МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ государственное БЮДЖЕТНОЕ

образовательное учреждение

высшего образования

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»



**Отчёт**

**по курсовой работе**

**по дисциплине:** Базы данных

**на тему**:

Разработка фрагмента информационной системы аптечного склада

Выполнил работу:

Студент гр. АП-126, АВТФ

Болотников М.С.

«27» декабря 2023г

Проверил работу:

Эстрайх И.В.

«\_\_\_» \_\_\_\_\_\_ 20\_\_г

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

# Реферат

Название работы: курсовая работа на тему “разработка фрагмента информационной системы аптечного склада”.

Выходные данные: одно приложение, отчёт на 102 страницы, содержащий 31 рисунок и 12 таблиц, 5 используемых источников литературы.

Ключевые слова: база данных, аптечный склад, разработка, access, c#, проектирование, СУБД, приложение.

Объект исследования: процессы хранения и распределения медицинских препаратов, а также ведение учета остатков.

Предмет исследования: разработка приложения.

Цель работы: разработка функционального приложения для управления аптечным складом, которое позволит автоматизировать процессы хранения и распределения медицинских препаратов, а также сократить время и ресурсы, затрачиваемые на учет и контроль аптечного склада.

Задачи: изучение существующих систем управления аптечными складами, анализ требований к функциональности приложения, проектирование структуры базы данных и ее связей с приложением, разработка приложения с использованием выбранных технологий, тестирование и отладка приложения.

Методы исследования (используемые инструменты): язык программирования C#, СУБД Access.

Краткая характеристика полученных результатов: в результате работы была разработана база данных, которая содержит всю необходимую информацию о медикаментах, их характеристиках, поставщиках, ценах и сроках годности. Также были реализованы основные функции приложения, включающие в себя добавление новых товаров, учет поступлений и расходов, формирование отчетов и поиск информации.

# Оглавление

[Введение 4](#_Toc154571233)

[Постановка задачи 5](#_Toc154571234)

[Проектирование базы данных 6](#_Toc154571235)

[Описание процедур доступа и алгоритмов обработки данных 7](#_Toc154571236)

[Описание разработки справочной системы 9](#_Toc154571237)

[Описание разработанного приложения (руководство пользователя) 10](#_Toc154571238)

[Заключение 31](#_Toc154571239)

[Список используемой литературы 32](#_Toc154571240)

[Приложение 33](#_Toc154571241)

# Введение

Современное развитие информационных технологий приводит к необходимости эффективного использования баз данных в различных сферах человеческой деятельности. Одной из таких сфер является управление аптечными складами, где основным предметом работы является хранение и распределение медицинских препаратов.

Актуальность данной работы обусловлена необходимостью автоматизации процессов учета и контроля аптечных складов, что позволит сократить время и ресурсы, затрачиваемые на эти операции, а также увеличит точность ведения учета и избежание ошибок. Результат работы может повысить показатель эффективности компании путём удаления невостребованной работы средством автоматизации.

Объектом исследования являются процессы хранения и распределения медицинских препаратов, а также ведение учета остатков. Также, исследуется процесс заказа пользователем необходимого перечня лекарств и процесс его обработки работником компании. Предметом исследования является разработка приложения, которое позволит автоматизировать данные процессы и обеспечит эффективное управление аптечным складом.

Целью данной работы является разработка функционального приложения для управления аптечным складом, которое позволит автоматизировать процессы хранения и распределения медицинских препаратов, а также сократить время и ресурсы, затрачиваемые на учет и контроль аптечного склада. Для достижения данной цели были поставлены следующие задачи:

- Изучение существующих систем управления аптечными складами;

- Анализ требований к функциональности приложения;

- Проектирование структуры базы данных и ее связей с приложением;

- Разработка приложения с использованием выбранных технологий;

- Тестирование и отладка приложения.

Разработка приложения была выполнена на языке программирования – С#, Исходные документы экспортируются в Microsoft Excel, СУБД была выбрана Microsoft Office Access.

# Постановка задачи

В данной работе необходимо реализовать приложение для учета деятельности аптечного склада, который занимается оптовой продажей лекарств аптекам и медицинским учреждениям. Прежде всего, в базе данных должен регистрироваться имеющийся ассортимент лекарств: название лекарства и его категория; дата производства; срок годности; регистрационный номер Минздрава РФ; данные о производителе; вид упаковки. Кроме того, необходимо иметь сведения по каждому поставщику, который снабжает склад лекарствами: название поставщика; адрес и телефон поставщика; банк поставщика и номер расчетного счета в этом банке; ИНН поставщика.

Поступающие партии лекарств сопровождаются документами – приходными накладными ведомостями. Для каждой поставки в БД заносится: номер приходной накладной ведомости; дата поступления на склад; перечень поступивших лекарств (с указанием цены и количества по каждой позиции).

Для покупателей, которые обращаются на склад за необходимыми им лекарствами, в базе данных должна присутствовать следующая информация: название покупателя; адрес и телефон покупателя; ИНН покупателя. Выдача лекарств покупателю оформляется с помощью счета-фактуры. При регистрации этих документов в БД заносится: номер счета-фактуры; дата выписки; перечень полученных лекарств (с указанием цены и количества по каждой позиции); сумма к уплате; фамилия продавца.

# Проектирование базы данных

Проектирование базы данных состоит из двух этапов. Первый этап – это проектирование концептуальной модели (рис. 1), второй – реляционная модель с описанием структур таблиц (рис. 2).

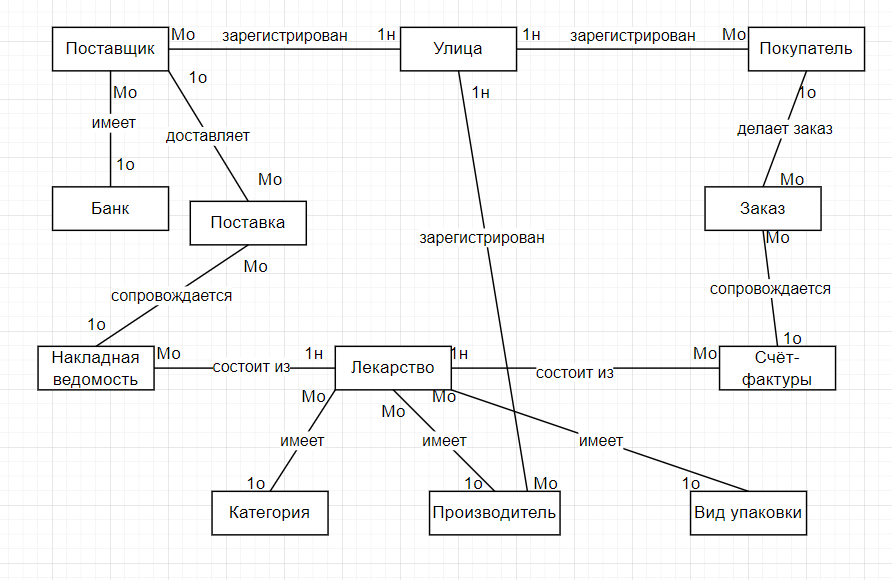


Рисунок 1 - Концептуальная модель для ИС аптечный склад

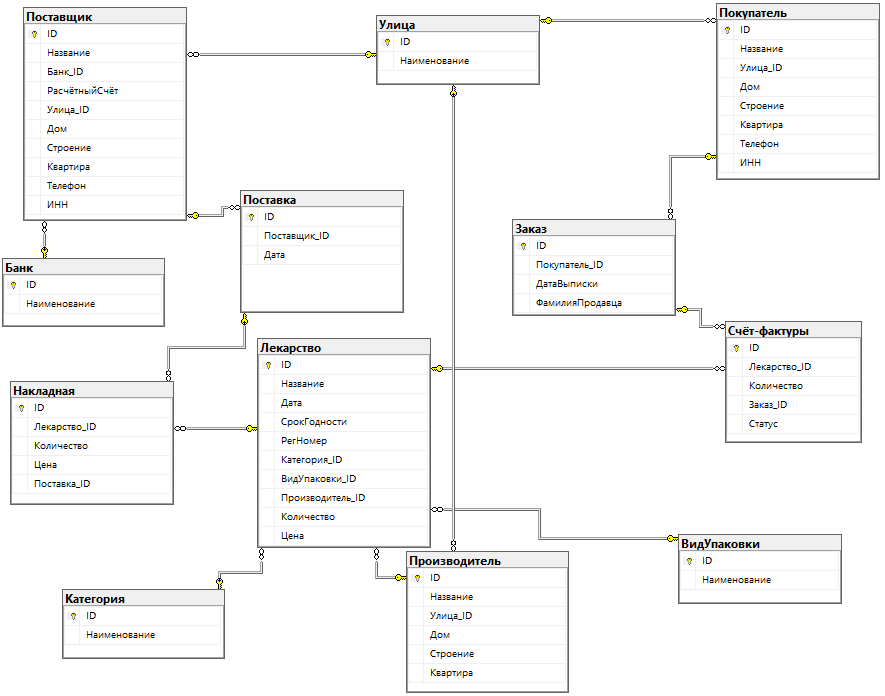


Рисунок 2 – Реляционная модель для ИС аптечный склад

# Описание процедур доступа и алгоритмов обработки данных

В базе данных хранятся 3 таблицы доступа, необходимые для авторизации пользователя. Первая имеет название Users (рис. 3). Она необходима для хранения ID, логина пользователя и его зашифрованного пароля. Шифрование пароля происходит с помощью алгоритма MD5. В таблице Menu (рис. 4) хранится информация о том, есть ли родительский пункт меню, имя меню, имя dll файла, который будет загружаться при нажатии на пункт меню, имя функции из dll файла и порядок, указывающий место пункта меню в приложении. Таблица AccessRights (рис. 5) хранит зависимость между пользователем и каждым пунктом меню. Для каждого пункта меню конкретного пользователя хранится информация о его правах на чтение, запись, редактирование, удаление.

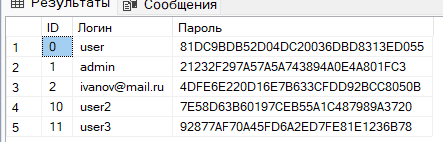


Рисунок 3 – Таблица Users

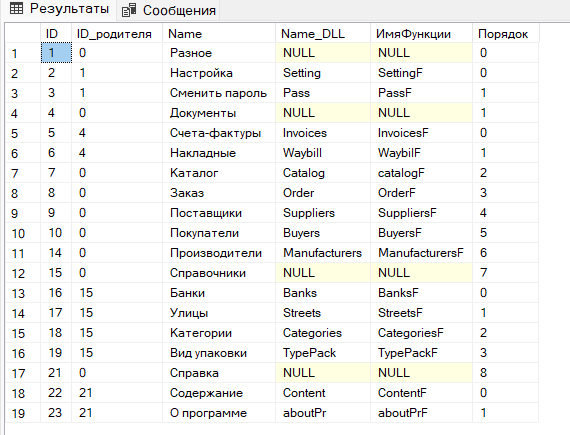


Рисунок 4 – Таблица Menu

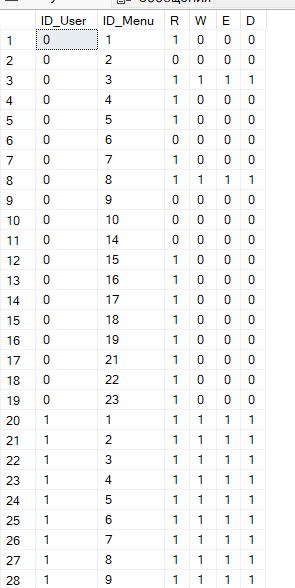


Рисунок 5 – Часть таблицы AccessRights

Обработка данных реализована на языке программирования C# путём создания строки запроса. После этого создаётся экземпляр класса OleDbCommand c двумя параметрами. Первый – это созданная строка. Второй – это вызов метода getconnection у объекта класса DataBase, позволяющий соединиться с базой данных. Далее для чтения использовался метод Read объекта класса OleDbDataReader. Для изменения, удаления, добавления информации в базу данных использовался метод ExecuteNonQuery у экземпляра класса OleDbCommand.

# Описание разработки справочной системы

Справочная система необходима для стандартизации повторяющихся данных. В данной работе такими данными являются: улицы, банки, виды упаковок, категории лекарств. Все эти данные вынесены в раздел справочники, где имеют свои подпункты. Каждая справочная информация представляет собой ID и наименование. Каждый справочник можно добавлять, удалять, редактировать при наличии соответствующих прав доступа. Наличие такой системы обусловлена стандартизацией выбора для пользователя. Например, пользователь не вводит название улицы, а выбирает её из выпадающего списка вариантов.

# Описание разработанного приложения (руководство пользователя)

Разработанное приложение предназначено для удобного ведения учёта аптечного склада. Приложение имеет функционал для формирования документов накладных, счёт-фактур. Также в нём доступны возможности такие как: оформление заказа, смена пароля, изменение прав доступа, изменение каталога, изменение информации о поставщиках, изменение информации о покупателях, изменение информации о производителях, изменение справочной информации. Документы можно экспортировать в файл Excel. Разработка велась на языке программирования C# с использованием элементов Windows Forms.

При запуске приложения пользователю предлагается окно авторизации (рис. 6). В нижней части экрана отображается текущий язык и активна ли клавиша CapsLock.

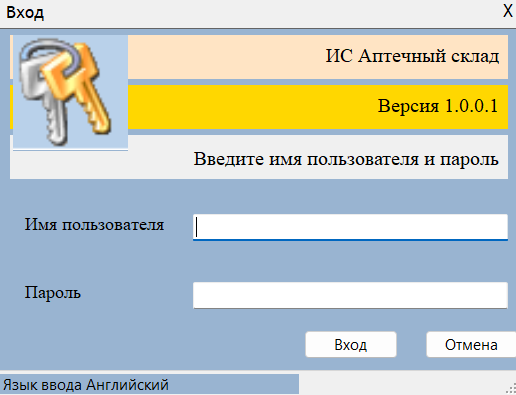


Рисунок 6 – Окно авторизации

При корректном вводе логина и пароля открывается основное меню (рис. 7). Для тестирования работы приложения со всеми правами можно ввести логин: admin, пароль: admin. Для тестирования приложения с ограниченными правами необходимо ввести логин: user, пароль: 1234. При ложном вводе отображается ошибка (рис. 8).

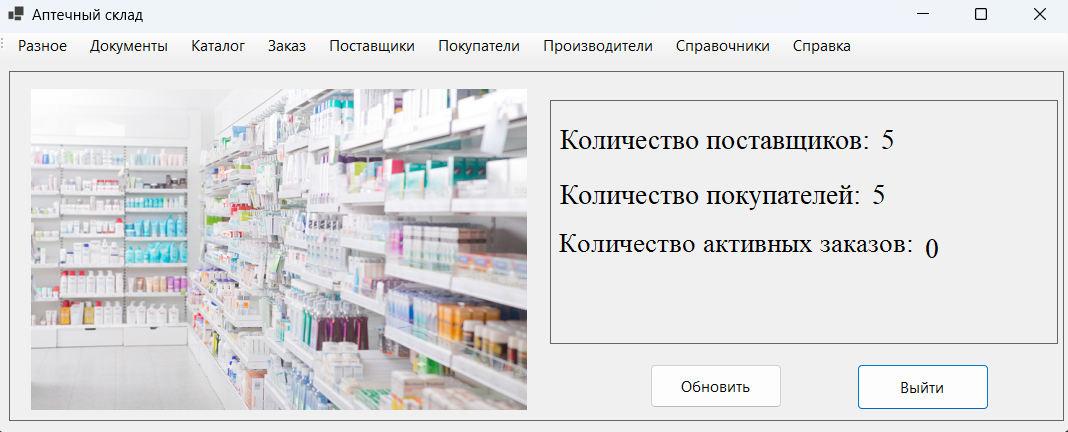


Рисунок 7 – Основное окно приложения

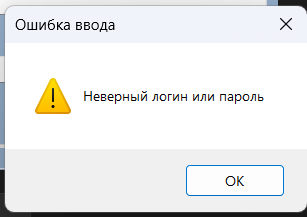


Рисунок 8 – Ошибка ввода

В основном окне показывается информация о количестве покупателей, поставщиков и активных заказов. При нажатии на кнопку “Обновить” данные три значения обновятся. Кнопка “Выйти” позволяет пользователю закрыть приложение.

При нажатии на пункт меню подгружается dll файл с дополнительным функционалом или появляются подпункты основного меню (рис. 9).

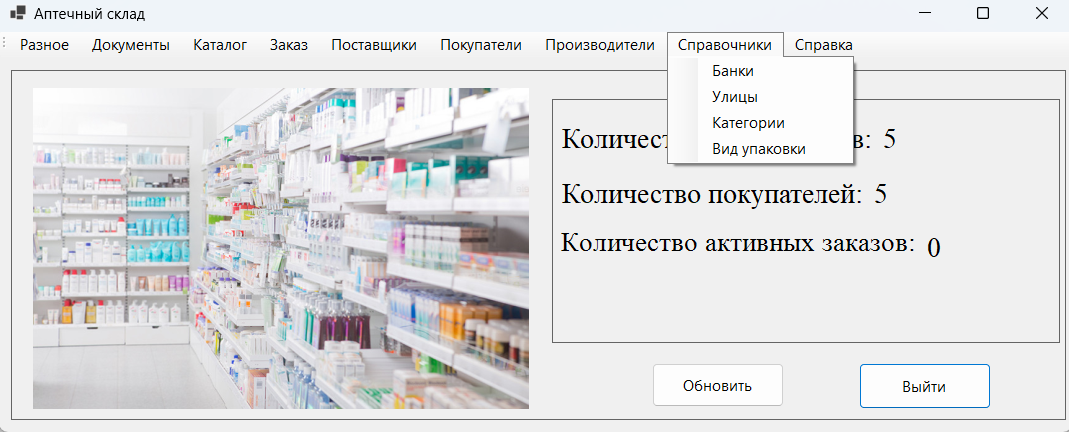


Рисунок 9 – Подпункты меню справочники

В меню справочники находятся подпункты “Банки” (рис. 10), “Улицы” (рис. 11), “Категории” (рис. 12), “Вид упаковки” (рис. 13). Функционал справочников одинаковый. Кнопка “Отчистка” удаляет данные из панели “Запись”. Кнопка “Обновить” также удаляет данные из панели “Запись” и обновляет данные в таблице. При нажатии на любую ячейку таблицы в панель “Запись” добавляются данные с текущей строки таблицы. “Новая запись” позволяет создать новую строку в базе данных (рис. 18). “Изменить” – редактирует данные в таблице в соответствии с данными в панели “Запись”. “Удалить” – вырезает активную строку из таблицы. “Сохранить” – изменяет таблицу в базе данных в соответствии с таблицей в приложении. При нажатии на название столбцов происходит сортировка по выбранному столбцу. Поиск необходим для нахождения строки по любым данным таблицы.

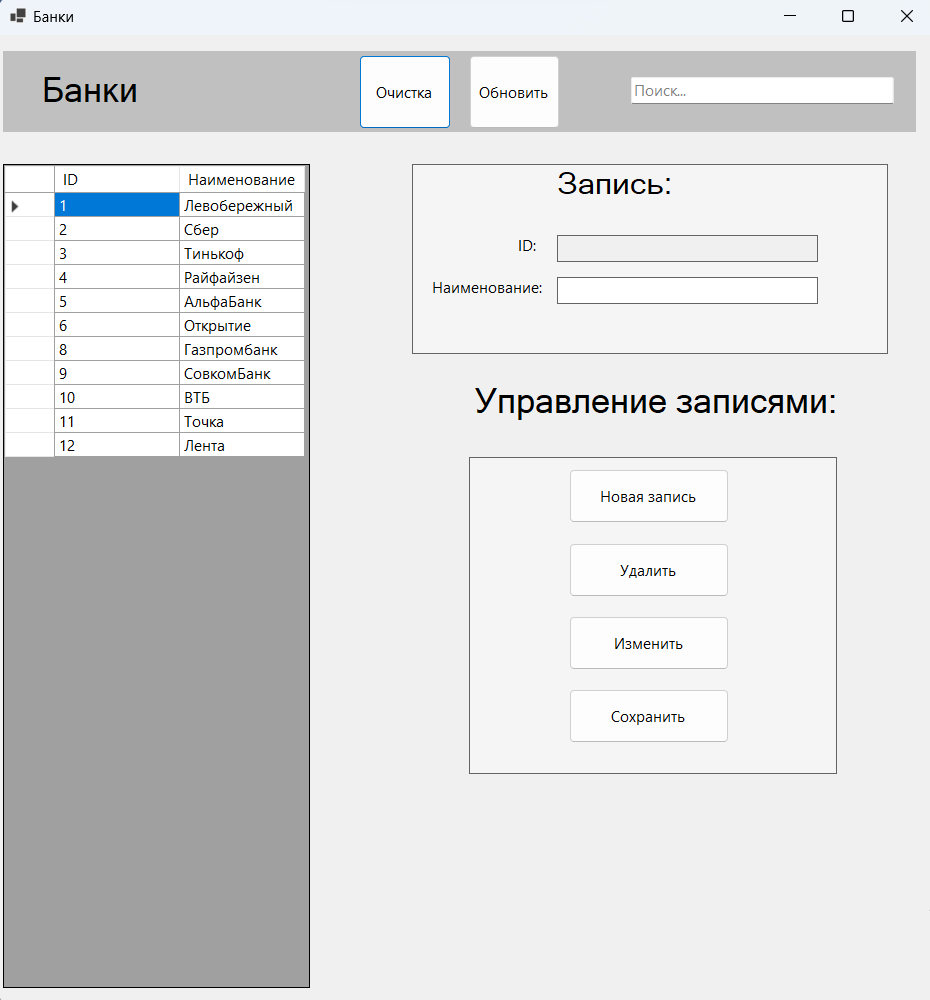


Рисунок 10 – интерфейс пункта “Банки”

