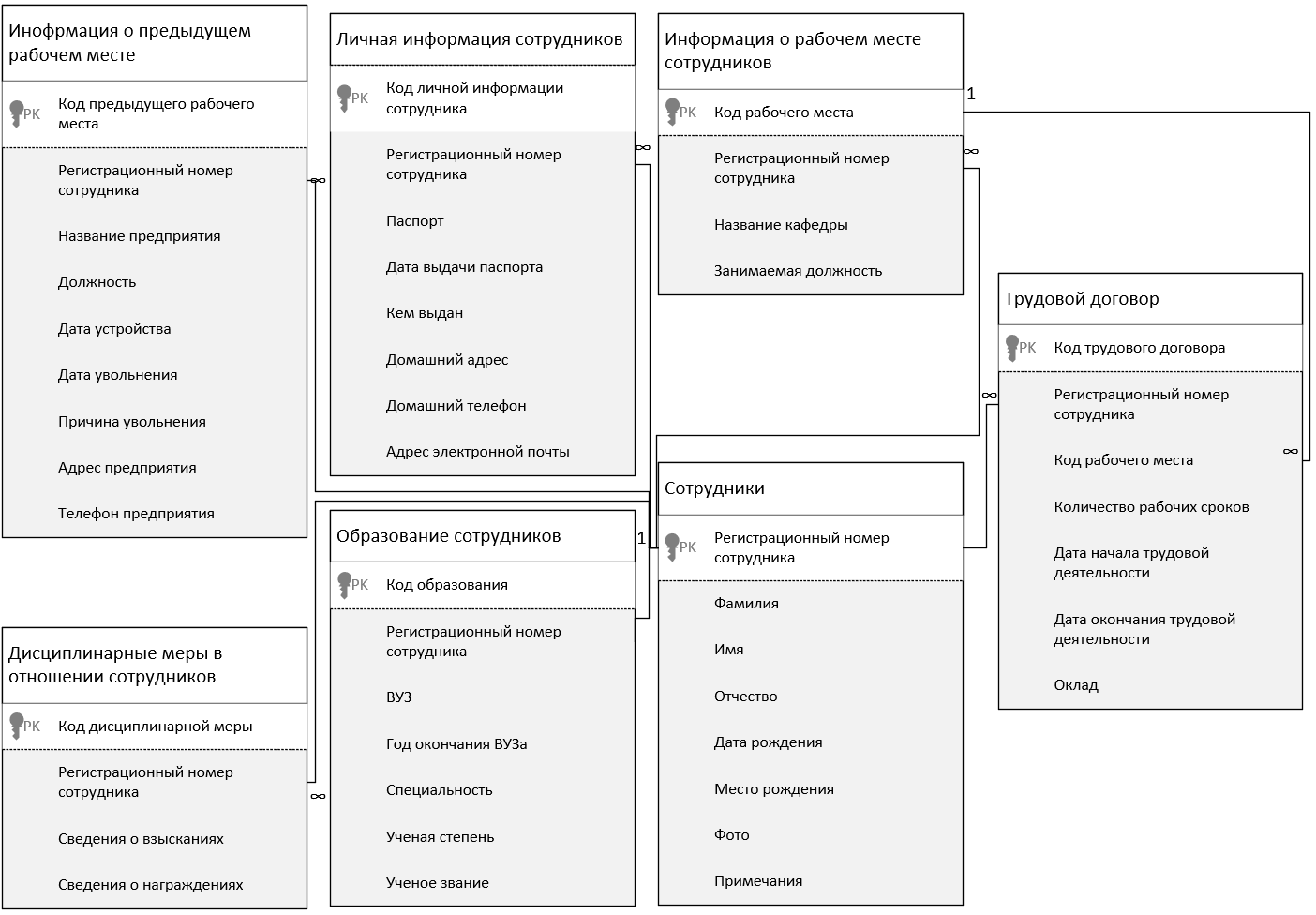


Рисунок 10 – Логическая модель данных

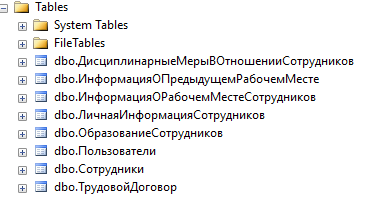


Рисунок 11 – Созданные таблицы