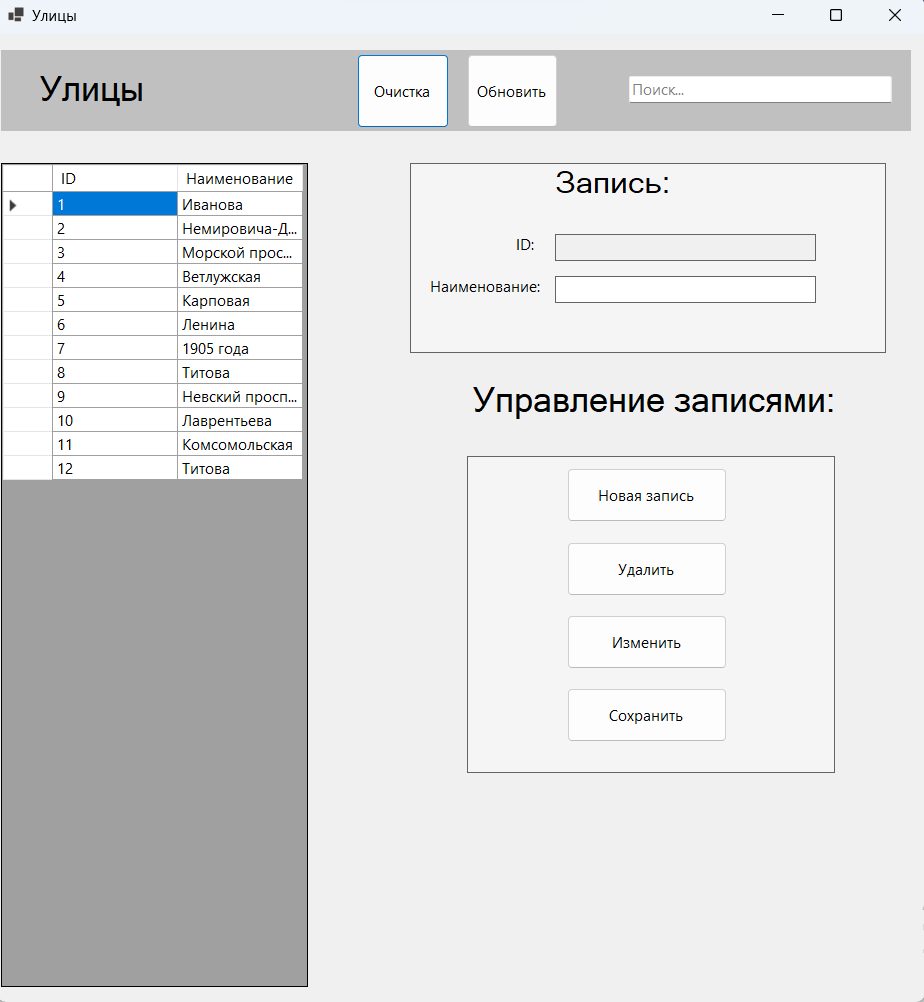


Рисунок 11 – интерфейс пункта “Улицы”

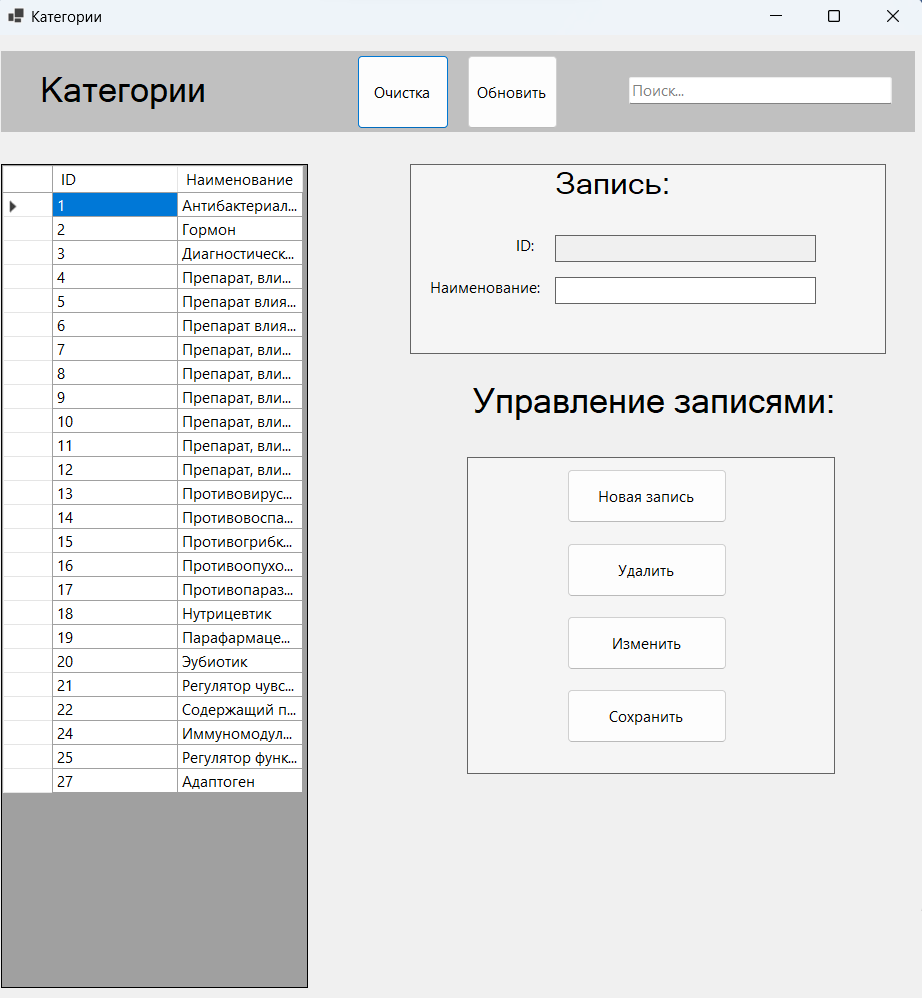


Рисунок 12 – интерфейс пункта “Категории”

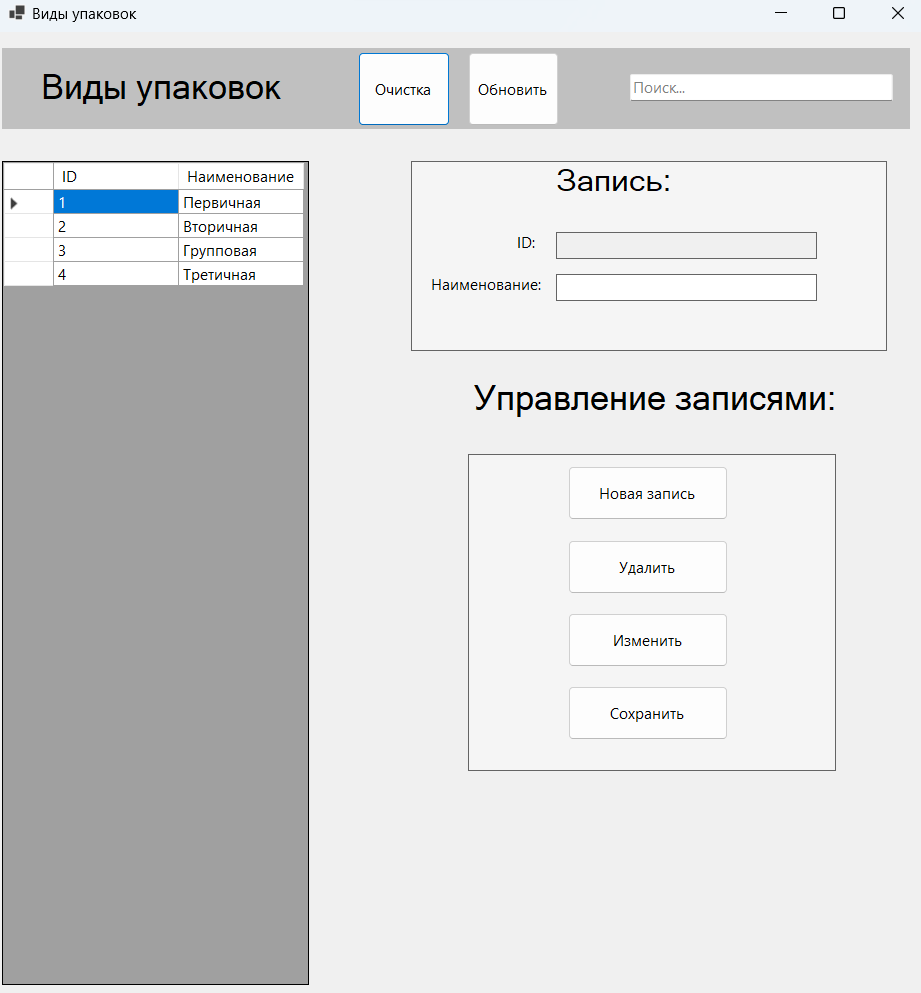


Рисунок 13 – интерфейс пункта “Вид упаковок”

Функционал пунктов меню “Производитель” (рис. 14), “Покупатель” (рис. 15), “Поставщик” (рис. 16), “Каталог” (рис. 17) не отличается от предыдущего. Различие заключается только в графическом интерфейсе.

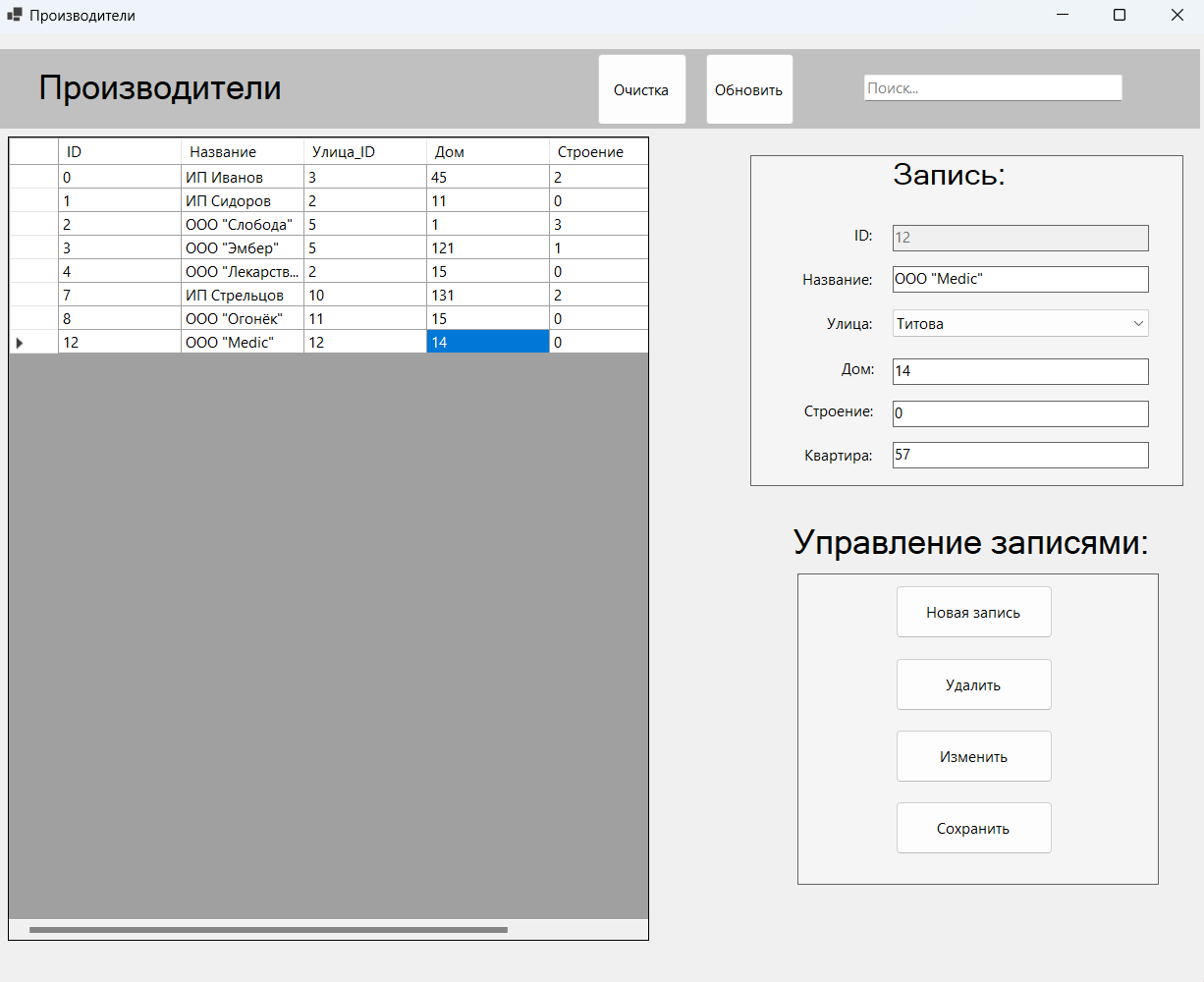


Рисунок 14 – интерфейс пункта “Производители”

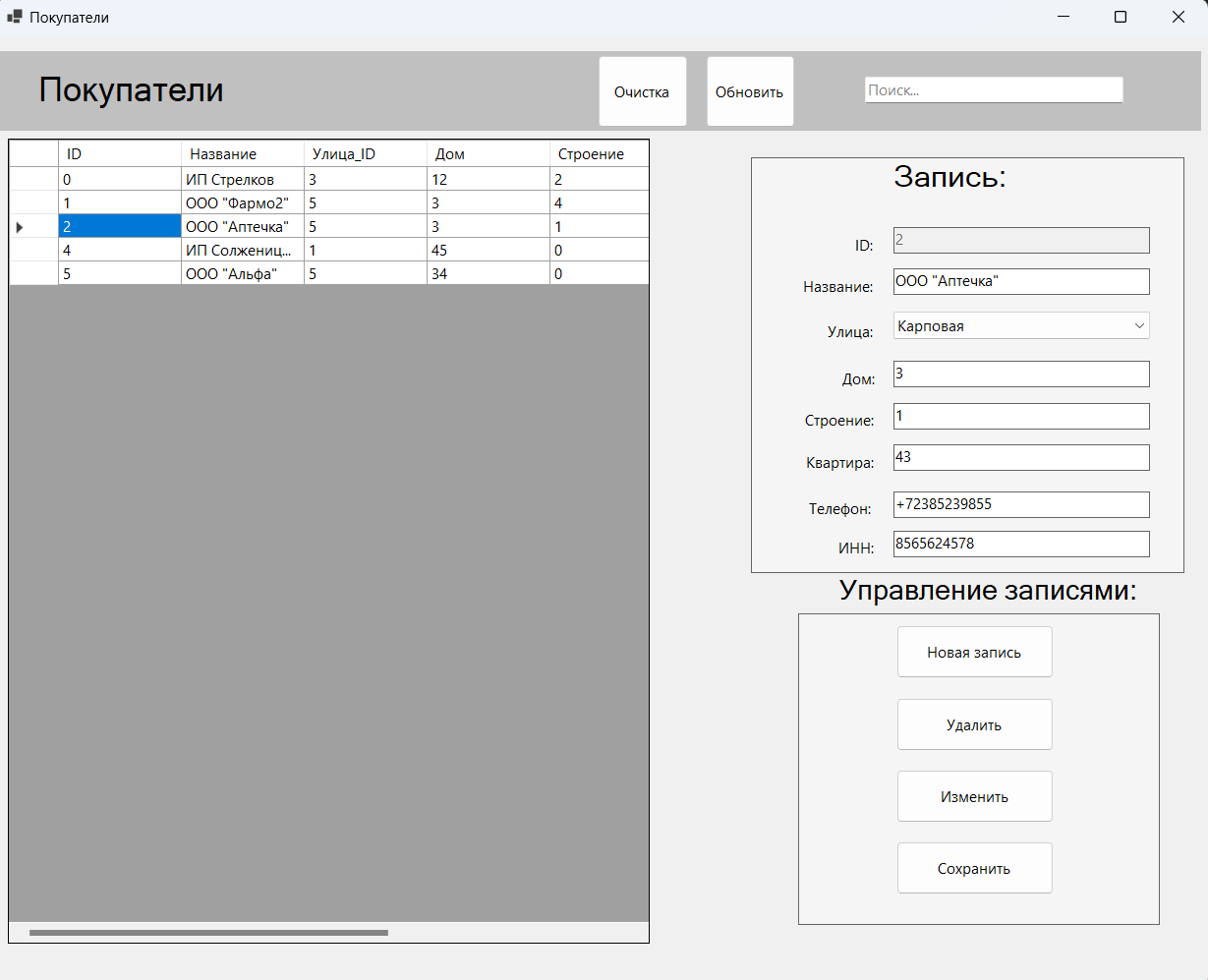


Рисунок 15 – интерфейс пункта “Покупатели”

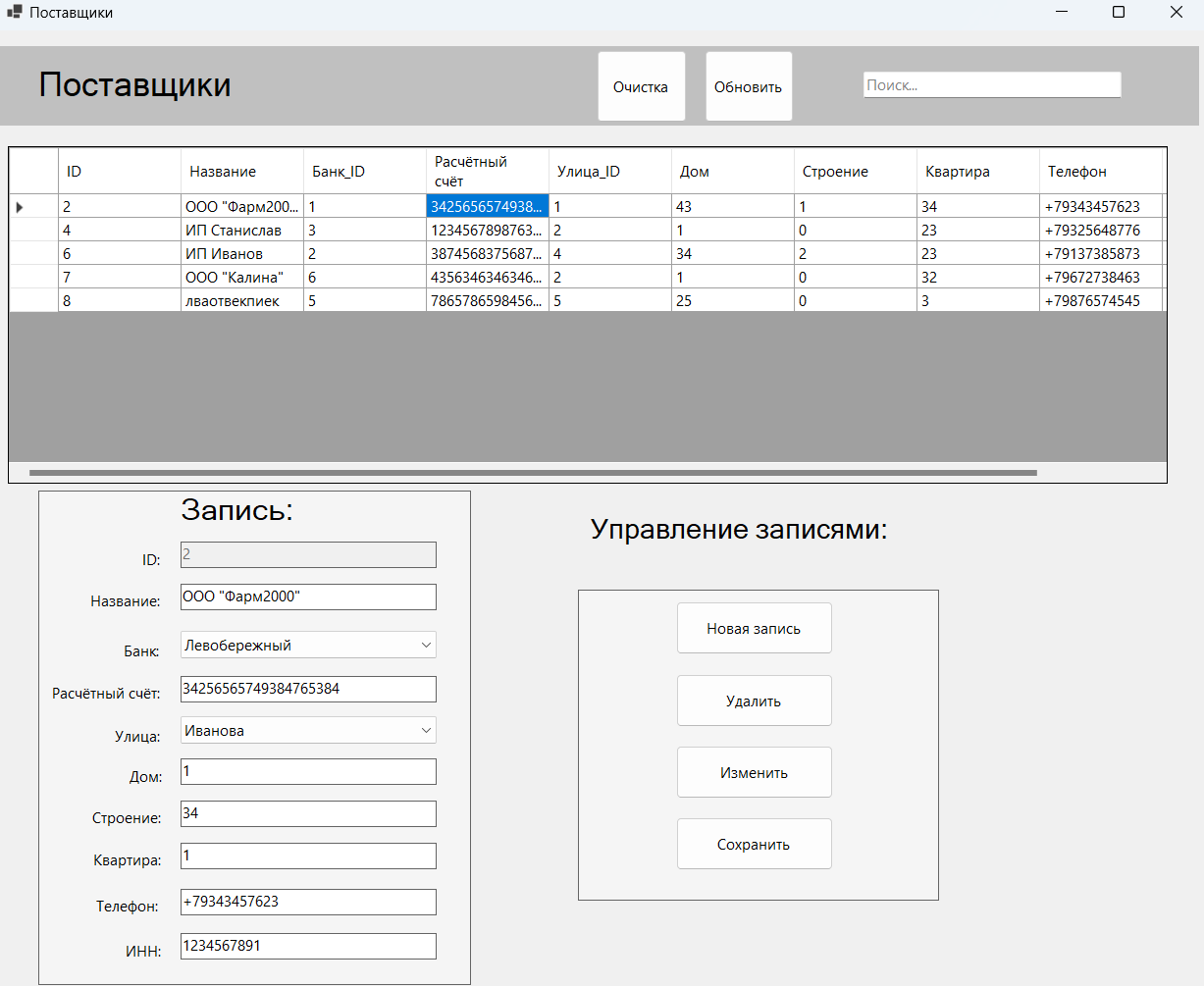


Рисунок 16 – интерфейс пункта “Поставщики”

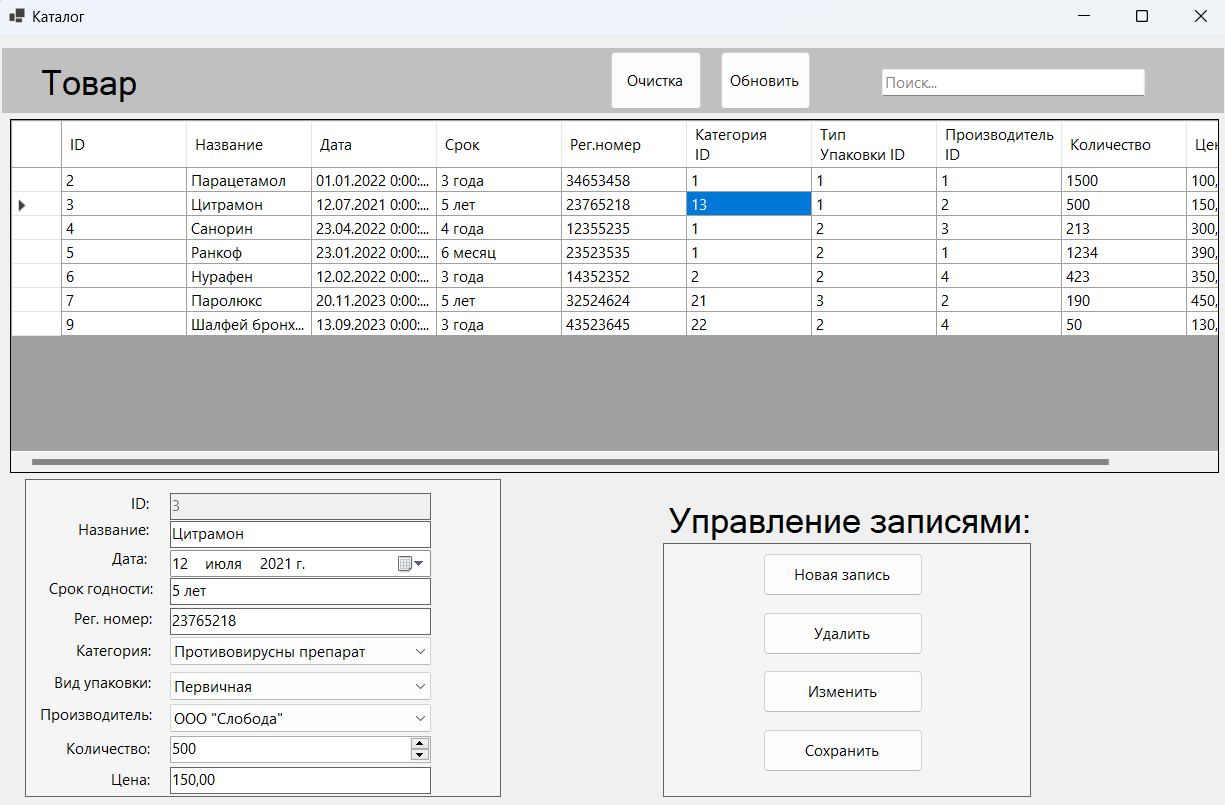


Рисунок 17 – интерфейс пункта “Каталог”

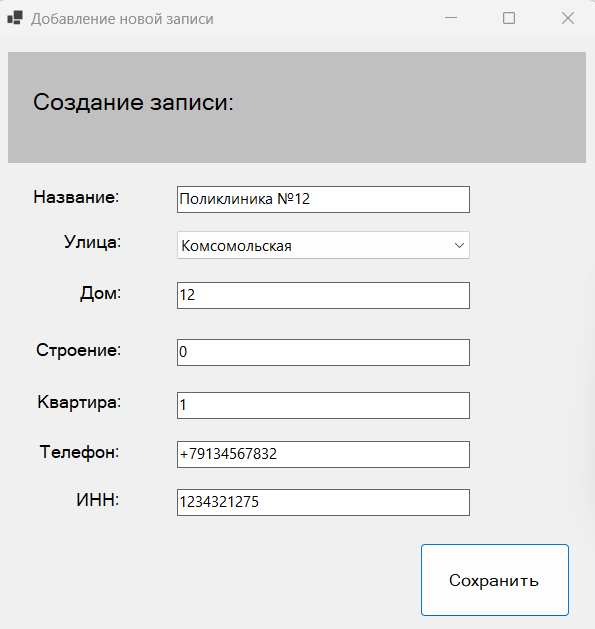


Рисунок 18 – интерфейс создания записи

Пункт “Справка” содержит подпункты “Содержание” (рис. 19) и “О программе” (рис. 20). В первом описывается назначение каждого пункта меню, второй – визитная карточка. Из функционала доступна только кнопка “Назад”, которая возвращает пользователя в основное меню.