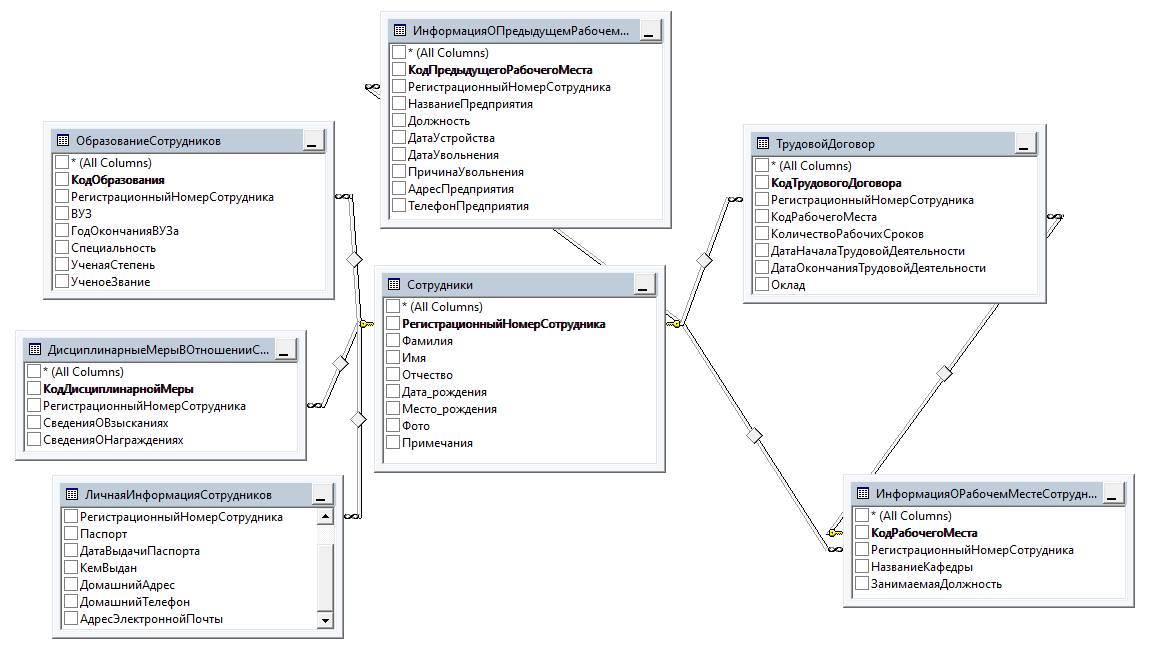


Рисунок 12 – Физическая модель базы данных

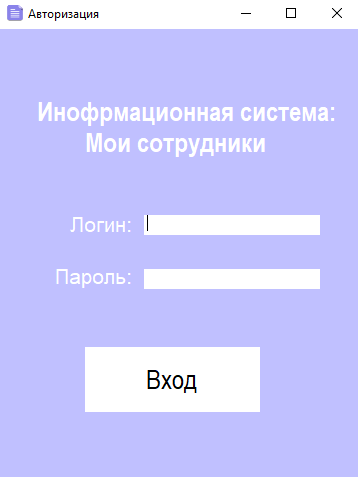


Рисунок 13 – Форма “Авторизация”

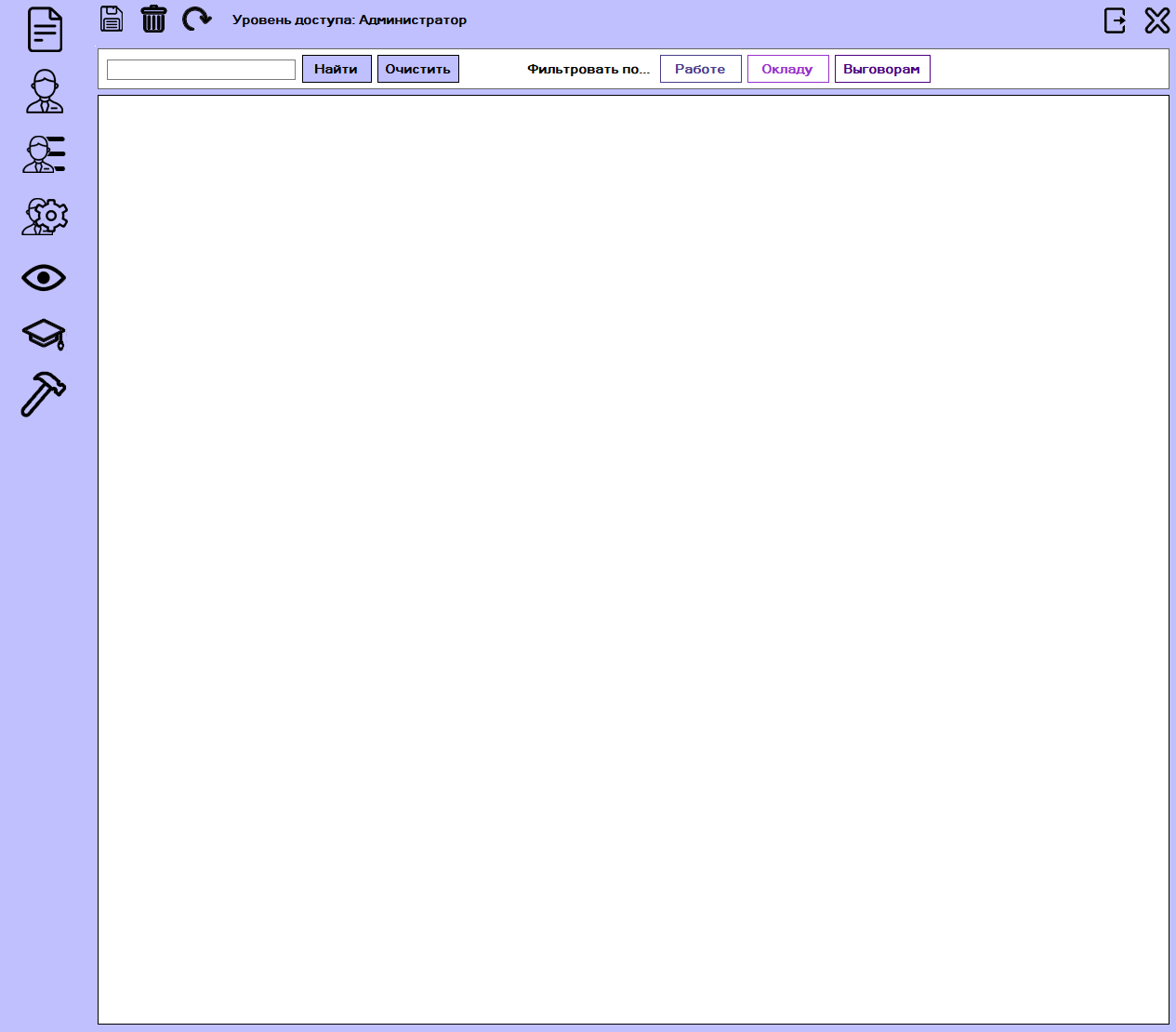


Рисунок 14 – Форма “Администратора”