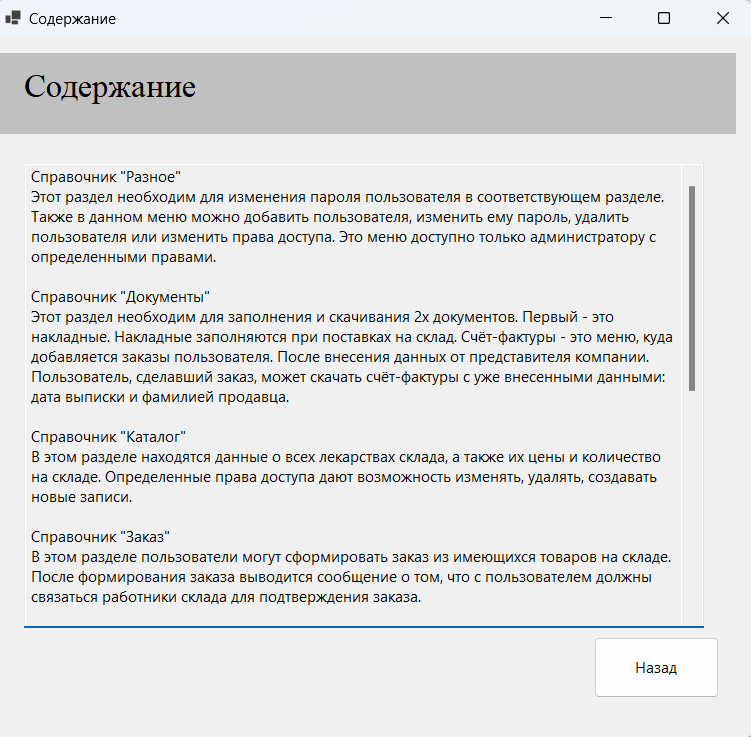


Рисунок 19 – интерфейс пункта “Содержание”

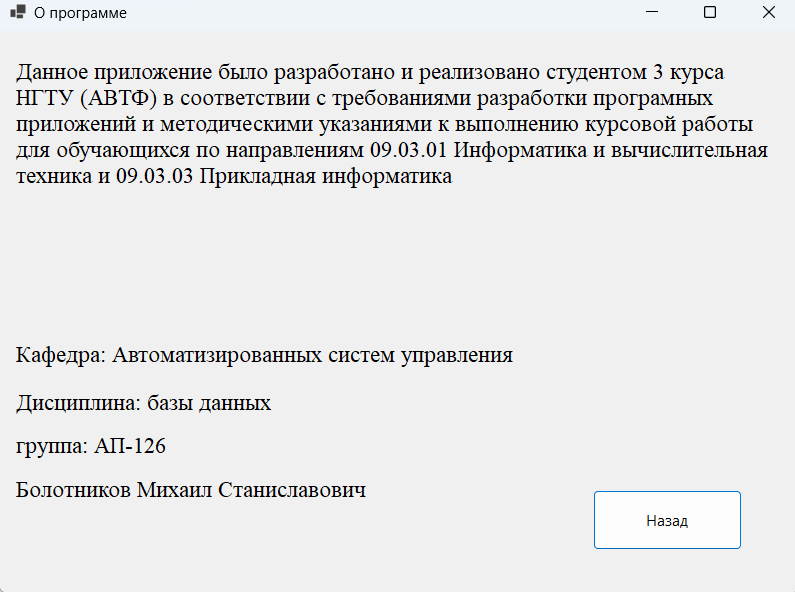


Рисунок 20 – интерфейс пункта “О программе”

Пункт меню “Разное” содержит подпункты “Настройка” (рис. 21) и “Сменить пароль” (рис. 22). Функционал подпункта “Настройка” позволяет выбрать пользователя из выпадающего списка и изменить ему права доступа. После изменения необходимо нажать на кнопку “Сохранить”. “Отчистка” позволяет снять все галочки на панели с таблицей и снять активного пользователя с выпадающего списка. “Управление записью” позволяет создать нового пользователя, изменить пароль выбранному пользователю, удалить выбранного пользователя.

“Сменить пароль” имеет интерфейс похожий на окно авторизации. Функционал позволяет внести новый пароль для активного пользователя.

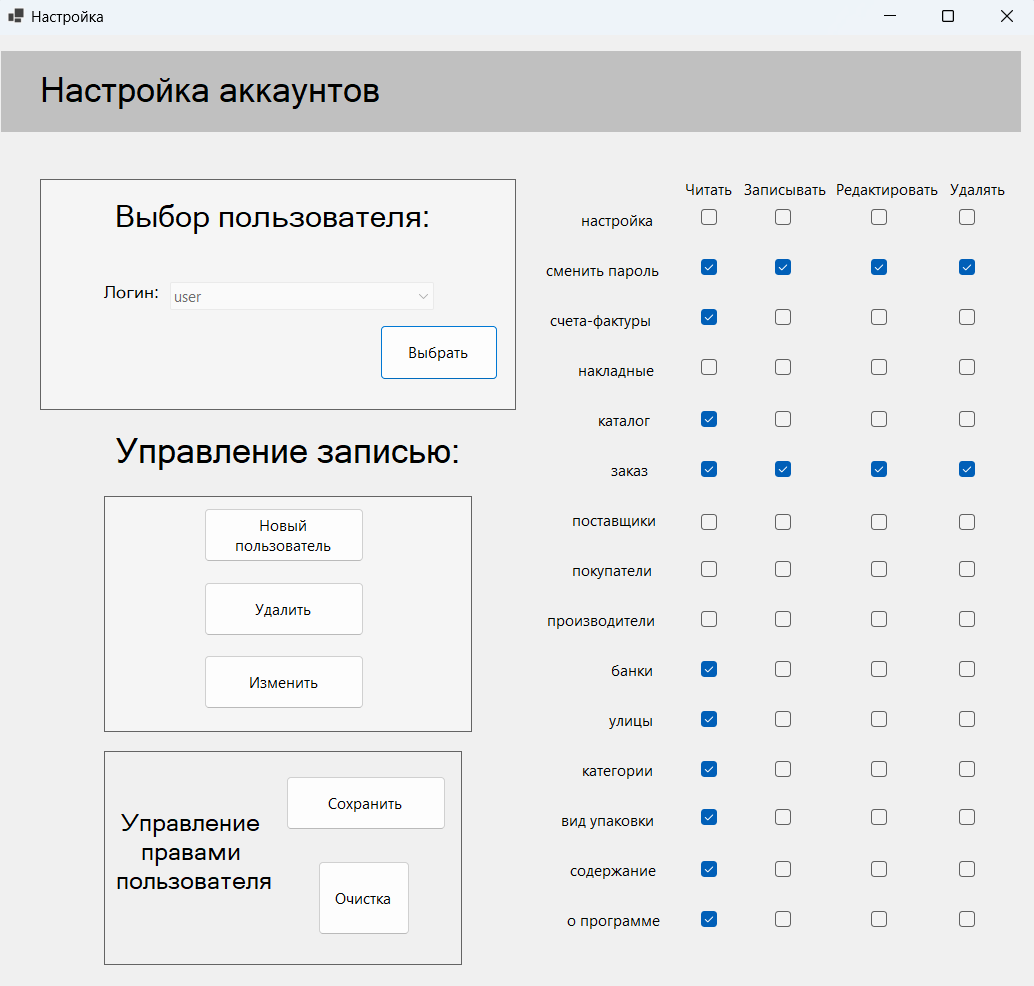


Рисунок 21 – интерфейс пункта “Настройка”

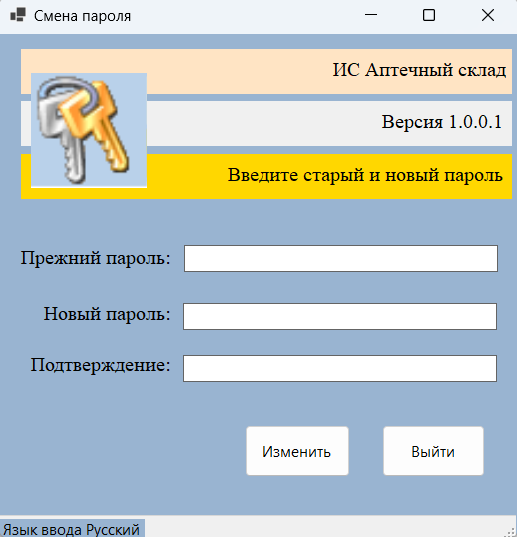


Рисунок 22 – интерфейс пункта “Сменить пароль”

При нажатии на пункт меню “Заказ” появляется окно “Заказ” (рис. 23)

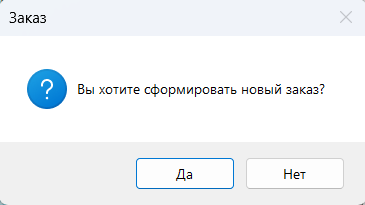


Рисунок 23 – окно “Заказ”

При нажатии на клавишу “Да” – открывается окно “Оформление заказа” (рис. 24). “Нет” – возврат в основное меню.

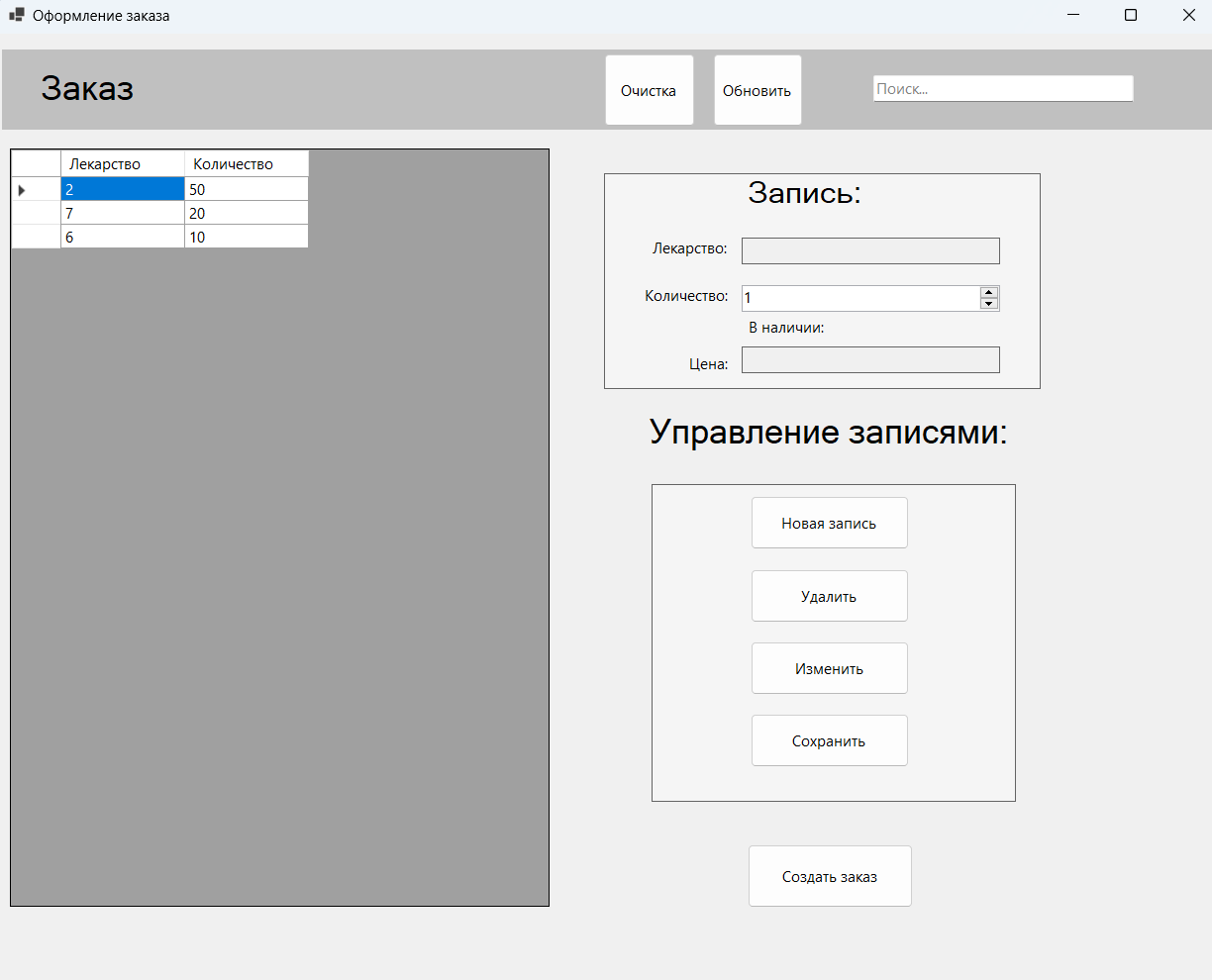


Рисунок 24 – интерфейс оформления заказа

Интерфейс оформления заказа отличается тем, что при нажатии на добавленные позиции в таблице на панели записи появляется информация о цене товара и доступном количестве на складе (рис. 25). Также добавлена кнопка “Создать заказ”, при нажатии на которую появляется новое окно (рис. 26).

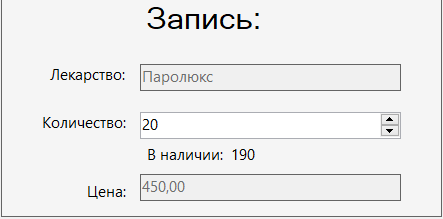
****

Рисунок 25 – интерфейс панели “Запись”

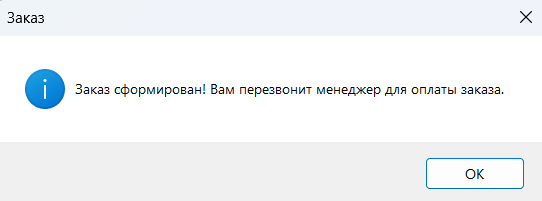


Рисунок 26 – Окно “Заказ” №2

Пункт меню “Документы” содержит подпункты “Счёт-фактуры” (рис. 26) и “Накладные”. Функционал подпункта “Счёт-фактуры” позволяет вносить данные в заказы. В таблице находятся заказы со статусами “Активен” и “Сформирован”. Первый говорит о том, что пользователь сделал заказ и его нужно обработать: внести фамилию продавца и дату выписки. После этого заказ получит статус “Сформирован”. Данный статус позволяет разблокировать кнопку “Загрузить счёт-фактуры”, которая активируется при нажатии на строку с заказом. “Экспорт в Excel” позволяет сохранить данные о заказе в Excel (рис. 27, 28). Пользователь с ограниченными правами доступа видит в таблице только заказы со статусом “Сформирован”.

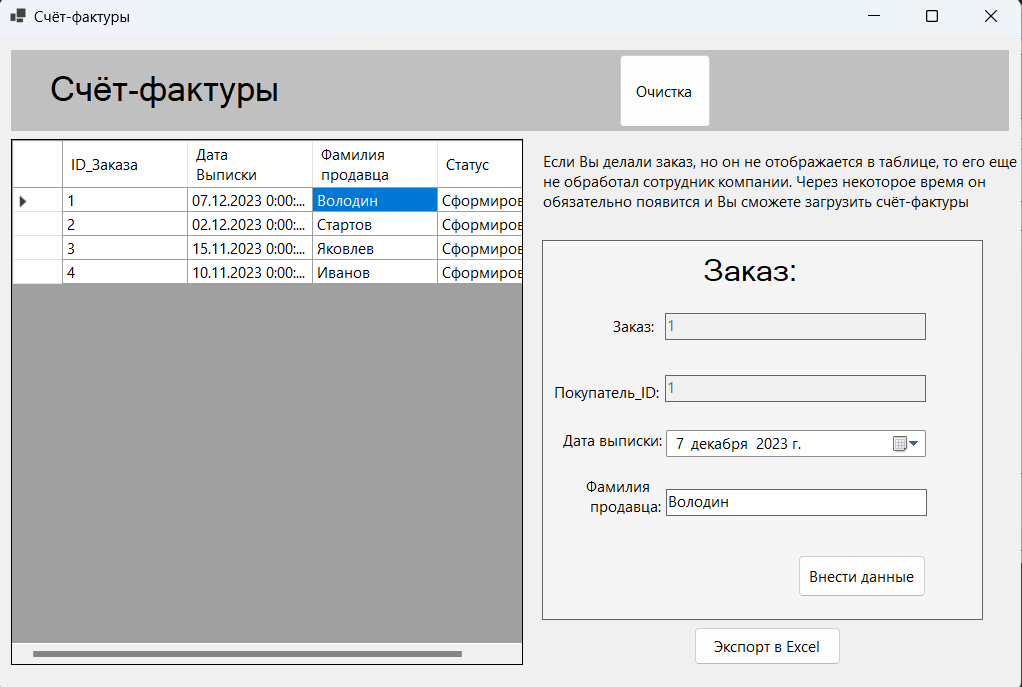
****

Рисунок 27 – интерфейс “Счёт-фактуры”

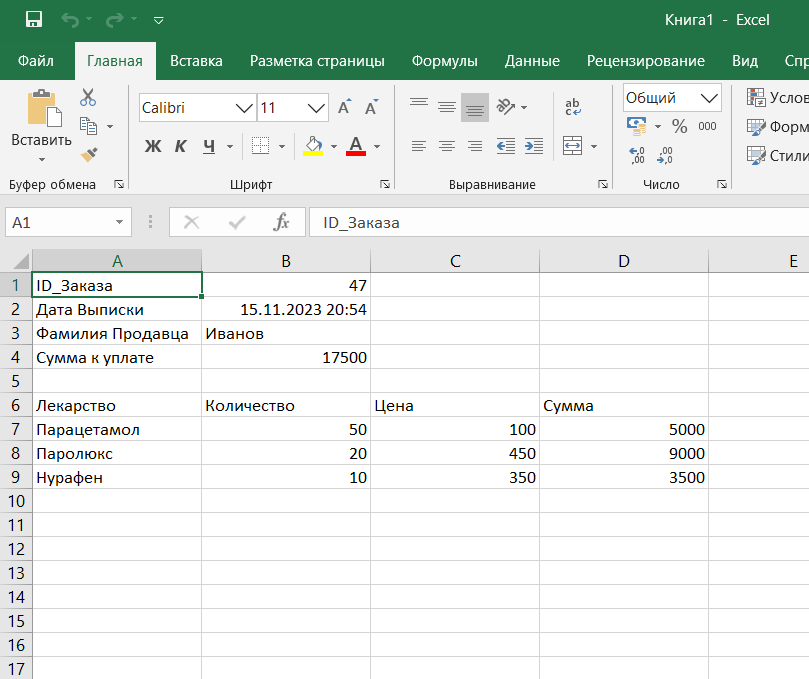
****

Рисунок 28 – экспорт счёт-фактуры

После нажатия на подпункт “Накладная” появляется окно создания накладной (рис. 29).

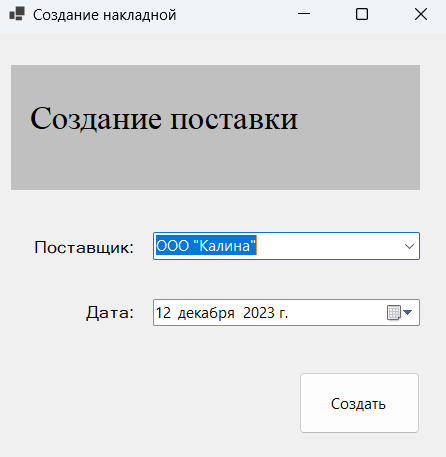
****

Рисунок 29 – Окно “Создание накладной”

После нажатия на клавишу “Создать” появляется окно формирования накладной (рис. 30).

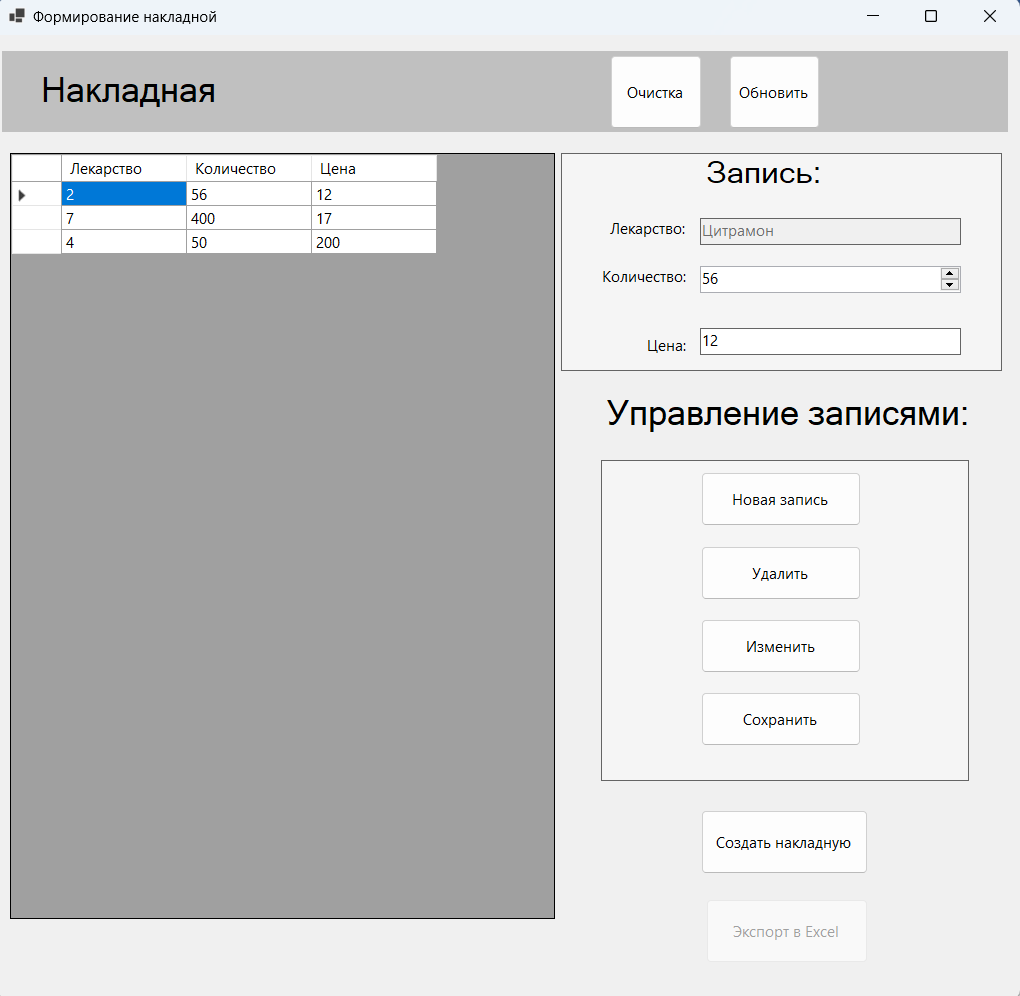


Рисунок 30 – интерфейс “Формирование накладной”

После добавление лекарств необходимо нажать на клавишу “Создать накладную”, после чего станет доступна функция экспорта в Excel (рис. 31).

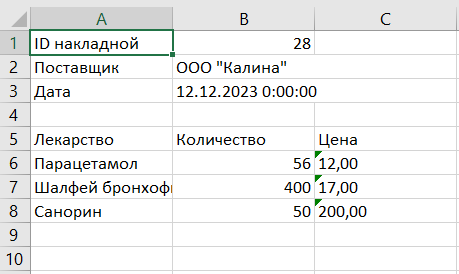


Рисунок 31 – экспорт накладной

# Заключение

В рамках данной курсовой работы было разработано приложение "Аптечный склад", предназначенное для автоматизации учета и управления аптечным складом. Основной целью работы было создание функционального и удобного инструмента, позволяющего эффективно управлять запасами медикаментов, контролировать их поступления и расходы, а также обеспечивать оперативный доступ к информации о наличии необходимых препаратов.

В результате работы была разработана база данных, которая содержит всю необходимую информацию о медикаментах, их характеристиках, поставщиках, ценах и сроках годности. Также были реализованы основные функции приложения, включающие в себя добавление новых товаров, учет поступлений и расходов, формирование отчетов и поиск информации.

В процессе разработки приложения было уделено особое внимание обеспечению безопасности данных пользователя. Были применены современные подходы и методы, включая шифрование и аутентификацию пользователей.

Разработка данного приложения позволяет повысить эффективность работы аптечного склада, уменьшить вероятность возникновения ошибок и потерю товаров, а также экономить время и ресурсы при решении организационных задач.

Исходя из вышесказанного, получаем, что поставленная задача успешно выполнена.

# Список используемой литературы

1. Вагнер, Билл C# Эффективное программирование / Билл Вагнер. – М.: ЛОРИ, 2013 год – 320 с.
2. <https://learn.microsoft.com/ru-ru/dotnet/csharp/fundamentals/coding-style/coding-conventions> (дата обращения 04.12.2023)
3. Цвалина, Кржиштоф Инфраструктура программных проектов. Соглашения, идиомы и шаблоны для многократно используемых библиотек .NET (2-е издание): Пер. с англ. / Кржиштоф Цвалина, Брэд Абрамс. - М.: ООО "И.Д. Вильямс", 2011. — 416 с.
4. Анисимов, А.Е.Требования и рекомендации по оформлению программного кода на языках С и С++ / А.Е. Анисимов. – Ижевск: Издательский центр «Удмуртский университет», 2020. – 48 с.
5. Гультяев А. К.Виртуальные машины: несколько компьютеров в одном / А. К. Гультяев - СПб.: Питер, 2006.- 224 с.

# Приложение

|  |
| --- |
| Содержимое модуля “Class1.cs” |
| using System.Data.OleDb;  namespace database  {  // Классс для работы с базой данных.  public class DataB  {  OleDbConnection myConnection = new OleDbConnection(@"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=Apteka1.mdb");  // Открытие соединения.  public void openConnection()  {  myConnection.Open();  }  // Закрытие соединения.  public void closeConnection()  {  if (myConnection != null)  {  myConnection.Close();  }  }  // Получение соединения.  public OleDbConnection getConnection()  {  return myConnection;  }  }  } |

Таблица 1 - содержимое решения DataBase

|  |
| --- |
| Содержимое модуля “Class1.cs” |
| using Banks;  using System;  namespace MyProj  {  public class Class1  {  static EventHandler BanksF()  {  return BanksF\_Click;  }  private static void BanksF\_Click(object sender, EventArgs e)  {  BanksForm form = new BanksForm();  form.Show();  }  }  } |
| Содержимое модуля “BanksForm.cs” |
| using System;  using System.Data;  using System.Drawing;  using System.Windows.Forms;  using database;  using System.Data.OleDb;  namespace Banks  {  public partial class BanksForm : Form  {  // Перечисление для состояния строки.  enum RowState  {  Existed,  New,  Modifide,  ModifideNew,  Deleted  }  DataB b = new DataB();  // Переменная для работы с DataGridView.  int selectedRow;  bool W, E, D;  public BanksForm()  {  InitializeComponent();  StartPosition = FormStartPosition.CenterScreen;  RightsFunction();  if (W == false)  {  buttonNew.Enabled = false;  }  if (E == false)  {  buttonChange.Enabled = false;  }  if (D == false)  {  buttonDel.Enabled = false;  }  }  // Авторизация прав пользователя на меню Банк.  private void RightsFunction()  {  b.openConnection();  string qwertystring = $"select \* from Rights where ID\_Menu = 16";  OleDbCommand command = new OleDbCommand(qwertystring, b.getConnection());  OleDbDataReader reader = command.ExecuteReader();  while (reader.Read())  {  W = Convert.ToBoolean(reader.GetBoolean(2));  E = Convert.ToBoolean(reader.GetBoolean(3));  D = Convert.ToBoolean(reader.GetBoolean(4));  }  reader.Close();  b.closeConnection();  }  // Создание столбцов в dataGridView.  private void CreateColums()  {  dataGridView1.Columns.Add("ID", "ID");  dataGridView1.Columns.Add("Name", "Наименование");  // Статус строки.  dataGridView1.Columns.Add("IsNew", String.Empty);  dataGridView1.Columns[2].Visible = false;  }  private void BanksForm\_Load(object sender, EventArgs e)  {  CreateColums();  RefreshDataGrid(dataGridView1);  }  // Отчистка таблицы "Запись".  private void ClearFields()  {  textBox1.Text = "";  textBox2.Text = "";  }  // Добавление Строки из базы данных в DataGridView.  private void ReadSingleRow(DataGridView dgw, IDataRecord record)  {  dgw.Rows.Add(record.GetInt32(0), record.GetString(1).Trim(), RowState.ModifideNew);  }  // Поиск данных в базе данных.  private void RefreshDataGrid(DataGridView dgw)  {  dgw.Rows.Clear();  string qwertystring = $"select \* from Банк";  OleDbCommand command = new OleDbCommand(qwertystring, b.getConnection());  b.openConnection();  OleDbDataReader reader = command.ExecuteReader();  while (reader.Read())  {  ReadSingleRow(dgw, reader);  }  reader.Close();  b.closeConnection();  }  // Запись данных в таблицу "Запись".  private void dataGridView1\_CellClick(object sender, DataGridViewCellEventArgs e)  {  selectedRow = e.RowIndex;  if (e.RowIndex >= 0)  {  DataGridViewRow row = dataGridView1.Rows[selectedRow];  textBox1.Text = row.Cells[0].Value.ToString().Trim();  textBox2.Text = row.Cells[1].Value.ToString().Trim();  }  }  // Нажатие на клавишу "Обновить".  private void button2\_Click(object sender, EventArgs e)  {  DialogResult result = MessageBox.Show("Сохранить изменения?", "Сохрание", MessageBoxButtons.YesNo, MessageBoxIcon.Question);  switch (result)  {  case DialogResult.Yes:  {  Update();  RefreshDataGrid(dataGridView1);  ClearFields();  break;  }  case DialogResult.No:  {  RefreshDataGrid(dataGridView1);  ClearFields();  break;  }  }  }  // Событие "Добавление записи".  private void buttonNew\_Click(object sender, EventArgs e)  {  AddBanks addfrm = new AddBanks();  addfrm.Show();  }  // Метод для поиска данных в БД.  private void Search(DataGridView dgw)  {  dgw.Rows.Clear();  string serchString = $"select \* from Банк where Наименование like '%" + textBox\_search.Text + "%'" + " or ID like '%" + textBox\_search.Text+ "%'";  OleDbCommand com = new OleDbCommand(serchString, b.getConnection());  b.openConnection();  OleDbDataReader read = com.ExecuteReader();  while (read.Read())  {  ReadSingleRow(dgw, read);  }  read.Close();  b.closeConnection();  }  // Событие когда поиск активен.  private void TextGotFocus(object sender, EventArgs e)  {  // Удаление подсказки.  if (textBox\_search.Text == "Поиск...")  {  textBox\_search.Text = "";  textBox\_search.ForeColor = Color.Black;  }  }  // Cобытие когда поиск неактивен.  private void TextLostFocus(object sender, EventArgs e)  {  // Возвращаем подсказку при неактивном состояни, если поле пустое.  if (string.IsNullOrWhiteSpace(textBox\_search.Text))  {  textBox\_search.Text = "Поиск...";  textBox\_search.ForeColor = Color.Gray;  RefreshDataGrid(dataGridView1);  }  }  // Событие при изменении поиска.  private void textBox\_search\_TextChanged(object sender, EventArgs e)  {  Search(dataGridView1);  }  // Метод обнавляет, удаляет или ничего не делает с данными в бд в соответствии со статусом каждой строки в DataGridView  private void Update()  {  b.openConnection();  for (int i = 0; i < dataGridView1.Rows.Count; i++)  {  var rowState = (RowState)dataGridView1.Rows[i].Cells[2].Value;  if (rowState == RowState.Existed) continue;  if (rowState == RowState.Deleted)  {  var id = Convert.ToInt32(dataGridView1.Rows[i].Cells[0].Value);  var deleteQwery = $"delete from Банк where ID = {id}";  var command = new OleDbCommand(deleteQwery, b.getConnection());  command.ExecuteNonQuery();  }  if (rowState == RowState.Modifide)  {  var id = dataGridView1.Rows[i].Cells[0].Value.ToString().Trim();  var name = dataGridView1.Rows[i].Cells[1].Value.ToString().Trim();  var changeQwery = $"update Банк set Наименование = '{name}' where ID ={id}";  var command = new OleDbCommand(changeQwery, b.getConnection());  command.ExecuteNonQuery();  }  }  b.closeConnection();  }  // Метод делает строку невидимой и присваивает ей статус "Удален".  private void deleteRow()  {  int index = dataGridView1.CurrentCell.RowIndex;  dataGridView1.Rows[index].Visible = false;  if (Convert.ToInt32(dataGridView1.Rows[index].Cells[0].Value) >= 0)  {  dataGridView1.Rows[index].Cells[2].Value = RowState.Deleted;  }  }  // Событие "Удаление записи".  private void buttonDel\_Click(object sender, EventArgs e)  {  deleteRow();  ClearFields();  }  // Событие "Сохраниение".  private void buttonSave\_Click(object sender, EventArgs e)  {  Update();  }  private void Change()  {  var selectedRowindex = dataGridView1.CurrentCell.RowIndex;  var id = textBox1.Text;  var name = textBox2.Text;  // Проверка на не пустоту строк и запрос на добавление новой строки в бд.  if (name != "")  {  dataGridView1.Rows[selectedRowindex].SetValues(id, name);  // установка статуса: Измененная строка.  dataGridView1.Rows[selectedRowindex].Cells[2].Value = RowState.Modifide;  }  else  {  MessageBox.Show("Неправильный ввод наименования", "Создание записи", MessageBoxButtons.OK, MessageBoxIcon.Warning);  }  }  private void buttonChange\_Click(object sender, EventArgs e)  {  Change();  ClearFields();  }  // Нажатие на клавишу "Отчистка".  private void button1\_Click(object sender, EventArgs e)  {  ClearFields();  }  }  } |
| Содержимое модуля “AddBanks.cs” |
| namespace Banks  {  public partial class AddBanks : Form  {  DataB database = new DataB();  public AddBanks()  {  InitializeComponent();  StartPosition = FormStartPosition.CenterScreen;  }  private void button1\_Click(object sender, EventArgs e)  {  database.openConnection();  var name = textBox1.Text;  // Проверка на не пустоту строки и запрос на добавление новой строки в бд.  if (name != "")  {  var addQwery = $"insert into Банк (Наименование) values ('{name}')";  var command4 = new OleDbCommand(addQwery, database.getConnection());  command4.ExecuteNonQuery();  MessageBox.Show("Запись успешно создана!", "Создание записи", MessageBoxButtons.OK, MessageBoxIcon.Information);  textBox1.Text = "";  }  else  {  MessageBox.Show("Неправильный ввод наименования ", "Создание записи", MessageBoxButtons.OK, MessageBoxIcon.Warning);  }  database.closeConnection();  }  }  } |

Таблица 2 - содержимое решения Banks

|  |
| --- |
| Содержимое модуля “Class1.cs” |
| using Buyers;  using System;  namespace MyProj  {  public class Class1  {  static EventHandler BuyersF()  {  return BuyersF\_Click;  }  private static void BuyersF\_Click(object sender, EventArgs e)  {  BuyersForm form = new BuyersForm();  form.Show();  }  }  } |
| Содержимое модуля “BuyersForm.cs” |
| namespace Buyers  {  public partial class BuyersForm : Form  {  // Перечисление для состояния строки.  enum RowState  {  Existed,  New,  Modifide,  ModifideNew,  Deleted  }  DataB b = new DataB();  // Переменная для работы с DataGridView.  int selectedRow;  bool W, E, D;  public BuyersForm()  {  InitializeComponent();  StartPosition = FormStartPosition.CenterScreen;  RightsFunction();  if (W == false)  {  buttonNew.Enabled = false;  }  if (E == false)  {  buttonChange.Enabled = false;  }  if (D == false)  {  buttonDel.Enabled = false;  }  }  // Авторизация прав пользователя на меню Покупатель.  private void RightsFunction()  {  string qwertystring = $"select \* from Rights where ID\_Menu = 10";  OleDbCommand command = new OleDbCommand(qwertystring, b.getConnection());  b.openConnection();  OleDbDataReader reader = command.ExecuteReader();  while (reader.Read())  {  W = Convert.ToBoolean(reader.GetBoolean(2));  E = Convert.ToBoolean(reader.GetBoolean(3));  D = Convert.ToBoolean(reader.GetBoolean(4));  }  reader.Close();  b.closeConnection();  }  // Создание столбцов в dataGridView.  private void CreateColums()  {  dataGridView1.Columns.Add("ID", "ID");  dataGridView1.Columns.Add("Name", "Название");  dataGridView1.Columns.Add("streets\_id", "Улица\_ID");  dataGridView1.Columns.Add("House", "Дом");  dataGridView1.Columns.Add("stroen", "Строение");  dataGridView1.Columns.Add("kvar", "Квартира");  dataGridView1.Columns.Add("tel", "Телефон");  dataGridView1.Columns.Add("inn", "ИНН");  // Статус строки.  dataGridView1.Columns.Add("IsNew", String.Empty);  dataGridView1.Columns[8].Visible = false;  }  // Отчистка таблицы "Запись".  private void ClearFields()  {  textBox1.Text = "";  textBox2.Text = "";  textBox3.Text = "";  textBox4.Text = "";  textBox5.Text = "";  textBox6.Text = "";  textBox7.Text = "";  comboBox1.SelectedIndex = -1;  comboBox1.Items.Clear();  }  // Добавление Строки из базы данных в DataGridView.  private void ReadSingleRow(DataGridView dgw, IDataRecord record)  {  int srtoen;  try  {  srtoen = Convert.ToInt32(record.GetInt32(4));  }  catch (Exception ex) { srtoen = 0; }  dgw.Rows.Add(record.GetInt32(0), record.GetString(1).Trim(), record.GetInt32(2), record.GetInt32(3), srtoen, record.GetInt32(5), record.GetString(6).Trim(), record.GetString(7).Trim(), RowState.ModifideNew);  }  // Поиск данных в базе данных.  private void RefreshDataGrid(DataGridView dgw)  {  dgw.Rows.Clear();  string qwertystring = $"select \* from Покупатель";  OleDbCommand command = new OleDbCommand(qwertystring, b.getConnection());  b.openConnection();  OleDbDataReader reader = command.ExecuteReader();  while (reader.Read())  {  ReadSingleRow(dgw, reader);  }  reader.Close();  b.closeConnection();  }  private void BuyersForm\_Load(object sender, EventArgs e)  {  CreateColums();  RefreshDataGrid(dataGridView1);  }  // Запись данных в таблицу "Запись".  private void dataGridView1\_CellClick(object sender, DataGridViewCellEventArgs e)  {  comboBox1.Items.Clear();  selectedRow = e.RowIndex;  if (e.RowIndex >= 0)  {  DataGridViewRow row = dataGridView1.Rows[selectedRow];  textBox1.Text = row.Cells[0].Value.ToString().Trim();  textBox2.Text = row.Cells[1].Value.ToString().Trim();  textBox3.Text = row.Cells[3].Value.ToString().Trim();  textBox5.Text = row.Cells[5].Value.ToString().Trim();  textBox6.Text = row.Cells[6].Value.ToString().Trim();  textBox7.Text = row.Cells[7].Value.ToString().Trim();  if (row.Cells[4].Value.ToString().Trim() == null)  {  textBox4.Text = "0";  }  else  {  textBox4.Text = row.Cells[4].Value.ToString().Trim();  }  // Поиск Улицы по id и добавление вкладок.  var qwery1 = $"select \* from Улица";  var command = new OleDbCommand(qwery1, b.getConnection());  b.openConnection();  OleDbDataReader reader = command.ExecuteReader();  int ID\_spr = 0;  while (reader.Read())  {  comboBox1.Items.Add(reader["Наименование"].ToString());  ID\_spr = reader.GetInt32(0);  if (ID\_spr == Convert.ToInt32(row.Cells[2].Value.ToString()))  {  comboBox1.Text = reader.GetString(1);  }  }  reader.Close();  b.closeConnection();  }  }  // Нажатие на клавишу "Обновить".  private void button2\_Click(object sender, EventArgs e)  {  DialogResult result = MessageBox.Show("Сохранить изменения?", "Сохрание", MessageBoxButtons.YesNo, MessageBoxIcon.Question);  switch (result)  {  case DialogResult.Yes:  {  Update();  RefreshDataGrid(dataGridView1);  ClearFields();  break;  }  case DialogResult.No:  {  RefreshDataGrid(dataGridView1);  ClearFields();  break;  }  } }  // Событие "Добавление записи".  private void buttonNew\_Click(object sender, EventArgs e)  {  AddBuyers addfrm = new AddBuyers();  addfrm.Show();  }  // Метод для поиска данных в БД.  private void Search(DataGridView dgw)  {  dgw.Rows.Clear();  string serchString = $"select \* from Покупатель where ID like '%" + textBox\_search.Text + "%'" + " or Название like '%" + textBox\_search.Text + "%'" + " or Улица\_ID like '%" + textBox\_search.Text + "%'" + " or Дом like '%" + textBox\_search.Text + "%'" + " or Строение like '%" + textBox\_search.Text + "%'" + " or Квартира like '%" + textBox\_search.Text + "%'" + " or Телефон like '%" + textBox\_search.Text + "%'" + " or ИНН like '%" + textBox\_search.Text + "%'";  OleDbCommand com = new OleDbCommand(serchString, b.getConnection());  b.openConnection();  OleDbDataReader read = com.ExecuteReader();  while (read.Read())  {  ReadSingleRow(dgw, read);  }  read.Close();  b.closeConnection();  }  // Событие когда поиск активен.  private void TextGotFocus(object sender, EventArgs e)  {  // Удаление подсказки.  if (textBox\_search.Text == "Поиск...")  {  textBox\_search.Text = "";  textBox\_search.ForeColor = Color.Black;  }  }  // Cобытие когда поиск неактивен.  private void TextLostFocus(object sender, EventArgs e)  {  // Возвращаем подсказку при неактивном состояни, если поле пустое.  if (string.IsNullOrWhiteSpace(textBox\_search.Text))  {  textBox\_search.Text = "Поиск...";  textBox\_search.ForeColor = Color.Gray;  RefreshDataGrid(dataGridView1);  }  }  // Событие при изменении поиска.  private void textBox\_search\_TextChanged(object sender, EventArgs e)  {  Search(dataGridView1);  }  // Метод обнавляет, удаляет или ничего не делает с данными в бд в соответствии со статусом каждой строки в DataGridView.  private void Update()  {  b.openConnection();  for (int i = 0; i < dataGridView1.Rows.Count; i++)  {  var rowState = (RowState)dataGridView1.Rows[i].Cells[8].Value;  if (rowState == RowState.Existed) continue;  if (rowState == RowState.Deleted)  {  var id = Convert.ToInt32(dataGridView1.Rows[i].Cells[0].Value);  var deleteQwery = $"delete from Покупатель where ID = {id}";  var command = new OleDbCommand(deleteQwery, b.getConnection());  command.ExecuteNonQuery();  }  if (rowState == RowState.Modifide)  {  var id = dataGridView1.Rows[i].Cells[0].Value.ToString().Trim();  var name = dataGridView1.Rows[i].Cells[1].Value.ToString().Trim();  var streets\_id = dataGridView1.Rows[i].Cells[2].Value.ToString().Trim();  var house = dataGridView1.Rows[i].Cells[3].Value.ToString().Trim();  var stroen = Convert.ToInt32(dataGridView1.Rows[i].Cells[4].Value.ToString().Trim());  var kvar = dataGridView1.Rows[i].Cells[5].Value.ToString().Trim();  var tel = dataGridView1.Rows[i].Cells[6].Value.ToString().Trim();  var inn = dataGridView1.Rows[i].Cells[7].Value.ToString().Trim();  string changeQwery;  if (stroen != 0)  {  changeQwery = $"update Покупатель set Название = '{name}', Улица\_ID ={streets\_id}, Дом = {house}, Строение= {stroen}, Квартира= {kvar}, Телефон= '{tel}', ИНН= '{inn}' where ID ={id}";  }  else  {  changeQwery = $"update Покупатель set Название = '{name}', Улица\_ID ={streets\_id}, Дом = {house}, Строение= NULL, Квартира= {kvar}, Телефон= '{tel}', ИНН= '{inn}' where ID ={id}";  }  var command = new OleDbCommand(changeQwery, b.getConnection());  command.ExecuteNonQuery();  }  }  b.closeConnection();  }  // Метод делает строку невидимой и присваивает ей статус "Удален".  private void deleteRow()  {  int index = dataGridView1.CurrentCell.RowIndex;  dataGridView1.Rows[index].Visible = false;  if (Convert.ToInt32(dataGridView1.Rows[index].Cells[0].Value) >= 0)  {  dataGridView1.Rows[index].Cells[8].Value = RowState.Deleted;  }  }  // Событие "Удаление записи".  private void buttonDel\_Click(object sender, EventArgs e)  {  deleteRow();  ClearFields();  }  // Событие "Сохраниение".  private void buttonSave\_Click(object sender, EventArgs e)  {  Update();  }  private void Change()  {  var selectedRowindex = dataGridView1.CurrentCell.RowIndex;  var id = textBox1.Text;  var name = textBox2.Text;  var streets\_id = 0;  int house;  int kvar;  var tel = textBox6.Text;  var inn = textBox7.Text;  bool isNumber1 = int.TryParse(textBox3.Text, out house);  bool isNumber3 = int.TryParse(textBox5.Text, out kvar);  // Обратный поиск ID по Наименованием и названиям виду упаковки, категории, производителю.  b.openConnection();  // Поиск Улица\_ID.  var qwery1 = $"select ID from Улица where Наименование = '{comboBox1.Text}'";  var command = new OleDbCommand(qwery1, b.getConnection());  OleDbDataReader reader = command.ExecuteReader();  while (reader.Read())  {  streets\_id = reader.GetInt32(0);  }  reader.Close();  // Проверка ввода.  if (isNumber1 == true && isNumber3 == true && tel.Length <=12 && inn.Length <=20 && name != "" && Convert.ToInt32(house) > 0 && Convert.ToInt32(kvar) > 0)  {  int strorn\_int;  bool isNumber2 = int.TryParse(textBox4.Text, out strorn\_int);  if (isNumber2 == true)  {  dataGridView1.Rows[selectedRowindex].SetValues(id, name, streets\_id, house, strorn\_int, kvar, tel, inn);  }  else  {  dataGridView1.Rows[selectedRowindex].SetValues(id, name, streets\_id, house, 0, kvar, tel, inn);  }  // установка статуса: Измененная строка.  dataGridView1.Rows[selectedRowindex].Cells[8].Value = RowState.Modifide;  }  else  {  MessageBox.Show("Неправильный ввод", "Создание записи", MessageBoxButtons.OK, MessageBoxIcon.Warning);  }  b.closeConnection();  }  private void buttonChange\_Click(object sender, EventArgs e)  {  Change();  ClearFields();  }  // Нажатие на клавишу "Отчистка".  private void button1\_Click(object sender, EventArgs e)  {  ClearFields();  }  }  } |
| Содержимое модуля “AddBuyers.cs” |
| using database;  using System;  using System.Data.OleDb;  using System.Drawing;  using System.Windows.Forms;  namespace Buyers  {  public partial class AddBuyers : Form  {  DataB database = new DataB();  public AddBuyers()  {  InitializeComponent();  StartPosition = FormStartPosition.CenterScreen;  database.openConnection();  // Поиск Улицы из бд.  var qwery1 = $"select \* from Улица";  var command = new OleDbCommand(qwery1, database.getConnection());  OleDbDataReader reader = command.ExecuteReader();  while (reader.Read())  {  comboBox1.Items.Add(reader["Наименование"].ToString());  }  reader.Close();  database.closeConnection();  }  private void button1\_Click(object sender, EventArgs e)  {  database.openConnection();  var name = textBox1.Text;  int streets\_id = 0;  int house;  int stroen;  int kvar;  var tel = textBox5.Text;  var inn = textBox6.Text;  // Поиск Улица\_ID.  var qwery1 = $"select ID from Улица where Наименование = '{comboBox1.Text}'";  var command = new OleDbCommand(qwery1, database.getConnection());  OleDbDataReader reader = command.ExecuteReader();  while (reader.Read())  {  streets\_id = reader.GetInt32(0);  }  reader.Close();  // Проверка ввода.  bool isNumber1 = int.TryParse(textBox2.Text, out house);  bool isNumber2 = int.TryParse(textBox3.Text, out stroen);  bool isNumber3 = int.TryParse(textBox4.Text, out kvar);  string addQwery;  // Проверка на не пустоту строк и запрос на добавление новой строки в бд.  if (isNumber1 == true && isNumber3 == true && tel.Length <=12 && inn.Length <=20 &&house > 0 && kvar > 0 && name != "" && streets\_id > 0)  {  if (isNumber2 == false)  {  addQwery = $"insert into Покупатель (Название, Улица\_ID, Дом, Строение, Квартира, Телефон, ИНН) values ('{name}', {streets\_id}, {house}, NULL, {kvar}, '{tel}', '{inn}')";  }  else  {  addQwery = $"insert into Покупатель (Название, Улица\_ID, Дом, Строение, Квартира,Телефон, ИНН) values ('{name}', {streets\_id}, {house}, {stroen}, {kvar},'{tel}', '{inn}' )";  }  var command4 = new OleDbCommand(addQwery, database.getConnection());  command4.ExecuteNonQuery();  MessageBox.Show("Запись успешно создана!", "Создание записи", MessageBoxButtons.OK, MessageBoxIcon.Information);  textBox1.Text = "";  textBox2.Text = "";  textBox3.Text = "";  textBox4.Text = "";  textBox5.Text = "";  textBox6.Text = "";  }  else  {  MessageBox.Show("Неправильный ввод", "Создание записи", MessageBoxButtons.OK, MessageBoxIcon.Warning);  }  database.closeConnection();  }  }  } |