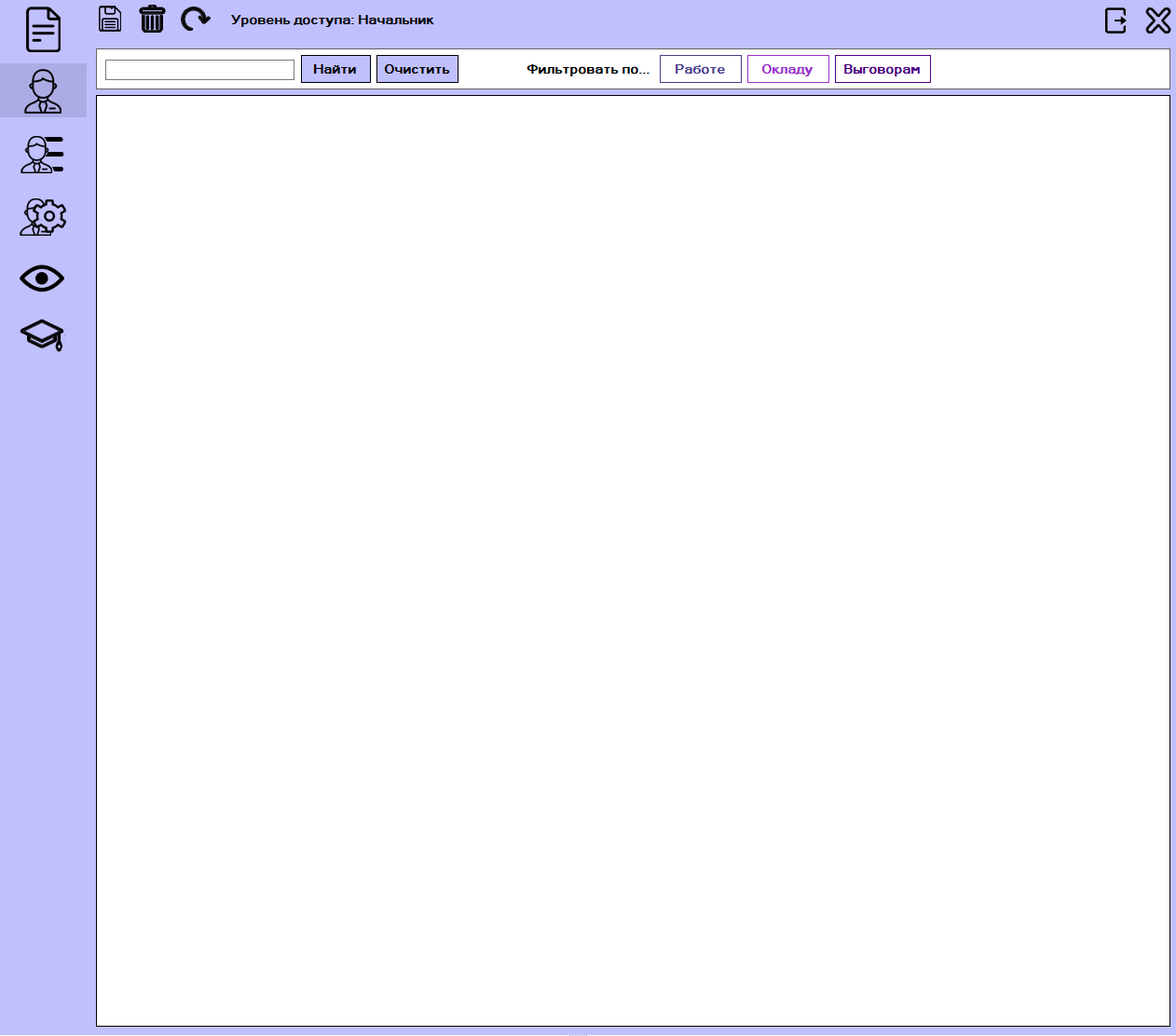


Рисунок 15 – Форма “Начальника”