Таблица 3 - содержимое решения Buyers

|  |
| --- |
| Содержимое модуля “Class1.cs” |
| using Catalog;  using System;  namespace MyProj  {  public class Class1  {  static EventHandler catalogF()  {  return catalogF\_Click;  }  private static void catalogF\_Click(object sender, EventArgs e)  {  CatalogForm form = new CatalogForm();  form.Show();  }  }  } |
| Содержимое модуля “CatalogForm.cs” |
| namespace Catalog  {  // Перечисление для состояния строки.  enum RowState  {  Existed,  New,  Modifide,  ModifideNew,  Deleted  }  public partial class CatalogForm : Form  {  DataB b = new DataB();  // Переменная для работы с DataGridView.  int selectedRow;  bool W, E, D;  public CatalogForm()  {  InitializeComponent();  StartPosition = FormStartPosition.CenterScreen;  RightsFunction();  if(W == false)  {  buttonNew.Enabled = false;  }  if (E == false)  {  buttonChange.Enabled = false;  textBox3.Enabled = false;  }  if (D == false)  {  buttonDel.Enabled = false;  }  }  // Авторизация прав пользователя на меню Каталог.  private void RightsFunction()  {  string qwertystring = $"select \* from Rights where ID\_Menu = 7";  OleDbCommand command = new OleDbCommand(qwertystring, b.getConnection());  b.openConnection();  OleDbDataReader reader = command.ExecuteReader();  while (reader.Read())  {  W = Convert.ToBoolean(reader.GetBoolean(2));  E = Convert.ToBoolean(reader.GetBoolean(3));  D = Convert.ToBoolean(reader.GetBoolean(4));  }  reader.Close();  b.closeConnection();  }  // Создание столбцов в dataGridView.  private void CreateColums()  {  dataGridView1.Columns.Add("ID", "ID");  dataGridView1.Columns.Add("Name", "Название");  dataGridView1.Columns.Add("data", "Дата");  dataGridView1.Columns.Add("Srok", "Срок");  dataGridView1.Columns.Add("Number", "Рег.номер");  dataGridView1.Columns.Add("KategId", "Категория ID");  dataGridView1.Columns.Add("Type", "Тип Упаковки ID");  dataGridView1.Columns.Add("Prod", "Производитель ID");  dataGridView1.Columns.Add("kol", "Количество");  dataGridView1.Columns.Add("price", "Цена");  // Статус строки.  dataGridView1.Columns.Add("IsNew", String.Empty);  dataGridView1.Columns[10].Visible = false;  }  // Отчистка таблицы "Запись".  private void ClearFields()  {  textBox1.Text = "";  textBox2.Text = "";  textBox4.Text = "";  textBox5.Text = "";  textBox3.Text = "";  comboBox1.SelectedIndex = -1;  comboBox2.SelectedIndex = -1;  comboBox3.SelectedIndex = -1;  comboBox1.Items.Clear();  comboBox2.Items.Clear();  comboBox3.Items.Clear();  numericUpDown1.Text = "";  }  // Добавление Строки из базы данных в DataGridView.  private void ReadSingleRow(DataGridView dgw, IDataRecord record)  {  dgw.Rows.Add(record.GetInt32(0), record.GetString(1).Trim(), record.GetDateTime(2), record.GetString(3).Trim(), record.GetString(4).Trim(), record.GetInt32(5), record.GetInt32(6), record.GetInt32(7), record.GetInt32(8),record.GetDecimal(9), RowState.ModifideNew);  }  // Поиск данных в базе данных.  private void RefreshDataGrid(DataGridView dgw)  {  dgw.Rows.Clear();  string qwertystring = $"select \* from Лекарство";  OleDbCommand command = new OleDbCommand(qwertystring, b.getConnection());  b.openConnection();  OleDbDataReader reader = command.ExecuteReader();  while (reader.Read())  {  ReadSingleRow(dgw, reader);  }  reader.Close();  b.closeConnection();  }  private void CatalogForm\_Load(object sender, EventArgs e)  {  CreateColums();  RefreshDataGrid(dataGridView1);  }  // Запись данных в таблицу "Запись".  private void dataGridView1\_CellClick(object sender, DataGridViewCellEventArgs e)  {  comboBox1.Items.Clear();  comboBox2.Items.Clear();  comboBox3.Items.Clear();  selectedRow = e.RowIndex;  if(e.RowIndex>=0)  {  DataGridViewRow row = dataGridView1.Rows[selectedRow];  textBox1.Text = row.Cells[0].Value.ToString().Trim();  textBox2.Text = row.Cells[1].Value.ToString().Trim();  dateTimePicker1.Text = row.Cells[2].Value.ToString();  textBox4.Text = row.Cells[3].Value.ToString().Trim();  textBox5.Text = row.Cells[4].Value.ToString().Trim();  textBox3.Text = row.Cells[9].Value.ToString().Trim();  numericUpDown1.Text = row.Cells[8].Value.ToString();  // Поиск Категории по id и добавление вкладок.  var qwery1 = $"select \* from Категория";  var command = new OleDbCommand(qwery1, b.getConnection());  b.openConnection();  OleDbDataReader reader = command.ExecuteReader();  int ID\_spr = 0;  while (reader.Read())  {  comboBox1.Items.Add(reader["Наименование"].ToString());  ID\_spr = reader.GetInt32(0);  if(ID\_spr == Convert.ToInt32(row.Cells[5].Value.ToString()))  {  comboBox1.Text = reader.GetString(1);  }  }  reader.Close();  // Поиск вида упаковки по id и добавление вкладок.  var qwery2 = $"select \* from ВидУпаковки";  var command2 = new OleDbCommand(qwery2, b.getConnection());  OleDbDataReader reader2 = command2.ExecuteReader();  while (reader2.Read())  {  comboBox2.Items.Add(reader2["Наименование"].ToString());  ID\_spr = reader2.GetInt32(0);  if (ID\_spr == Convert.ToInt32(row.Cells[6].Value.ToString()))  {  comboBox2.Text = reader2.GetString(1);  }  }  reader2.Close();  // Поиск Производителя по ID и добавление вкладок.  var qwery3 = $"select \* from Производитель";  var command3 = new OleDbCommand(qwery3, b.getConnection());  OleDbDataReader reader3 = command3.ExecuteReader();  while (reader3.Read())  {  comboBox3.Items.Add(reader3["Название"].ToString());  ID\_spr = reader3.GetInt32(0);  if (ID\_spr == Convert.ToInt32(row.Cells[7].Value.ToString()))  {  comboBox3.Text = reader3.GetString(1);  }  }  reader3.Close();  b.closeConnection();  }  }  // Нажатие на клавишу "Обновить".  private void button2\_Click(object sender, EventArgs e)  {  DialogResult result = MessageBox.Show("Сохранить изменения?", "Сохрание", MessageBoxButtons.YesNo, MessageBoxIcon.Question);  switch (result)  {  case DialogResult.Yes:  {  Update();  RefreshDataGrid(dataGridView1);  ClearFields();  break;  }  case DialogResult.No:  {  RefreshDataGrid(dataGridView1);  ClearFields();  break;  }  } }  // Событие "Добавление записи".  private void buttonNew\_Click(object sender, EventArgs e)  {  Add\_Form addfrm = new Add\_Form();  addfrm.Show();  }  // Метод для поиска данных в БД.  private void Search(DataGridView dgw)  {  dgw.Rows.Clear();  string serchString = $"select \* from Лекарство where ID like '%" + textBox\_search.Text + "%'" + " or Название like '%" + textBox\_search.Text + "%'" + " or Дата like '%" + textBox\_search.Text + "%'" + " or СрокГодности like '%" + textBox\_search.Text + "%'" + " or РегНомер like '%" + textBox\_search.Text + "%'" + " or Категория\_ID like '%" + textBox\_search.Text + "%'" + " or ВидУпаковки\_ID like '%" + textBox\_search.Text + "%'" + " or Производитель\_ID like '%" + textBox\_search.Text + "%'" + " or Количество like '%" + textBox\_search.Text + "%'" + " or Цена like '%" + textBox\_search.Text + "%'";  OleDbCommand com = new OleDbCommand(serchString, b.getConnection());  b.openConnection();  OleDbDataReader read = com.ExecuteReader();  while(read.Read())  {  ReadSingleRow(dgw, read);  }  read.Close();  b.closeConnection();  }  // Событие когда поиск активен.  private void TextGotFocus(object sender, EventArgs e)  {  // Удаление подсказки.  if (textBox\_search.Text == "Поиск...")  {  textBox\_search.Text = "";  textBox\_search.ForeColor = Color.Black;  }  }  // Cобытие когда поиск неактивен.  private void TextLostFocus(object sender, EventArgs e)  {  // Возвращаем подсказку при неактивном состояни, если поле пустое.  if (string.IsNullOrWhiteSpace(textBox\_search.Text))  {  textBox\_search.Text = "Поиск...";  textBox\_search.ForeColor = Color.Gray;  RefreshDataGrid(dataGridView1);  }  }  // Событие при изменении поиска.  private void textBox\_search\_TextChanged(object sender, EventArgs e)  {  Search(dataGridView1);  }  // Метод обнавляет, удаляет или ничего не делает с данными в бд в соответствии со статусом каждой строки в DataGridView.  private void Update()  {  b.openConnection();  for(int i = 0; i < dataGridView1.Rows.Count; i++)  {  var rowState = (RowState)dataGridView1.Rows[i].Cells[10].Value;  if (rowState == RowState.Existed) continue;  if(rowState == RowState.Deleted)  {  var id = Convert.ToInt32(dataGridView1.Rows[i].Cells[0].Value);  var deleteQwery = $"delete from Лекарство where ID = {id}";  var command = new OleDbCommand(deleteQwery, b.getConnection());  command.ExecuteNonQuery();  }  if(rowState == RowState.Modifide)  {  var id = dataGridView1.Rows[i].Cells[0].Value.ToString().Trim();  var name = dataGridView1.Rows[i].Cells[1].Value.ToString().Trim();  var dat = dataGridView1.Rows[i].Cells[2].Value.ToString().Trim();  var Srok = dataGridView1.Rows[i].Cells[3].Value.ToString().Trim();  var num = dataGridView1.Rows[i].Cells[4].Value.ToString().Trim();  var type = dataGridView1.Rows[i].Cells[5].Value.ToString().Trim();  var Vid = dataGridView1.Rows[i].Cells[6].Value.ToString().Trim();  var prod = dataGridView1.Rows[i].Cells[7].Value.ToString().Trim();  var count = dataGridView1.Rows[i].Cells[8].Value.ToString().Trim();  string price = dataGridView1.Rows[i].Cells[9].Value.ToString().Trim();  price = price.Replace(",", ".");  var changeQwery = $"update Лекарство set Название = '{name}', Дата ='{dat}', СрокГодности = '{Srok}', РегНомер= '{num}', Категория\_ID= {type}, ВидУпаковки\_ID= {Vid}, Производитель\_ID= {prod}, Количество= {count}, Цена = {price} where ID ={id}";  var command = new OleDbCommand(changeQwery, b.getConnection());  command.ExecuteNonQuery();  }  }  b.closeConnection();  }  // Метод делает строку невидимой и присваивает ей статус "Удален".  private void deleteRow()  {  int index = dataGridView1.CurrentCell.RowIndex;  dataGridView1.Rows[index].Visible = false;  if (Convert.ToInt32(dataGridView1.Rows[index].Cells[0].Value) >= 0)  {  dataGridView1.Rows[index].Cells[10].Value = RowState.Deleted;  }  }  // Событие "Удаление записи".  private void buttonDel\_Click(object sender, EventArgs e)  {  deleteRow();  ClearFields();  }  // Событие "Сохраниение".  private void buttonSave\_Click(object sender, EventArgs e)  {  Update();  }  private void Change()  {  var selectedRowindex = dataGridView1.CurrentCell.RowIndex;  var id = textBox1.Text;  var name = textBox2.Text;  var dat = dateTimePicker1.Value.Date;  var Srok = textBox4.Text;  var num = textBox5.Text;  var price = textBox3.Text;  int type\_id = 0;  int Vid\_id = 0;  int prod\_id = 0;  int count = Convert.ToInt32(numericUpDown1.Value);  //  // Обратный поиск ID по Наименованием и названиям виду упаковки, категории, производителю.  //  b.openConnection();  // Поиск Категории\_ID.  var qwery1 = $"select ID from Категория where Наименование = '{comboBox1.Text}'";  var command = new OleDbCommand(qwery1, b.getConnection());  OleDbDataReader reader = command.ExecuteReader();  while (reader.Read())  {  type\_id = reader.GetInt32(0);  }  reader.Close();  // Поиск ВидУпаковки\_ID.  var qwery2 = $"select ID from ВидУпаковки where Наименование = '{comboBox2.Text}'";  var command2 = new OleDbCommand(qwery2, b.getConnection());  OleDbDataReader reader2 = command2.ExecuteReader();  while (reader2.Read())  {  Vid\_id = reader2.GetInt32(0);  }  reader2.Close();  // Поиск Производитель\_ID.  var qwery3 = $"select ID from Производитель where Название = '{comboBox3.Text}'";  var command3 = new OleDbCommand(qwery3, b.getConnection());  OleDbDataReader reader3 = command3.ExecuteReader();  while (reader3.Read())  {  prod\_id = reader3.GetInt32(0);  }  reader3.Close();  // Проверка на не пустоту строк и запрос на добавление новой строки в бд.  if (name != "" && Srok != "" && num != "" && Srok.Length < 11 && num.Length < 46 && type\_id > 0 && Vid\_id > 0 && prod\_id > 0)  {  dataGridView1.Rows[selectedRowindex].SetValues(id, name, dat, Srok, num, type\_id, Vid\_id, prod\_id, count, price);  // установка статуса: Измененная строка.  dataGridView1.Rows[selectedRowindex].Cells[10].Value = RowState.Modifide;  }  else  {  MessageBox.Show("Неправильный ввод", "Создание записи", MessageBoxButtons.OK, MessageBoxIcon.Warning);  }  b.closeConnection();  }  private void buttonChange\_Click(object sender, EventArgs e)  {  Change();  ClearFields();  }  // Нажатие на клавишу "Отчистка"  private void button1\_Click(object sender, EventArgs e)  {  ClearFields();  }  }  } |
| Содержимое модуля “Add\_Form.cs” |
| namespace Catalog  {  public partial class Add\_Form : Form  {  DataB database = new DataB();  public Add\_Form()  {  InitializeComponent();  StartPosition = FormStartPosition.CenterScreen;  dateTimePicker1.Text = "";  database.openConnection();  // Поиск Категорий из бд.  var qwery1 = $"select \* from Категория";  var command = new OleDbCommand(qwery1, database.getConnection());  OleDbDataReader reader = command.ExecuteReader();  while (reader.Read())  {  comboBoxCat.Items.Add(reader["Наименование"].ToString());  }  reader.Close();  // Поиск Вида упаковки из бд.  var qwery2 = $"select \* from ВидУпаковки";  var command2 = new OleDbCommand(qwery2, database.getConnection());  OleDbDataReader reader2 = command2.ExecuteReader();  while (reader2.Read())  {  comboBoxVid.Items.Add(reader2["Наименование"].ToString());  }  reader2.Close();  // Поиск Производителя из бд.  var qwery3 = $"select \* from Производитель";  var command3 = new OleDbCommand(qwery3, database.getConnection());  OleDbDataReader reader3 = command3.ExecuteReader();  while (reader3.Read())  {  comboBoxPro.Items.Add(reader3["Название"].ToString());  }  reader3.Close();  database.closeConnection();  }  private void button1\_Click(object sender, EventArgs e)  {  database.openConnection();  var name = textBox1.Text;  var dat = dateTimePicker1.Value.Date;  string Srok = textBox3.Text;  string num = textBox4.Text;  int type\_id = 0;  int Vid\_id = 0;  int prod\_id = 0;  int count = Convert.ToInt32(numericUpDown1.Value);  decimal price;  bool isNumber1 = decimal.TryParse(textBox5.Text, out price);  // Поиск Категории\_ID.  var qwery1 = $"select ID from Категория where Наименование = '{comboBoxCat.Text}'";  var command = new OleDbCommand(qwery1, database.getConnection());  OleDbDataReader reader = command.ExecuteReader();  while (reader.Read())  {  type\_id = reader.GetInt32(0);  }  reader.Close();  // Поиск ВидУпаковки\_ID.  var qwery2 = $"select ID from ВидУпаковки where Наименование = '{comboBoxVid.Text}'";  var command2 = new OleDbCommand(qwery2, database.getConnection());  OleDbDataReader reader2 = command2.ExecuteReader();  while (reader2.Read())  {  Vid\_id = reader2.GetInt32(0);  }  reader2.Close();  // Поиск Производитель\_ID.  var qwery3 = $"select ID from Производитель where Название = '{comboBoxPro.Text}'";  var command3 = new OleDbCommand(qwery3, database.getConnection());  OleDbDataReader reader3 = command3.ExecuteReader();  while (reader3.Read())  {  prod\_id = reader3.GetInt32(0);  }  reader3.Close();  // Проверка на не пустоту строк и запрос на добавление новой строки в бд.  if (isNumber1 == true && name != "" && Srok != "" && num != "" && Srok.Length<11 && num.Length<45 && type\_id>0 && Vid\_id >0 && prod\_id>0)  {  var addQwery = $"insert into Лекарство (Название, Дата, СрокГодности, РегНомер, Категория\_ID, ВидУпаковки\_ID, Производитель\_ID, Количество, Цена ) values ('{name}', '{dat}', '{Srok}', '{num}', {type\_id}, {Vid\_id}, {prod\_id}, {count}, {price})";  var command4 =new OleDbCommand(addQwery, database.getConnection());  command4.ExecuteNonQuery();  MessageBox.Show("Запись успешно создана!", "Создание записи", MessageBoxButtons.OK, MessageBoxIcon.Information);  textBox1.Text = "";  textBox3.Text = "";  textBox4.Text = "";  textBox5.Text = "";  comboBoxCat.Text = "";  dateTimePicker1.Text = "";  comboBoxVid.Text = "";  comboBoxPro.Text = "";  }  else  {  MessageBox.Show("Неправильный ввод", "Создание записи", MessageBoxButtons.OK, MessageBoxIcon.Warning);  }  database.closeConnection();  }  }  } |

Таблица 4 - содержимое решения Catalog

|  |
| --- |
| Содержимое модуля “Class1.cs” |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using Invoices;  namespace MyProj  {  public class Class1  {  static EventHandler InvoicesF()  {  return InvoicesF\_Click;  }  private static void InvoicesF\_Click(object sender, EventArgs e)  {  InvoicesForm form = new InvoicesForm();  form.Show();  }  }  } |
| Содержимое модуля “InvoicesForm.cs” |
| using System;  using database;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Drawing;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Windows.Forms;  using Microsoft.Office.Interop.Excel;  using System.Data.OleDb;  using System.Diagnostics;  namespace Invoices  {  public partial class InvoicesForm : Form  {  int selectedRow;  public bool W, E, D;  public int id\_user;  DataB b = new DataB();  public InvoicesForm()  {  InitializeComponent();  StartPosition = FormStartPosition.CenterScreen;  RightsFunction();  if (W == true && E == true)  {  button3.Enabled = true;  }  }  // Авторизация прав пользователя на меню Производитель.  private void RightsFunction()  {  string qwertystring = $"select \* from Rights where ID\_Menu = 5";  OleDbCommand command = new OleDbCommand(qwertystring, b.getConnection());  b.openConnection();  OleDbDataReader reader = command.ExecuteReader();  while (reader.Read())  {  W = Convert.ToBoolean(reader.GetBoolean(2));  E = Convert.ToBoolean(reader.GetBoolean(3));  id\_user = reader.GetInt32(5);  }  reader.Close();  b.closeConnection();  }  private void CreateColums()  {  dataGridView1.Columns.Add("ID", "ID\_Заказа");  dataGridView1.Columns.Add("pokup\_id", "Покупатель\_ID");  dataGridView1.Columns.Add("data", "Дата Выписки");  dataGridView1.Columns.Add("secondName", "Фамилия продавца");  dataGridView1.Columns.Add("status", "Статус");  dataGridView1.Columns[1].Visible = false;  }  // Отчистка таблицы "Запись".  private void ClearFields()  {  dateTimePicker1.Text = "";  textBox3.Text = "";  }  // Поиск данных в базе данных.  private void RefreshDataGrid(DataGridView dgw)  {  b.openConnection();  // Кол-во активных заказов.  string qwerystring3 = $"select \* from `Счёт-фактуры` order by Заказ\_ID";  OleDbCommand command3 = new OleDbCommand(qwerystring3, b.getConnection());  OleDbDataReader reader3 = command3.ExecuteReader();  List<int> zak\_aktiv= new List<int>();  List<int> zak\_sform = new List<int>();  int last\_id = 0;  while (reader3.Read())  {  if (last\_id != reader3.GetInt32(3))  {  if(reader3.GetString(4).Trim() == "Активен")  {  zak\_aktiv.Add(reader3.GetInt32(3));  }  if(reader3.GetString(4).Trim() == "Сформирован")  {  zak\_sform.Add(reader3.GetInt32(3));  }    }  last\_id = reader3.GetInt32(3);  }  reader3.Close();  if (W == true && E == true)  {  for (int i = 0; i < zak\_aktiv.Count; i++)  {  // Поиск Покупателя\_ID.  string qwerystring = $"select \* from Заказ where ID = {zak\_aktiv[i]}";  OleDbCommand command = new OleDbCommand(qwerystring, b.getConnection());  OleDbDataReader reader = command.ExecuteReader();  while (reader.Read())  {  dgw.Rows.Add(zak\_aktiv[i], reader.GetInt32(1), "-", "-", "Активен");  }  reader.Close();  }  for (int i = 0; i < zak\_sform.Count; i++)  {  // Поиск Покупателя\_ID.  string qwerystring2 = $"select \* from Заказ where ID = {zak\_sform[i]}";  OleDbCommand command2 = new OleDbCommand(qwerystring2, b.getConnection());  OleDbDataReader reader2 = command2.ExecuteReader();  while (reader2.Read())  {  dgw.Rows.Add(zak\_sform[i], reader2.GetInt32(1), reader2.GetDateTime(2), reader2.GetString(3).Trim(), "Сформирован");  }  reader2.Close();  }  }  else  {  for (int i = 0; i < zak\_sform.Count; i++)  {  // Поиск Покупателя\_ID.  string qwerystring2 = $"select \* from Заказ where ID = {zak\_sform[i]} and Покупатель\_ID = {id\_user}";  OleDbCommand command2 = new OleDbCommand(qwerystring2, b.getConnection());  OleDbDataReader reader2 = command2.ExecuteReader();  while (reader2.Read())  {  dgw.Rows.Add(zak\_sform[i], reader2.GetInt32(1), reader2.GetDateTime(2), reader2.GetString(3).Trim(), "Сформирован");  }  reader2.Close();  }  }  b.closeConnection();  }  // Нажатие на клавишу "Обновить".  private void button2\_Click(object sender, EventArgs e)  {  DialogResult result = MessageBox.Show("Сохранить изменения?", "Сохрание", MessageBoxButtons.YesNo, MessageBoxIcon.Question);  switch (result)  {  case DialogResult.Yes:  {  Update();  RefreshDataGrid(dataGridView1);  ClearFields();  break;  }  case DialogResult.No:  {  RefreshDataGrid(dataGridView1);  ClearFields();  break;  }  } }  private void InvoicesForm\_Load(object sender, EventArgs e)  {  CreateColums();  RefreshDataGrid(dataGridView1);  }  // Запись данных в таблицу "Запись".  private void dataGridView1\_CellClick(object sender, DataGridViewCellEventArgs e)  {  selectedRow = e.RowIndex;  if (e.RowIndex >= 0)  {  if(W == true && E == true)  {  button3.Enabled = true;  }  DataGridViewRow row = dataGridView1.Rows[selectedRow];  textBox1.Text = row.Cells[0].Value.ToString().Trim();  textBox2.Text = row.Cells[1].Value.ToString().Trim();  if(row.Cells[2].Value.ToString().Trim() != "-")  {  dateTimePicker1.Text = row.Cells[2].Value.ToString().Trim();  }  else  {  dateTimePicker1.Text = "";  }  if (row.Cells[3].Value.ToString().Trim() != "-")  {  textBox3.Text = row.Cells[3].Value.ToString().Trim();  button4.Enabled = true;  }  else  {  textBox3.Text = "";  }  if(row.Cells[4].Value.ToString().Trim() == "Сформирован")  {  button4.Enabled = true;  }  else  {  button4.Enabled = false;  }  }  }  private void button1\_Click(object sender, EventArgs e)  {  ClearFields();  }  private void button3\_Click(object sender, EventArgs e)  {    int selectedRowindex = dataGridView1.CurrentCell.RowIndex;    // Проверка ввода.  if (textBox3.Text.Length >0 && textBox3.Text != "-")  {  dataGridView1.Rows[selectedRowindex].Cells[2].Value = dateTimePicker1.Value;  dataGridView1.Rows[selectedRowindex].Cells[3].Value = textBox3.Text;  dataGridView1.Rows[selectedRowindex].Cells[4].Value = "Сформирован";  // Обновление данных в таблице Заказ.  b.openConnection();  string changeQwery = $"update Заказ set ДатаВыписки = '{dataGridView1.Rows[selectedRowindex].Cells[2].Value}', ФамилияПродавца = '{textBox3.Text}' where ID ={textBox1.Text.Trim()}";  var command2 = new OleDbCommand(changeQwery, b.getConnection());  command2.ExecuteNonQuery();  b.closeConnection();  // Обновление данных в таблице Счёт-фактуры.  b.openConnection();  string str = $"update `Счёт-фактуры` set Статус = 'Сформирован' where Заказ\_ID ={textBox1.Text.Trim()}";  var comm = new OleDbCommand(str, b.getConnection());  comm.ExecuteNonQuery();  b.closeConnection();  }  else  {  MessageBox.Show("Неправильный ввод", "Внесение данных", MessageBoxButtons.OK, MessageBoxIcon.Warning);  }  button3.Enabled = false;  ClearFields();  }  // Экспорт файла.  private void button4\_Click(object sender, EventArgs e)  {  // Создание объекта для запуска Excel.  Microsoft.Office.Interop.Excel.Application ExcelApp = new Microsoft.Office.Interop.Excel.Application();  // Создаём рабочую книгу.  ExcelApp.Application.Workbooks.Add(Type.Missing);  ExcelApp.Cells[1, 1] = "ID\_Заказа";  ExcelApp.Cells[2, 1] = "Дата Выписки";  ExcelApp.Cells[3, 1] = "Фамилия Продавца";  ExcelApp.Cells[4, 1] = "Сумма к уплате";  ExcelApp.Cells[6, 1] = "Лекарство";  ExcelApp.Cells[6, 2] = "Количество";  ExcelApp.Cells[6, 3] = "Цена";  ExcelApp.Cells[6, 4] = "Сумма";  ExcelApp.Columns.ColumnWidth = 18;  int selectedRowindex = dataGridView1.CurrentCell.RowIndex;  ExcelApp.Cells[1, 2] = dataGridView1.Rows[selectedRowindex].Cells[0].Value;  ExcelApp.Cells[2, 2] = dataGridView1.Rows[selectedRowindex].Cells[2].Value;  ExcelApp.Cells[3, 2] = dataGridView1.Rows[selectedRowindex].Cells[3].Value;  // Информация по заказу.  List<decimal> sum = new List<decimal>();  List<decimal> price = new List<decimal>();  List<int> kol = new List<int>();  List<int> lek\_id = new List<int>();  List<string> lek = new List<string>();  string qwertystring3 = $"select \* from `Счёт-фактуры` where Заказ\_ID = {dataGridView1.Rows[selectedRowindex].Cells[0].Value}";  b.openConnection();  OleDbCommand command3 = new OleDbCommand(qwertystring3, b.getConnection());  OleDbDataReader reader3 = command3.ExecuteReader();  while (reader3.Read())  {  lek\_id.Add(reader3.GetInt32(1));  kol.Add(reader3.GetInt32(2));  }  reader3.Close();  // Запрос на получение данных о Лекарстве.  for(int i=0; i< lek\_id.Count; i++)  {  string qwertystring = $"select \* from Лекарство where ID = {lek\_id[i]}";  OleDbCommand command = new OleDbCommand(qwertystring, b.getConnection());  OleDbDataReader reader = command.ExecuteReader();  while (reader.Read())  {  lek.Add(reader.GetString(1));  price.Add(reader.GetDecimal(9));  }  reader.Close();  }  b.closeConnection();  // Вывод данных в Excel  for (int i = 0; i < lek.Count; i++)  {  ExcelApp.Cells[i + 7, 1] = lek[i];  ExcelApp.Cells[i + 7, 2] = kol[i];  ExcelApp.Cells[i + 7, 3] = price[i];  sum.Add(kol[i] \* price[i]);  ExcelApp.Cells[i + 7, 4] = sum[i];  }  // Подсчёт суммы.  ExcelApp.Cells[4, 2] = sum.ToArray().Sum();  ExcelApp.Visible = true;  b.closeConnection();  MessageBox.Show("Таблица успешно создана", "Создание таблицы", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  }  } |

Таблица 5 - содержимое решения Invoices

|  |
| --- |
| Содержимое модуля “Class1.cs” |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Security.Cryptography;  using System.Text;  using System.Threading.Tasks;  namespace Md5  {  // Класс для шифрования пароля.  public static class md5  {  public static string hashPassword(string password)  {  MD5 md5 = MD5.Create();  // Перевод пароля в байты.  byte[] b = Encoding.ASCII.GetBytes(password);  // Перевод в хеш функцию.  byte[] hash = md5.ComputeHash(b);  StringBuilder sb = new StringBuilder();  foreach (var a in hash)  sb.Append(a.ToString("X2"));  return Convert.ToString(sb);  }  }  } |

Таблица 6 - содержимое решения MD5

|  |
| --- |
| Содержимое модуля “Class1.cs” |
| using Order;  using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Windows.Forms;  using static System.ActivationContext;  namespace MyProj  {  public class Class1  {  static EventHandler OrderF()  {  return OrderF\_Click;  }  private static void OrderF\_Click(object sender, EventArgs e)  {  DialogResult result = MessageBox.Show("Вы хотите сформировать новый заказ?", "Заказ", MessageBoxButtons.YesNo , MessageBoxIcon.Question);  switch (result)  {  case DialogResult.Yes:  {  OrderForm form = new OrderForm();  form.Show();  break;  }    case DialogResult.No:  break;  }  }  }  } |
| Содержимое модуля “OrderForm.cs” |
| namespace Order  {  public partial class OrderForm : Form  {  // Перечисление для состояния строки.  enum RowState  {  Existed,  New,  Modifide,  ModifideNew,  Deleted  }  DataB b = new DataB();  // Переменная для работы с DataGridView.  int selectedRow;  public bool W, E, D;  public int id\_user;  public int id\_zak;  public OrderForm()  {  InitializeComponent();  StartPosition = FormStartPosition.CenterScreen;  RightsFunction();  if (W == false)  {  buttonNew.Enabled = false;  }  if (E == false)  {  buttonChange.Enabled = false;  }  if (D == false)  {  buttonDel.Enabled = false;  }  // Получение ID пользователя.  string qwertystring = $"select \* from Rights where ID\_Menu = 8";  OleDbCommand command = new OleDbCommand(qwertystring, b.getConnection());  b.openConnection();  OleDbDataReader reader = command.ExecuteReader();  while (reader.Read())  {  id\_user = Convert.ToInt32(reader.GetInt32(5));  }  reader.Close();    // Добавление заказа.  string addQwery = $"insert into Заказ (Покупатель\_ID) values ({id\_user})";  var command4 = new OleDbCommand(addQwery, b.getConnection());  command4.ExecuteNonQuery();  // Получение ID созданного заказа.  string qwerty = $"SELECT \* FROM Заказ WHERE Покупатель\_ID = {id\_user} AND ID = (SELECT MAX(ID) FROM Заказ)";  OleDbCommand command2 = new OleDbCommand(qwerty, b.getConnection());  OleDbDataReader reader2 = command2.ExecuteReader();  while (reader2.Read())  {  id\_zak = Convert.ToInt32(reader2.GetInt32(0));  }  reader2.Close();  b.closeConnection();  }  // Авторизация прав пользователя на меню Производитель.  private void RightsFunction()  {  string qwertystring = $"select \* from Rights where ID\_Menu = 8";  OleDbCommand command = new OleDbCommand(qwertystring, b.getConnection());  b.openConnection();  OleDbDataReader reader = command.ExecuteReader();  while (reader.Read())  {  W = Convert.ToBoolean(reader.GetBoolean(2));  E = Convert.ToBoolean(reader.GetBoolean(3));  D = Convert.ToBoolean(reader.GetBoolean(4));  }  reader.Close();  b.closeConnection();  }  // Создание столбцов в dataGridView.  private void CreateColums()  {  dataGridView1.Columns.Add("ID", "ID");  dataGridView1.Columns.Add("lekar", "Лекарство");  dataGridView1.Columns.Add("kol", "Количество");  dataGridView1.Columns.Add("zak\_id", "Заказ\_id");  dataGridView1.Columns.Add("Status", "Статус");  // Статус строки.  dataGridView1.Columns.Add("IsNew", String.Empty);  dataGridView1.Columns[0].Visible = false;  dataGridView1.Columns[3].Visible = false;  dataGridView1.Columns[4].Visible = false;  dataGridView1.Columns[5].Visible = false;  }  // Отчистка таблицы "Запись".  private void ClearFields()  {  textBox1.Text = "";  numericUpDown1.Value = 1;  textBox2.Text = "";  label3.Visible = false;  }  // Добавление Строки из базы данных в DataGridView.  private void ReadSingleRow(DataGridView dgw, IDataRecord record)  {  dgw.Rows.Add(record.GetInt32(0), record.GetInt32(1), record.GetInt32(2), record.GetInt32(3), record.GetString(4), RowState.ModifideNew);  }  // Поиск данных в базе данных.  private void RefreshDataGrid(DataGridView dgw)  {  dgw.Rows.Clear();  string qwertystring = $"select \* from `Счёт-фактуры` where Заказ\_ID = {id\_zak}";  OleDbCommand command = new OleDbCommand(qwertystring, b.getConnection());  b.openConnection();  OleDbDataReader reader = command.ExecuteReader();  while (reader.Read())  {  ReadSingleRow(dgw, reader);  }  reader.Close();  b.closeConnection();  }  private void OrderForm\_Load(object sender, EventArgs e)  {  CreateColums();  }  // Запись данных в таблицу "Запись".  private void dataGridView1\_CellClick(object sender, DataGridViewCellEventArgs e)  {  selectedRow = e.RowIndex;  if (e.RowIndex >= 0)  {  b.openConnection();  DataGridViewRow row = dataGridView1.Rows[selectedRow];  numericUpDown1.Value = Convert.ToInt32(row.Cells[2].Value.ToString().Trim());  var id = row.Cells[1].Value;  // Поиск Лекарства по id для получения названия и количества.  var qwery1 = $"select \* from Лекарство where ID = {id}";  var command = new OleDbCommand(qwery1, b.getConnection());  OleDbDataReader reader = command.ExecuteReader();  while (reader.Read())  {  textBox2.Text = reader["Название"].ToString();  label3.Visible = true;  textBox1.Text = reader["Цена"].ToString();  label3.Text = reader["Количество"].ToString();  }  reader.Close();  b.closeConnection();  }  }  // Нажатие на клавишу "Обновить"  private void button2\_Click(object sender, EventArgs e)  {  DialogResult result = MessageBox.Show("Сохранить изменения?", "Сохрание", MessageBoxButtons.YesNo, MessageBoxIcon.Question);  switch (result)  {  case DialogResult.Yes:  {  Update();  RefreshDataGrid(dataGridView1);  ClearFields();  break;  }  case DialogResult.No:  {  RefreshDataGrid(dataGridView1);  ClearFields();  break;  }  } }  // Событие "Добавление записи".  private void buttonNew\_Click(object sender, EventArgs e)  {  AddLekZak addfrm = new AddLekZak(id\_user, id\_zak, dataGridView1);  RefreshDataGrid(dataGridView1);  addfrm.Show();  }  // Метод обнавляет, удаляет или ничего не делает с данными в бд в соответствии со статусом каждой строки в DataGridView.  private void Update()  {  b.openConnection();  for (int i = 0; i < dataGridView1.Rows.Count; i++)  {  var rowState = (RowState)dataGridView1.Rows[i].Cells[5].Value;  if (rowState == RowState.Existed) continue;  if (rowState == RowState.Deleted)  {  var id = dataGridView1.Rows[i].Cells[0].Value.ToString().Trim();  var deleteQwery = $"delete from `Счёт-фактуры` where ID = {id}";  var command = new OleDbCommand(deleteQwery, b.getConnection());  command.ExecuteNonQuery();  }  if (rowState == RowState.Modifide)  {  var id = dataGridView1.Rows[i].Cells[0].Value.ToString().Trim();  var kol = dataGridView1.Rows[i].Cells[2].Value.ToString().Trim();  // Обновление данных  string changeQwery = $"update `Счёт-фактуры` set Количество = {kol} where ID ={id}";  var command2 = new OleDbCommand(changeQwery, b.getConnection());  command2.ExecuteNonQuery();  }  }  b.closeConnection();  }  // Метод делает строку невидимой и присваивает ей статус "Удален"  private void deleteRow()  {  int index = dataGridView1.CurrentCell.RowIndex;  dataGridView1.Rows[index].Visible = false;  if (Convert.ToInt32(dataGridView1.Rows[index].Cells[0].Value) >= 0)  {  dataGridView1.Rows[index].Cells[5].Value = RowState.Deleted;  }  }  // Событие "Удаление записи".  private void buttonDel\_Click(object sender, EventArgs e)  {  deleteRow();  ClearFields();  }  // Событие "Сохраниение".  private void buttonSave\_Click(object sender, EventArgs e)  {  Update();  }  private void Change()  {  int selectedRowindex = dataGridView1.CurrentCell.RowIndex;  b.openConnection();  int kol = Convert.ToInt32(numericUpDown1.Value);  // Проверка ввода.  if (kol <= Convert.ToInt32(label3.Text))  {  dataGridView1.Rows[selectedRowindex].Cells[2].Value = kol;  dataGridView1.Rows[selectedRowindex].Cells[5].Value = RowState.Modifide;  }  else  {  MessageBox.Show("Неправильный ввод", "Создание записи", MessageBoxButtons.OK, MessageBoxIcon.Warning);  }  b.closeConnection();  }  private void button3\_Click(object sender, EventArgs e)  {  b.openConnection();  // Обновление cтатусов у всех позиций заказа  string changeQwery = $"update `Счёт-фактуры` set Статус = 'Активен' where Заказ\_ID ={id\_zak}";  var command2 = new OleDbCommand(changeQwery, b.getConnection());  command2.ExecuteNonQuery();  b.closeConnection();  MessageBox.Show("Заказ сформирован! Вам перезвонит менеджер для оплаты заказа.", "Заказ", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  private void buttonChange\_Click(object sender, EventArgs e)  {  Change();  ClearFields();  }  // Нажатие на клавишу "Отчистка"  private void button1\_Click(object sender, EventArgs e)  {  ClearFields();  }  }  } |
| Содержимое модуля “AddLekZak.cs” |
| using database;  using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Data.OleDb;  using System.Drawing;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Windows.Forms;  using static System.Windows.Forms.VisualStyles.VisualStyleElement;  namespace Order  {  public partial class AddLekZak : Form  {  DataB database = new DataB();  public int id\_Zak;  public int id\_User;  public AddLekZak(int id\_user, int id\_zak, DataGridView dgw)  {  id\_User = id\_user;  id\_Zak = id\_zak;  InitializeComponent();  StartPosition = FormStartPosition.CenterScreen;  AddItems(dgw);  }  // Добавление лекарств в комбобокс.  private void AddItems(DataGridView dgw)  {  List<int> Name\_in\_dgw = new List<int>();  for(int i =0; i< dgw.Rows.Count; i++)  {  Name\_in\_dgw.Add(Convert.ToInt32(dgw.Rows[i].Cells[1].Value));  }  database.openConnection();  // Поиск Лекарств из бд.  var qwery1 = $"select \* from Лекарство";  var command = new OleDbCommand(qwery1, database.getConnection());  OleDbDataReader reader = command.ExecuteReader();  while (reader.Read())  {  // Если Лекарство уже есть в Заказе, то не добавляем его в combobox.  if (Name\_in\_dgw.Contains(Convert.ToInt32(reader.GetInt32(0))))  continue;  else  comboBox1.Items.Add(reader["Название"].ToString().Trim());  }  reader.Close();  database.closeConnection();  }  private void button1\_Click(object sender, EventArgs e)  {  database.openConnection();  var kol = numericUpDown1.Value;  int lek\_id = 0;  // Поиск Улица\_ID.  int kol\_in\_kat = 0;  var qwery1 = $"select \* from Лекарство where Название = '{comboBox1.Text}'";  var command = new OleDbCommand(qwery1, database.getConnection());  OleDbDataReader reader = command.ExecuteReader();  while (reader.Read())  {  lek\_id = reader.GetInt32(0);  kol\_in\_kat = reader.GetInt32(8);  }  reader.Close();  string addQwery;  // Проверка на не пустоту строк и запрос на добавление новой строки в бд.  if (lek\_id>=0 )  {  if(kol <= kol\_in\_kat)  {  addQwery = $"insert into `Счёт-фактуры` (Лекарство\_ID, Количество, Заказ\_ID, Статус) values ({lek\_id}, {kol}, {id\_Zak}, 'Добавлен')";  var command4 = new OleDbCommand(addQwery, database.getConnection());  command4.ExecuteNonQuery();  MessageBox.Show("Запись успешно создана!", "Создание записи", MessageBoxButtons.OK, MessageBoxIcon.Information);  comboBox1.Text = "";  numericUpDown1.Value = 1;  Close();  }  else  {  MessageBox.Show("Такого количества нет на складе!", "Создание записи", MessageBoxButtons.OK, MessageBoxIcon.Warning);  }  }  else  {  MessageBox.Show("Не выбран пункт!", "Создание записи", MessageBoxButtons.OK, MessageBoxIcon.Warning);  }  database.closeConnection();  }  }  } |