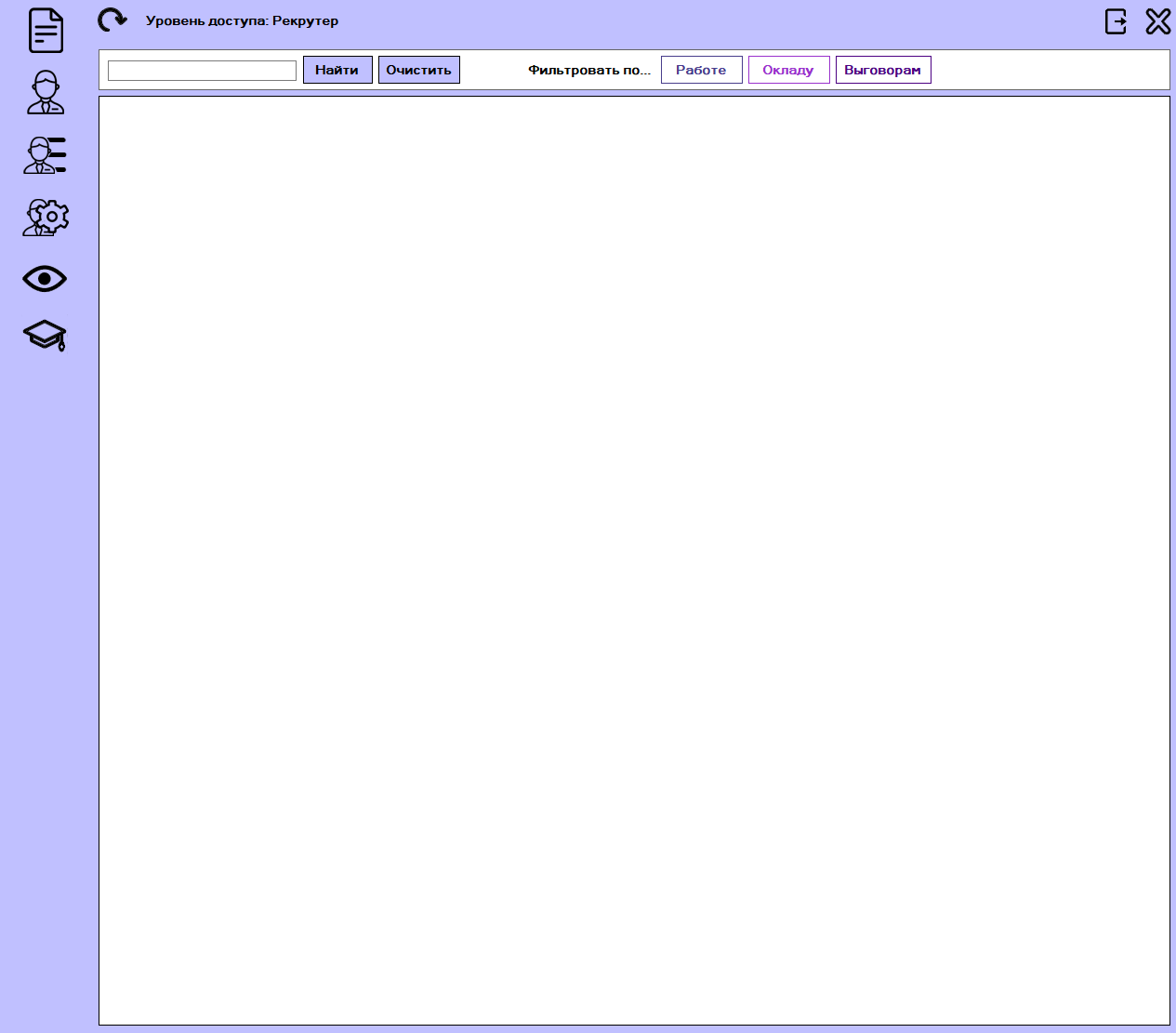


Рисунок 16 – Форма “Рекрутера”

# ПРИЛОЖЕНИЯ Б

Листинг кода cоздания таблиц в MS SQL Server:

CREATE TABLE Сотрудники (

РегистрационныйНомерСотрудника INT IDENTITY(1,1) PRIMARY KEY,

Фамилия VARCHAR(40),

Имя VARCHAR(40),

Отчество VARCHAR(40),

Дата\_рождения DATE,

Место\_рождения VARCHAR(20),

Фото VARBINARY(MAX),

Примечания VARCHAR(MAX)

);

CREATE TABLE ИнформацияОПредыдущемРабочемМесте (

КодПредыдущегоРабочегоМеста INT IDENTITY(1,1) PRIMARY KEY,

РегистрационныйНомерСотрудника INT,

НазваниеПредприятия VARCHAR(100),

Должность VARCHAR(50),

ДатаУстройства DATE,

ДатаУвольнения DATE,

ПричинаУвольнения VARCHAR(100),

АдресПредприятия VARCHAR(200),

ТелефонПредприятия VARCHAR(11),

FOREIGN KEY (РегистрационныйНомерСотрудника) REFERENCES Сотрудники(РегистрационныйНомерСотрудника)

);