Таблица 7 - содержимое решения Order

|  |
| --- |
| Содержимое модуля “Class1.cs” |
| namespace MyProj  {  public class Class1  {  static EventHandler PassF()  {  return PassF\_Click;  }  private static void PassF\_Click(object sender, EventArgs e)  {  PassForm form = new PassForm();  form.Show();  }  }  } |
| Содержимое модуля “PassForm.cs” |
| namespace Pass  {  public partial class PassForm : Form  {  // Таймер для постоянной проверки состояния формы.  System.Windows.Forms.Timer formTimer = new System.Windows.Forms.Timer();  bool W, E;  DataB dataBase = new DataB();  public PassForm()  {  InitializeComponent();  StartPosition = FormStartPosition.CenterScreen;  RightsFunction();  if (W == false && E == false)  {  button1.Enabled = false;  }  ActiveControl = textBox1;  }  // Авторизация прав пользователя на меню Смена пароля.  private void RightsFunction()  {  string qwertystring = $"select \* from Rights where ID\_Menu = 3";  OleDbCommand command = new OleDbCommand(qwertystring, dataBase.getConnection());  dataBase.openConnection();  OleDbDataReader reader = command.ExecuteReader();  while (reader.Read())  {  W = Convert.ToBoolean(reader.GetBoolean(2));  E = Convert.ToBoolean(reader.GetBoolean(3));  }  reader.Close();  dataBase.closeConnection();  }  private void PassForm\_Load(object sender, EventArgs e)  {  formTimer.Interval = 500;  formTimer.Start();  formTimer.Tick += new EventHandler(FormTimer\_Tick);  }  private void FormTimer\_Tick(object sender, EventArgs e)  {  toolStripStatusLabel2.Text = (Console.CapsLock ? "Клавиша CapsLock нажата" : "");  if (InputLanguage.CurrentInputLanguage.LayoutName == "США")  toolStripStatusLabel1.Text = "Язык ввода Английский";  else if (InputLanguage.CurrentInputLanguage.LayoutName == "Русская")  toolStripStatusLabel1.Text = "Язык ввода Русский";  }  private void button2\_Click(object sender, EventArgs e)  {  Close();  }  private void button1\_Click(object sender, EventArgs e)  {  // Считывание введенного логина и пароля.  var pass1 = md5.hashPassword(textBox1.Text);  var pass2 = md5.hashPassword(textBox2.Text);  var pass3 = md5.hashPassword(textBox3.Text);  int id=0;  string pass\_bd="";  // получение ID пользователя  var qwery1 = $"select \* from Rights where ID\_Menu = 3";  dataBase.openConnection();  var command = new OleDbCommand(qwery1, dataBase.getConnection());  OleDbDataReader reader = command.ExecuteReader();  while (reader.Read())  {  id = Convert.ToInt32(reader.GetInt32(5));  }  dataBase.closeConnection();  reader.Close();  // получение старого пароля по ID пользователя  var qwery2 = $"select \* from Users where ID = {id}";  dataBase.openConnection();  var command2 = new OleDbCommand(qwery2, dataBase.getConnection());  OleDbDataReader reader2 = command2.ExecuteReader();  while (reader2.Read())  {  pass\_bd = reader2.GetString(2).Trim();  }  dataBase.closeConnection();  reader2.Close();  // Проверка на правильность прежнего пароля.  if ( pass1 == pass\_bd)  {  // Проверка на равенство.  if (pass2 == pass3)  {  dataBase.openConnection();  string changeQwery = $"update Users set Пароль = '{pass3}' where ID ={id}";  var command3 = new OleDbCommand(changeQwery, dataBase.getConnection());  command3.ExecuteNonQuery();  MessageBox.Show("Пароль изменён!", "ОК", MessageBoxButtons.OK, MessageBoxIcon.Information);  Close();  }  else  {  MessageBox.Show("Введены разные пароли", "Ошибка ввода", MessageBoxButtons.OK, MessageBoxIcon.Warning);  textBox2.Text = "";  textBox3.Text = "";  }  }  else  {  MessageBox.Show("Прежний пароль введен неправтльно!", "Ошибка ввода", MessageBoxButtons.OK, MessageBoxIcon.Warning);  textBox1.Text = "";  textBox2.Text = "";  textBox3.Text = "";  }    }  }  } |

Таблица 8 - содержимое решения Pass

|  |
| --- |
| Содержимое модуля “Class1.cs” |
| using Setting;  using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Windows.Forms;  namespace MyProj  {  public class Class1  {  static EventHandler SettingF()  {  return SettingF\_Click;  }  private static void SettingF\_Click(object sender, EventArgs e)  {  SettingForm form = new SettingForm();  form.Show();  }  }  } |
| Содержимое модуля “AddUserForm.cs” |
| namespace Setting  {  public partial class AddUserForm : Form  {  DataB database = new DataB();  public AddUserForm()  {  InitializeComponent();  StartPosition = FormStartPosition.CenterScreen;  }  private void button1\_Click(object sender, EventArgs e)  {  database.openConnection();  int id = 0;  var name = textBox1.Text;  var pass1 = textBox2.Text;  var pass2 = textBox3.Text;  // Проверка на не пустоту строк и запрос на добавление новой строки в бд.  if (1<=name.Length && name.Length <= 20 && pass1.Length >=0 && pass1 == pass2)  {  string qwerty = $"select \* from Users where Логин = '{name}'";  OleDbCommand com = new OleDbCommand(qwerty, database.getConnection());  OleDbDataReader rea = com.ExecuteReader();  int count = 0;  while (rea.Read())  {  count++;  }  rea.Close();  if (count == 0)  {  string addQwery = $"insert into Users (Логин, Пароль) values ('{name}', '{md5.hashPassword(pass1)}')";  var comm = new OleDbCommand(addQwery, database.getConnection());  comm.ExecuteNonQuery();  string qwertystring = $"select ID from Users where Логин = '{name}'";  OleDbCommand command = new OleDbCommand(qwertystring, database.getConnection());  OleDbDataReader reader = command.ExecuteReader();  while (reader.Read())  {  id = reader.GetInt32(0);  }  reader.Close();  // Добавление стандартных прав пользователя.  string addRight1 = $"insert into AccessRights(ID\_User, ID\_Menu, R, W, E, D) values({id}, 1, 1, 0, 0, 0)";  string addRight2 = $"insert into AccessRights(ID\_User, ID\_Menu, R, W, E, D) values({id}, 2, 1, 0, 0, 0)";  string addRight3 = $"insert into AccessRights(ID\_User, ID\_Menu, R, W, E, D) values({id}, 3, 1, 1, 1, 1)";  string addRight4 = $"insert into AccessRights(ID\_User, ID\_Menu, R, W, E, D) values({id}, 4, 1, 0, 0, 0)";  string addRight5 = $"insert into AccessRights(ID\_User, ID\_Menu, R, W, E, D) values({id}, 5, 1, 0, 0, 0)";  string addRight6 = $"insert into AccessRights(ID\_User, ID\_Menu, R, W, E, D) values({id}, 6, 0, 0, 0, 0)";  string addRight7 = $"insert into AccessRights(ID\_User, ID\_Menu, R, W, E, D) values({id}, 7, 1, 0, 0, 0)";  string addRight8 = $"insert into AccessRights(ID\_User, ID\_Menu, R, W, E, D) values({id}, 8, 1, 1, 1, 1)";  string addRight9 = $"insert into AccessRights(ID\_User, ID\_Menu, R, W, E, D) values({id}, 9, 0, 0, 0, 0)";  string addRight10 = $"insert into AccessRights(ID\_User, ID\_Menu, R, W, E, D) values({id}, 10, 0, 0, 0, 0)";  string addRight11 = $"insert into AccessRights(ID\_User, ID\_Menu, R, W, E, D) values({id}, 14, 0, 0, 0, 0)";  string addRight12 = $"insert into AccessRights(ID\_User, ID\_Menu, R, W, E, D) values({id}, 15, 1, 0, 0, 0)";  string addRight13 = $"insert into AccessRights(ID\_User, ID\_Menu, R, W, E, D) values({id}, 16, 1, 0, 0, 0)";  string addRight14 = $"insert into AccessRights(ID\_User, ID\_Menu, R, W, E, D) values({id}, 17, 1, 0, 0, 0)";  string addRight15 = $"insert into AccessRights(ID\_User, ID\_Menu, R, W, E, D) values({id}, 18, 1, 0, 0, 0)";  string addRight16 = $"insert into AccessRights(ID\_User, ID\_Menu, R, W, E, D) values({id}, 19, 1, 0, 0, 0)";  string addRight17 = $"insert into AccessRights(ID\_User, ID\_Menu, R, W, E, D) values({id}, 21, 1, 0, 0, 0)";  string addRight18 = $"insert into AccessRights(ID\_User, ID\_Menu, R, W, E, D) values({id}, 22, 1, 0, 0, 0)";  string addRight19 = $"insert into AccessRights(ID\_User, ID\_Menu, R, W, E, D) values({id}, 23, 1, 0, 0, 0)";  var command1 = new OleDbCommand(addRight1, database.getConnection());  var command2 = new OleDbCommand(addRight2, database.getConnection());  var command3 = new OleDbCommand(addRight3, database.getConnection());  var command4 = new OleDbCommand(addRight4, database.getConnection());  var command5 = new OleDbCommand(addRight5, database.getConnection());  var command6 = new OleDbCommand(addRight6, database.getConnection());  var command7 = new OleDbCommand(addRight7, database.getConnection());  var command8 = new OleDbCommand(addRight8, database.getConnection());  var command9 = new OleDbCommand(addRight9, database.getConnection());  var command10 = new OleDbCommand(addRight10, database.getConnection());  var command11 = new OleDbCommand(addRight11, database.getConnection());  var command12 = new OleDbCommand(addRight12, database.getConnection());  var command13 = new OleDbCommand(addRight13, database.getConnection());  var command14 = new OleDbCommand(addRight14, database.getConnection());  var command15 = new OleDbCommand(addRight15, database.getConnection());  var command16 = new OleDbCommand(addRight16, database.getConnection());  var command17 = new OleDbCommand(addRight17, database.getConnection());  var command18 = new OleDbCommand(addRight18, database.getConnection());  var command19 = new OleDbCommand(addRight19, database.getConnection());  command1.ExecuteNonQuery();  command2.ExecuteNonQuery();  command3.ExecuteNonQuery();  command4.ExecuteNonQuery();  command5.ExecuteNonQuery();  command6.ExecuteNonQuery();  command7.ExecuteNonQuery();  command8.ExecuteNonQuery();  command9.ExecuteNonQuery();  command10.ExecuteNonQuery();  command11.ExecuteNonQuery();  command12.ExecuteNonQuery();  command13.ExecuteNonQuery();  command14.ExecuteNonQuery();  command15.ExecuteNonQuery();  command16.ExecuteNonQuery();  command17.ExecuteNonQuery();  command18.ExecuteNonQuery();  command19.ExecuteNonQuery();  MessageBox.Show("Пользователь создан!", "Создание записи", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  else  {  MessageBox.Show("Пользователь с таким логином уже существует.", "Создание записи", MessageBoxButtons.OK, MessageBoxIcon.Warning);  }  }  else  {  MessageBox.Show("Неправильный ввод", "Создание записи", MessageBoxButtons.OK, MessageBoxIcon.Warning);  }  database.closeConnection();  }  }  } |
| Содержимое модуля “ ChangeUserForm.cs” |
| using System;  using database;  using Md5;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Drawing;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Windows.Forms;  using System.Data.OleDb;  using System.Xml.Linq;  namespace Setting  {  public partial class ChangeUserForm : Form  {  DataB database = new DataB();  public int ID;  public ChangeUserForm(int id)  {  ID = id;  InitializeComponent();  StartPosition = FormStartPosition.CenterScreen;  string qwertystring = $"select \* from Users where ID = {id}";  OleDbCommand command = new OleDbCommand(qwertystring, database.getConnection());  database.openConnection();  OleDbDataReader reader = command.ExecuteReader();  while (reader.Read())  {  textBox1.Text = reader.GetString(1);  }  database.closeConnection();  }  private void button1\_Click(object sender, EventArgs e)  {  database.openConnection();    var pass1 = textBox3.Text;  var pass2 = textBox4.Text;  // Проверка на не пустоту строк и запрос на изменение строки в бд.  if (pass1.Length >= 0 && pass1 == pass2)  {  var changeQwery = $"update Users set Пароль = '{md5.hashPassword(pass1)}' where ID ={ID}";  var command = new OleDbCommand(changeQwery, database.getConnection());  command.ExecuteNonQuery();  MessageBox.Show("Пароль изменён!", "Изменение записи", MessageBoxButtons.OK, MessageBoxIcon.Information);  Close();  }  else  {  MessageBox.Show("Неправильный ввод", "Создание записи", MessageBoxButtons.OK, MessageBoxIcon.Warning);  }  database.closeConnection();  }  }  } |
| Содержимое модуля “SettingForm.cs” |
| using database;  using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Data.OleDb;  using System.Drawing;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Windows.Forms;  using System.Xml.Linq;  using static System.Windows.Forms.VisualStyles.VisualStyleElement;  namespace Setting  {  public partial class SettingForm : Form  {  DataB b = new DataB();  int id\_user;  bool W, E, D;  public SettingForm()  {  InitializeComponent();  StartPosition = FormStartPosition.CenterScreen;  RightsFunction();  if (W == false)  {  buttonNew.Enabled = false;  }  if (E == false)  {  buttonChange.Enabled = false;  }  if (D == false)  {  buttonDel.Enabled = false;  }  if (E == false || W == false)  {  buttonChoice.Enabled = false;  }  Get\_id\_user();  Add\_ComboboxItem();  buttonChange.Enabled = false;  buttonDel.Enabled = false;  button3.Enabled = false;  }  // Авторизация прав пользователя на меню Улица.  private void RightsFunction()  {  string qwertystring = $"select \* from Rights where ID\_Menu = 2";  OleDbCommand command = new OleDbCommand(qwertystring, b.getConnection());  b.openConnection();  OleDbDataReader reader = command.ExecuteReader();  while (reader.Read())  {  W = Convert.ToBoolean(reader.GetBoolean(2));  E = Convert.ToBoolean(reader.GetBoolean(3));  D = Convert.ToBoolean(reader.GetBoolean(4));  }  reader.Close();  b.closeConnection();  }  private void Get\_id\_user()  {  string qwertystring = $"select \* from Rights where ID\_Menu = 2";  OleDbCommand command = new OleDbCommand(qwertystring, b.getConnection());  b.openConnection();  OleDbDataReader reader = command.ExecuteReader();  while (reader.Read())  {  id\_user = Convert.ToInt32(reader.GetInt32(5));  }  reader.Close();  b.closeConnection();  }  private void Add\_ComboboxItem()  {  // Поиск Логинов и добавление вкладок.  var qwery1 = $"select \* from Users";  var command = new OleDbCommand(qwery1, b.getConnection());  b.openConnection();  OleDbDataReader reader = command.ExecuteReader();  while (reader.Read())  {  comboBox1.Items.Add(reader["Логин"].ToString().Trim());  }  reader.Close();  b.closeConnection();  }  // Отчистка таблицы.  private void ClearFields()  {  checkBox1\_1.Checked = false; checkBox1\_2.Checked = false; checkBox1\_3.Checked = false; checkBox1\_4.Checked = false;  checkBox2\_1.Checked = false; checkBox2\_2.Checked = false; checkBox2\_3.Checked = false; checkBox2\_4.Checked = false;  checkBox3\_1.Checked = false; checkBox3\_2.Checked = false; checkBox3\_3.Checked = false; checkBox3\_4.Checked = false;  checkBox4\_1.Checked = false; checkBox4\_2.Checked = false; checkBox4\_3.Checked = false; checkBox4\_4.Checked = false;  checkBox5\_1.Checked = false; checkBox5\_2.Checked = false; checkBox5\_3.Checked = false; checkBox5\_4.Checked = false;  checkBox6\_1.Checked = false; checkBox6\_2.Checked = false; checkBox6\_3.Checked = false; checkBox6\_4.Checked = false;  checkBox7\_1.Checked = false; checkBox7\_2.Checked = false; checkBox7\_3.Checked = false; checkBox7\_4.Checked = false;  checkBox8\_1.Checked = false; checkBox8\_2.Checked = false; checkBox8\_3.Checked = false; checkBox8\_4.Checked = false;  checkBox9\_1.Checked = false; checkBox9\_2.Checked = false; checkBox9\_3.Checked = false; checkBox9\_4.Checked = false;  checkBox10\_1.Checked = false; checkBox10\_2.Checked = false; checkBox10\_3.Checked = false; checkBox10\_4.Checked = false;  checkBox11\_1.Checked = false; checkBox11\_2.Checked = false; checkBox11\_3.Checked = false; checkBox11\_4.Checked = false;  checkBox12\_1.Checked = false; checkBox12\_2.Checked = false; checkBox12\_3.Checked = false; checkBox12\_4.Checked = false;  checkBox13\_1.Checked = false; checkBox13\_2.Checked = false; checkBox13\_3.Checked = false; checkBox13\_4.Checked = false;  checkBox14\_1.Checked = false; checkBox14\_2.Checked = false; checkBox14\_3.Checked = false; checkBox14\_4.Checked = false;  checkBox15\_1.Checked = false; checkBox15\_2.Checked = false; checkBox15\_3.Checked = false; checkBox15\_4.Checked = false;  }  // Нажатие на клавишу "Отчистка".  private void button1\_Click(object sender, EventArgs e)  {  ClearFields();  comboBox1.Text = "";  comboBox1.Items.Clear();  Add\_ComboboxItem();  comboBox1.Enabled = true;  buttonChange.Enabled = false;  buttonDel.Enabled = false;  button3.Enabled = false;  }  // Нажатие на "Сохранить".  private void button3\_Click(object sender, EventArgs e)  {  b.openConnection();  int id = 0;  string qwertystring = $"select \* from Users where Логин = '{comboBox1.Text.Trim()}'";  OleDbCommand command = new OleDbCommand(qwertystring, b.getConnection());  OleDbDataReader reader = command.ExecuteReader();  while (reader.Read())  {  id = reader.GetInt32(0);  }  b.closeConnection();  reader.Close();  if (id < 0)  {  MessageBox.Show("Пользователь не выбран", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Warning);  }  else  {  b.openConnection();  string changeQwery1 = $"update AccessRights set R = {checkBox1\_1.Checked}, W ={checkBox1\_2.Checked}, E = {checkBox1\_3.Checked}, D= {checkBox1\_4.Checked} where ID\_User = {id} and ID\_Menu =2";  string changeQwery2 = $"update AccessRights set R = {checkBox2\_1.Checked}, W ={checkBox2\_2.Checked}, E = {checkBox2\_3.Checked}, D= {checkBox2\_4.Checked} where ID\_User = {id} and ID\_Menu =3";  string changeQwery3 = $"update AccessRights set R = {checkBox3\_1.Checked}, W ={checkBox3\_2.Checked}, E = {checkBox3\_3.Checked}, D= {checkBox3\_4.Checked} where ID\_User = {id} and ID\_Menu =5";  string changeQwery4 = $"update AccessRights set R = {checkBox4\_1.Checked}, W ={checkBox4\_2.Checked}, E = {checkBox4\_3.Checked}, D= {checkBox4\_4.Checked} where ID\_User = {id} and ID\_Menu =6";  string changeQwery5 = $"update AccessRights set R = {checkBox5\_1.Checked}, W ={checkBox5\_2.Checked}, E = {checkBox5\_3.Checked}, D= {checkBox5\_4.Checked} where ID\_User = {id} and ID\_Menu =7";  string changeQwery6 = $"update AccessRights set R = {checkBox6\_1.Checked}, W ={checkBox6\_2.Checked}, E = {checkBox6\_3.Checked}, D= {checkBox6\_4.Checked} where ID\_User = {id} and ID\_Menu =8";  string changeQwery7 = $"update AccessRights set R = {checkBox7\_1.Checked}, W ={checkBox7\_2.Checked}, E = {checkBox7\_3.Checked}, D= {checkBox7\_4.Checked} where ID\_User = {id} and ID\_Menu =9";  string changeQwery8 = $"update AccessRights set R = {checkBox8\_1.Checked}, W ={checkBox8\_2.Checked}, E = {checkBox8\_3.Checked}, D= {checkBox8\_4.Checked} where ID\_User = {id} and ID\_Menu =10";  string changeQwery9 = $"update AccessRights set R = {checkBox9\_1.Checked}, W ={checkBox9\_2.Checked}, E = {checkBox9\_3.Checked}, D= {checkBox9\_4.Checked} where ID\_User = {id} and ID\_Menu =14";  string changeQwery10 = $"update AccessRights set R = {checkBox10\_1.Checked}, W ={checkBox10\_2.Checked}, E = {checkBox10\_3.Checked}, D= {checkBox10\_4.Checked} where ID\_User = {id} and ID\_Menu =16";  string changeQwery11 = $"update AccessRights set R = {checkBox11\_1.Checked}, W ={checkBox11\_2.Checked}, E = {checkBox11\_3.Checked}, D= {checkBox11\_4.Checked} where ID\_User = {id} and ID\_Menu =17";  string changeQwery12 = $"update AccessRights set R = {checkBox12\_1.Checked}, W ={checkBox12\_2.Checked}, E = {checkBox12\_3.Checked}, D= {checkBox12\_4.Checked} where ID\_User = {id} and ID\_Menu =18";  string changeQwery13 = $"update AccessRights set R = {checkBox13\_1.Checked}, W ={checkBox13\_2.Checked}, E = {checkBox13\_3.Checked}, D= {checkBox13\_4.Checked} where ID\_User = {id} and ID\_Menu =19";  string changeQwery14 = $"update AccessRights set R = {checkBox14\_1.Checked}, W ={checkBox14\_2.Checked}, E = {checkBox14\_3.Checked}, D= {checkBox14\_4.Checked} where ID\_User = {id} and ID\_Menu =22";  string changeQwery15 = $"update AccessRights set R = {checkBox15\_1.Checked}, W ={checkBox15\_2.Checked}, E = {checkBox15\_3.Checked}, D= {checkBox15\_4.Checked} where ID\_User = {id} and ID\_Menu =23";  var command1 = new OleDbCommand(changeQwery1, b.getConnection());  var command2 = new OleDbCommand(changeQwery2, b.getConnection());  var command3 = new OleDbCommand(changeQwery3, b.getConnection());  var command4 = new OleDbCommand(changeQwery4, b.getConnection());  var command5 = new OleDbCommand(changeQwery5, b.getConnection());  var command6 = new OleDbCommand(changeQwery6, b.getConnection());  var command7 = new OleDbCommand(changeQwery7, b.getConnection());  var command8 = new OleDbCommand(changeQwery8, b.getConnection());  var command9 = new OleDbCommand(changeQwery9, b.getConnection());  var command10 = new OleDbCommand(changeQwery10, b.getConnection());  var command11 = new OleDbCommand(changeQwery11, b.getConnection());  var command12 = new OleDbCommand(changeQwery12, b.getConnection());  var command13 = new OleDbCommand(changeQwery13, b.getConnection());  var command14 = new OleDbCommand(changeQwery14, b.getConnection());  var command15 = new OleDbCommand(changeQwery15, b.getConnection());  command1.ExecuteNonQuery();  command2.ExecuteNonQuery();  command3.ExecuteNonQuery();  command4.ExecuteNonQuery();  command5.ExecuteNonQuery();  command6.ExecuteNonQuery();  command7.ExecuteNonQuery();  command8.ExecuteNonQuery();  command9.ExecuteNonQuery();  command10.ExecuteNonQuery();  command11.ExecuteNonQuery();  command12.ExecuteNonQuery();  command13.ExecuteNonQuery();  command14.ExecuteNonQuery();  command15.ExecuteNonQuery();  b.closeConnection();  MessageBox.Show("Изменения внесены!", "Ок", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  }  private void buttonNew\_Click(object sender, EventArgs e)  {  AddUserForm frm = new AddUserForm();  frm.Show();  comboBox1.Text = "";  comboBox1.Items.Clear();  Add\_ComboboxItem();  comboBox1.Enabled = true;  buttonChange.Enabled = false;  buttonDel.Enabled = false;  button3.Enabled = false;  }  private void buttonDel\_Click(object sender, EventArgs e)  {  b.openConnection();  int id = 0;  string qwertystring = $"select \* from Users where Логин = '{comboBox1.Text.Trim()}'";  OleDbCommand command = new OleDbCommand(qwertystring, b.getConnection());  OleDbDataReader reader = command.ExecuteReader();  while (reader.Read())  {  id = reader.GetInt32(0);  }  reader.Close();  var deleteQwery = $"delete from Users where ID = {id}";  var command2 = new OleDbCommand(deleteQwery, b.getConnection());  command2.ExecuteNonQuery();  b.closeConnection();  MessageBox.Show("Пользователь удалён", "Удаление", MessageBoxButtons.OK, MessageBoxIcon.Information);  comboBox1.Text = "";  comboBox1.Items.Clear();  Add\_ComboboxItem();  comboBox1.Enabled = true;  buttonChange.Enabled = false;  buttonDel.Enabled = false;  button3.Enabled = false;  }  private void buttonChange\_Click(object sender, EventArgs e)  {  b.openConnection();  int id = 0;  string qwertystring = $"select \* from Users where Логин = '{comboBox1.Text.Trim()}'";  OleDbCommand command = new OleDbCommand(qwertystring, b.getConnection());  OleDbDataReader reader = command.ExecuteReader();  while (reader.Read())  {  id = reader.GetInt32(0);  }  reader.Close();  b.closeConnection();  ChangeUserForm frm = new ChangeUserForm(id);  frm.Show();  }  // Поиск данных в базе данных.  private void RefreshCheckBox(int id)  {    ClearFields();  string qwertystring = $"select \* from AccessRights where ID\_User = {id}";  OleDbCommand command = new OleDbCommand(qwertystring, b.getConnection());  b.openConnection();  OleDbDataReader reader = command.ExecuteReader();  bool[,] mass = new bool[15,4];  int i = 0;  while (reader.Read())  {  //1,4,15,21  if (reader.GetInt32(1) != 1 && reader.GetInt32(1) != 4 && reader.GetInt32(1) != 15 && reader.GetInt32(1) != 21)  {  mass[i, 0] = reader.GetBoolean(2);  mass[i, 1] = reader.GetBoolean(3);  mass[i, 2] = reader.GetBoolean(4);  mass[i, 3] = reader.GetBoolean(5);  i++;  }  }  reader.Close();  b.closeConnection();  checkBox1\_1.Checked = mass[0, 0]; checkBox1\_2.Checked = mass[0, 1]; checkBox1\_3.Checked = mass[0, 2]; checkBox1\_4.Checked = mass[0, 3];  checkBox2\_1.Checked = mass[1, 0]; checkBox2\_2.Checked = mass[1, 1]; checkBox2\_3.Checked = mass[1, 2]; checkBox2\_4.Checked = mass[1, 3];  checkBox3\_1.Checked = mass[2, 0]; checkBox3\_2.Checked = mass[2, 1]; checkBox3\_3.Checked = mass[2, 2]; checkBox3\_4.Checked = mass[2, 3];  checkBox4\_1.Checked = mass[3, 0]; checkBox4\_2.Checked = mass[3, 1]; checkBox4\_3.Checked = mass[3, 2]; checkBox4\_4.Checked = mass[3, 3];  checkBox5\_1.Checked = mass[4, 0]; checkBox5\_2.Checked = mass[4, 1]; checkBox5\_3.Checked = mass[4, 2]; checkBox5\_4.Checked = mass[4, 3];  checkBox6\_1.Checked = mass[5, 0]; checkBox6\_2.Checked = mass[5, 1]; checkBox6\_3.Checked = mass[5, 2]; checkBox6\_4.Checked = mass[5, 3];  checkBox7\_1.Checked = mass[6, 0]; checkBox7\_2.Checked = mass[6, 1]; checkBox7\_3.Checked = mass[6, 2]; checkBox7\_4.Checked = mass[6, 3];  checkBox8\_1.Checked = mass[7, 0]; checkBox8\_2.Checked = mass[7, 1]; checkBox8\_3.Checked = mass[7, 2]; checkBox8\_4.Checked = mass[7, 3];  checkBox9\_1.Checked = mass[8, 0]; checkBox9\_2.Checked = mass[8, 1]; checkBox9\_3.Checked = mass[8, 2]; checkBox9\_4.Checked = mass[8, 3];  checkBox10\_1.Checked = mass[9, 0]; checkBox10\_2.Checked = mass[9, 1]; checkBox10\_3.Checked = mass[9, 2]; checkBox10\_4.Checked = mass[9, 3];  checkBox11\_1.Checked = mass[10, 0]; checkBox11\_2.Checked = mass[10, 1]; checkBox11\_3.Checked = mass[10, 2]; checkBox11\_4.Checked = mass[10, 3];  checkBox12\_1.Checked = mass[11, 0]; checkBox12\_2.Checked = mass[11, 1]; checkBox12\_3.Checked = mass[11, 2]; checkBox12\_4.Checked = mass[11, 3];  checkBox13\_1.Checked = mass[12, 0]; checkBox13\_2.Checked = mass[12, 1]; checkBox13\_3.Checked = mass[12, 2]; checkBox13\_4.Checked = mass[12, 3];  checkBox14\_1.Checked = mass[13, 0]; checkBox14\_2.Checked = mass[13, 1]; checkBox14\_3.Checked = mass[13, 2]; checkBox14\_4.Checked = mass[13, 3];  checkBox15\_1.Checked = mass[14, 0]; checkBox15\_2.Checked = mass[14, 1]; checkBox15\_3.Checked = mass[14, 2]; checkBox15\_4.Checked = mass[14, 3];  }  // Нажатие на клавишу "Выбрать".  private void button2\_Click(object sender, EventArgs e)  {  b.openConnection();  int id = 0;  string qwertystring = $"select \* from Users where Логин = '{comboBox1.Text.Trim()}'";  OleDbCommand command = new OleDbCommand(qwertystring, b.getConnection());  OleDbDataReader reader = command.ExecuteReader();  while (reader.Read())  {  id = reader.GetInt32(0);  }  b.closeConnection();  reader.Close();  if (id < 0)  {  MessageBox.Show("Пользователь не выбран", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Warning);  }  else  {  buttonChange.Enabled = true;  buttonDel.Enabled = true;  button3.Enabled = true;  comboBox1.Enabled = false;  RefreshCheckBox(id);  }  }  }  } |