CREATE TABLE ЛичнаяИнформацияСотрудников (

КодЛичнойИнформацииСотрудника INT IDENTITY(1,1) PRIMARY KEY,

РегистрационныйНомерСотрудника INT,

Паспорт VARCHAR(14),

ДатаВыдачиПаспорта DATE,

КемВыдан VARCHAR(100),

ДомашнийАдрес VARCHAR(200),

ДомашнийТелефон VARCHAR(11),

АдресЭлектроннойПочты VARCHAR(100),

FOREIGN KEY (РегистрационныйНомерСотрудника) REFERENCES Сотрудники(РегистрационныйНомерСотрудника)

);

CREATE TABLE ДисциплинарныеМерыВОтношенииСотрудников (

КодДисциплинарнойМеры INT IDENTITY(1,1) PRIMARY KEY,

РегистрационныйНомерСотрудника INT,

СведенияОВзысканиях VARCHAR(200),

СведенияОНаграждениях VARCHAR(200),

FOREIGN KEY (РегистрационныйНомерСотрудника) REFERENCES Сотрудники(РегистрационныйНомерСотрудника)

);

CREATE TABLE ОбразованиеСотрудников (

КодОбразования INT IDENTITY(1,1) PRIMARY KEY,

РегистрационныйНомерСотрудника INT,

ВУЗ VARCHAR(100),

ГодОкончанияВУЗа DATE,

Специальность VARCHAR(100),

УченаяСтепень VARCHAR(50),

УченоеЗвание VARCHAR(50),

FOREIGN KEY (РегистрационныйНомерСотрудника) REFERENCES Сотрудники(РегистрационныйНомерСотрудника)

);

CREATE TABLE ИнформацияОРабочемМестеСотрудников (

КодРабочегоМеста INT IDENTITY(1,1) PRIMARY KEY,

РегистрационныйНомерСотрудника INT,

НазваниеКафедры VARCHAR(100),

ЗанимаемаяДолжность VARCHAR(50),

FOREIGN KEY (РегистрационныйНомерСотрудника) REFERENCES Сотрудники(РегистрационныйНомерСотрудника)

);

CREATE TABLE ТрудовойДоговор (

КодТрудовогоДоговора INT IDENTITY(1,1) PRIMARY KEY,

РегистрационныйНомерСотрудника INT,

КодРабочегоМеста INT,

КоличествоРабочихСроков INT,

ДатаНачалаТрудовойДеятельности DATE,

ДатаОкончанияТрудовойДеятельности DATE,

Оклад MONEY,

FOREIGN KEY (РегистрационныйНомерСотрудника) REFERENCES Сотрудники(РегистрационныйНомерСотрудника),

FOREIGN KEY (КодРабочегоМеста) REFERENCES ИнформацияОРабочемМестеСотрудников(КодРабочегоМеста)

);

Листинг кода класса OpenPanels.

namespace Отдел\_кадров\_университета

{

static internal class OpenPanels

{

public static void openPanels(TableLayoutPanel tableLayoutPanel, int indexOfPanel)

{

(int column, int row) = GetIndexColumnAndRowIndexOfPanel(indexOfPanel);

tableLayoutPanel.ColumnStyles[column] = new ColumnStyle(SizeType.Percent, 100);

tableLayoutPanel.RowStyles[row] = new RowStyle(SizeType.Percent, 100);

for (int i = 0; i < tableLayoutPanel.ColumnCount; i++)

{

if (i != column)

{

tableLayoutPanel.ColumnStyles[i] = new ColumnStyle(SizeType.Absolute, 0);

}

}

for (int j = 0; j < tableLayoutPanel.RowCount; j++)

{

if (j != row)

{

tableLayoutPanel.RowStyles[j] = new RowStyle(SizeType.Absolute, 0);

}

}

}

static (int column, int row) GetIndexColumnAndRowIndexOfPanel(int indexOfPanel)

{

int column = 0;

int row = 0;

switch (indexOfPanel)

{

case 0: column = 0; row = 0; break;

case 1: column = 1; row = 0; break;

case 2: column = 2; row = 0; break;

case 3: column = 3; row = 0; break;

case 4: column = 0; row = 1; break;

case 5: column = 1; row = 1; break;

case 6: column = 2; row = 1; break;

case 7: column = 3; row = 1; break;

case 8: column = 0; row = 2; break;

}

return (column, row);

}

}

}