Таблица 9 - содержимое решения Setting

|  |
| --- |
| Содержимое модуля “Class1.cs” |
| namespace MyProj  {  public class Class1  {  static EventHandler SuppliersF()  {  return SuppliersF\_Click;  }  private static void SuppliersF\_Click(object sender, EventArgs e)  {  SuppliersForm form = new SuppliersForm();  form.Show();  }  }  } |
| Содержимое модуля “SuppliersForm.cs” |
| using database;  using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Data.OleDb;  using System.Drawing;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Windows.Forms;  namespace Suppliers  {    public partial class SuppliersForm : Form  {  // Перечисление для состояния строки.  enum RowState  {  Existed,  New,  Modifide,  ModifideNew,  Deleted  }  DataB b = new DataB();  // Переменная для работы с DataGridView.  int selectedRow;  bool W, E, D;  public SuppliersForm()  {  InitializeComponent();  StartPosition = FormStartPosition.CenterScreen;  RightsFunction();  if (W == false)  {  buttonNew.Enabled = false;  }  if (E == false)  {  buttonChange.Enabled = false;  }  if (D == false)  {  buttonDel.Enabled = false;  }  }  // Авторизация прав пользователя на меню Покупатель.  private void RightsFunction()  {  b.openConnection();  string qwertystring = $"select \* from Rights where ID\_Menu = 9";  OleDbCommand command = new OleDbCommand(qwertystring, b.getConnection());  OleDbDataReader reader = command.ExecuteReader();  while (reader.Read())  {  W = Convert.ToBoolean(reader.GetBoolean(2));  E = Convert.ToBoolean(reader.GetBoolean(3));  D = Convert.ToBoolean(reader.GetBoolean(4));  }  reader.Close();  b.closeConnection();  }  // Создание столбцов в dataGridView.  private void CreateColums()  {  dataGridView1.Columns.Add("ID", "ID");  dataGridView1.Columns.Add("Name", "Название");  dataGridView1.Columns.Add("banks\_id", "Банк\_ID");  dataGridView1.Columns.Add("rasch", "Расчётный счёт");  dataGridView1.Columns.Add("streets\_id", "Улица\_ID");  dataGridView1.Columns.Add("House", "Дом");  dataGridView1.Columns.Add("stroen", "Строение");  dataGridView1.Columns.Add("kvar", "Квартира");  dataGridView1.Columns.Add("tel", "Телефон");  dataGridView1.Columns.Add("inn", "ИНН");  // Статус строки.  dataGridView1.Columns.Add("IsNew", String.Empty);  dataGridView1.Columns[10].Visible = false;  }  // Отчистка таблицы "Запись".  private void ClearFields()  {  textBox1.Text = "";  textBox2.Text = "";  textBox3.Text = "";  textBox4.Text = "";  textBox5.Text = "";  textBox6.Text = "";  textBox7.Text = "";  textBox8.Text = "";  comboBox1.SelectedIndex = -1;  comboBox1.Items.Clear();  comboBox2.SelectedIndex = -1;  comboBox2.Items.Clear();  }  // Добавление Строки из базы данных в DataGridView.  private void ReadSingleRow(DataGridView dgw, IDataRecord record)  {  int srtoen;  try  {  srtoen = Convert.ToInt32(record.GetInt32(6));  }  catch (Exception ex) { srtoen = 0; }  dgw.Rows.Add(record.GetInt32(0), record.GetString(1).Trim(), record.GetInt32(2), record.GetString(3).Trim(), record.GetInt32(4), record.GetInt32(5), srtoen, record.GetInt32(7), record.GetString(8).Trim(), record.GetString(9).Trim(), RowState.ModifideNew);  }  // Поиск данных в базе данных.  private void RefreshDataGrid(DataGridView dgw)  {  dgw.Rows.Clear();  string qwertystring = $"select \* from Поставщик";  OleDbCommand command = new OleDbCommand(qwertystring, b.getConnection());  b.openConnection();  OleDbDataReader reader = command.ExecuteReader();  while (reader.Read())  {  ReadSingleRow(dgw, reader);  }  reader.Close();  b.closeConnection();  }  private void SuppliersForm\_Load(object sender, EventArgs e)  {  CreateColums();  RefreshDataGrid(dataGridView1);  }  // Запись данных в таблицу "Запись".  private void dataGridView1\_CellClick(object sender, DataGridViewCellEventArgs e)  {  comboBox1.Items.Clear();  comboBox2.Items.Clear();  selectedRow = e.RowIndex;  if (e.RowIndex >= 0)  {  DataGridViewRow row = dataGridView1.Rows[selectedRow];  textBox1.Text = row.Cells[0].Value.ToString().Trim();  textBox2.Text = row.Cells[1].Value.ToString().Trim();  textBox3.Text = row.Cells[3].Value.ToString().Trim();  textBox4.Text = row.Cells[4].Value.ToString().Trim();  textBox5.Text = row.Cells[7].Value.ToString().Trim();  textBox7.Text = row.Cells[8].Value.ToString().Trim();  textBox8.Text = row.Cells[9].Value.ToString().Trim();  if (row.Cells[6].Value.ToString().Trim() == null)  {  textBox6.Text = "0";  }  else  {  textBox6.Text = row.Cells[6].Value.ToString().Trim();  }  // Поиск Улицы по id и добавление вкладок.  var qwery1 = $"select \* from Улица";  var command = new OleDbCommand(qwery1, b.getConnection());  b.openConnection();  OleDbDataReader reader = command.ExecuteReader();  int ID\_spr = 0;  while (reader.Read())  {  comboBox2.Items.Add(reader["Наименование"].ToString());  ID\_spr = reader.GetInt32(0);  if (ID\_spr == Convert.ToInt32(row.Cells[4].Value.ToString()))  {  comboBox2.Text = reader.GetString(1);  }  }  reader.Close();  // Поиск Банка по id и добавление вкладок.  var qwery2 = $"select \* from Банк";  var command2 = new OleDbCommand(qwery2, b.getConnection());  OleDbDataReader reader2 = command2.ExecuteReader();  while (reader2.Read())  {  comboBox1.Items.Add(reader2["Наименование"].ToString());  ID\_spr = reader2.GetInt32(0);  if (ID\_spr == Convert.ToInt32(row.Cells[2].Value.ToString()))  {  comboBox1.Text = reader2.GetString(1);  }  }  reader2.Close();  b.closeConnection();  }  }  // Нажатие на клавишу "Обновить"  private void button2\_Click(object sender, EventArgs e)  {  DialogResult result = MessageBox.Show("Сохранить изменения?", "Сохрание", MessageBoxButtons.YesNo, MessageBoxIcon.Question);  switch (result)  {  case DialogResult.Yes:  {  Update();  RefreshDataGrid(dataGridView1);  ClearFields();  break;  }  case DialogResult.No:  {  RefreshDataGrid(dataGridView1);  ClearFields();  break;  }  } }  // Событие "Добавление записи".  private void buttonNew\_Click(object sender, EventArgs e)  {  AddSuppliers addfrm = new AddSuppliers();  addfrm.Show();  }  // Метод для поиска данных в БД.  private void Search(DataGridView dgw)  {  dgw.Rows.Clear();  string serchString = $"select \* from Поставщик where Название like '%" + textBox\_search.Text + "%'" + " or ID like '%" + textBox\_search.Text+ "%'" + " or Банк\_ID like '%" + textBox\_search.Text + "%'" + " or РасчётныйСчёт like '%" + textBox\_search.Text + "%'" + " or Улица\_ID like '%" + textBox\_search.Text + "%'" + " or Дом like '%" + textBox\_search.Text + "%'" + " or Строение like '%" + textBox\_search.Text + "%'" + " or Квартира like '%" + textBox\_search.Text + "%'" + " or Телефон like '%" + textBox\_search.Text + "%'" + " or ИНН like '%" + textBox\_search.Text + "%'";  OleDbCommand com = new OleDbCommand(serchString, b.getConnection());  b.openConnection();  OleDbDataReader read = com.ExecuteReader();  while (read.Read())  {  ReadSingleRow(dgw, read);  }  read.Close();  b.closeConnection();  }  // Событие когда поиск активен.  private void TextGotFocus(object sender, EventArgs e)  {  // Удаление подсказки.  if (textBox\_search.Text == "Поиск...")  {  textBox\_search.Text = "";  textBox\_search.ForeColor = Color.Black;  }  }  // Cобытие когда поиск неактивен.  private void TextLostFocus(object sender, EventArgs e)  {  // Возвращаем подсказку при неактивном состояни, если поле пустое.  if (string.IsNullOrWhiteSpace(textBox\_search.Text))  {  textBox\_search.Text = "Поиск...";  textBox\_search.ForeColor = Color.Gray;  RefreshDataGrid(dataGridView1);  }  }  // Событие при изменении поиска.  private void textBox\_search\_TextChanged(object sender, EventArgs e)  {  Search(dataGridView1);  }  // Метод обнавляет, удаляет или ничего не делает с данными в бд в соответствии со статусом каждой строки в DataGridView  private void Update()  {  b.openConnection();  for (int i = 0; i < dataGridView1.Rows.Count; i++)  {  var rowState = (RowState)dataGridView1.Rows[i].Cells[10].Value;  if (rowState == RowState.Existed) continue;  if (rowState == RowState.Deleted)  {  var id = Convert.ToInt32(dataGridView1.Rows[i].Cells[0].Value);  var deleteQwery = $"delete from Поставщик where ID = {id}";  var command = new OleDbCommand(deleteQwery, b.getConnection());  command.ExecuteNonQuery();  }  if (rowState == RowState.Modifide)  {  var id = dataGridView1.Rows[i].Cells[0].Value.ToString().Trim();  var name = dataGridView1.Rows[i].Cells[1].Value.ToString().Trim();  var banks\_id = dataGridView1.Rows[i].Cells[2].Value.ToString().Trim();  var rasch = dataGridView1.Rows[i].Cells[3].Value.ToString().Trim();  var streets\_id = dataGridView1.Rows[i].Cells[4].Value.ToString().Trim();  var house = dataGridView1.Rows[i].Cells[5].Value.ToString().Trim();  var stroen = Convert.ToInt32(dataGridView1.Rows[i].Cells[6].Value.ToString().Trim());  var kvar = dataGridView1.Rows[i].Cells[7].Value.ToString().Trim();  var tel = dataGridView1.Rows[i].Cells[8].Value.ToString().Trim();  var inn = dataGridView1.Rows[i].Cells[9].Value.ToString().Trim();  string changeQwery;  if (stroen != 0)  {  changeQwery = $"update Поставщик set Название = '{name}', Банк\_ID = {banks\_id}, РасчётныйСчёт = '{rasch}', Улица\_ID ={streets\_id}, Дом = {house}, Строение= {stroen}, Квартира= {kvar}, Телефон= '{tel}', ИНН= '{inn}' where ID ={id}";  }  else  {  changeQwery = $"update Поставщик set Название = '{name}', Банк\_ID = {banks\_id}, РасчётныйСчёт = '{rasch}', Улица\_ID ={streets\_id}, Дом = {house}, Строение= NULL, Квартира= {kvar}, Телефон= '{tel}', ИНН= '{inn}' where ID ={id}";  }  var command = new OleDbCommand(changeQwery, b.getConnection());  command.ExecuteNonQuery();  }  }  b.closeConnection();  }  // Метод делает строку невидимой и присваивает ей статус "Удален"  private void deleteRow()  {  int index = dataGridView1.CurrentCell.RowIndex;  dataGridView1.Rows[index].Visible = false;  if (Convert.ToInt32(dataGridView1.Rows[index].Cells[0].Value) >= 0)  {  dataGridView1.Rows[index].Cells[10].Value = RowState.Deleted;  }  }  // Событие "Удаление записи".  private void buttonDel\_Click(object sender, EventArgs e)  {  deleteRow();  ClearFields();  }  // Событие "Сохраниение".  private void buttonSave\_Click(object sender, EventArgs e)  {  Update();  }  private void Change()  {  var selectedRowindex = dataGridView1.CurrentCell.RowIndex;  var id = textBox1.Text;  var name = textBox2.Text;  var banks\_id = 0;  var rasch = textBox3.Text;  var streets\_id = 0;  int house;  int kvar;  var tel = textBox7.Text;  var inn = textBox8.Text;  bool isNumber1 = int.TryParse(textBox4.Text, out house);  bool isNumber3 = int.TryParse(textBox6.Text, out kvar);  // Обратный поиск ID по Наименованием и названиям виду упаковки, категории, производителю.  b.openConnection();  // Поиск Улица\_ID.  var qwery1 = $"select ID from Улица where Наименование = '{comboBox2.Text}'";  var command = new OleDbCommand(qwery1, b.getConnection());  OleDbDataReader reader = command.ExecuteReader();  while (reader.Read())  {  streets\_id = reader.GetInt32(0);  }  reader.Close();  // Поиск Банк\_ID.  var qwery2 = $"select ID from Банк where Наименование = '{comboBox1.Text}'";  var command2 = new OleDbCommand(qwery2, b.getConnection());  OleDbDataReader reader2 = command.ExecuteReader();  while (reader2.Read())  {  banks\_id = reader2.GetInt32(0);  }  reader2.Close();  // Проверка ввода.  if (isNumber1 == true && isNumber3 == true && rasch.Length <= 20 && tel.Length <= 12 && inn.Length <= 10 && name != "" && Convert.ToInt32(house) > 0 && Convert.ToInt32(kvar) > 0)  {  int stroen\_int;  bool isNumber2 = int.TryParse(textBox5.Text, out stroen\_int);  if (isNumber2 == true)  {  dataGridView1.Rows[selectedRowindex].SetValues(id, name, banks\_id, rasch, streets\_id, house, stroen\_int, kvar, tel, inn);  }  else  {  dataGridView1.Rows[selectedRowindex].SetValues(id, name, banks\_id, rasch, streets\_id, house, 0, kvar, tel, inn);  }  // установка статуса: Измененная строка.  dataGridView1.Rows[selectedRowindex].Cells[10].Value = RowState.Modifide;  }  else  {  MessageBox.Show("Неправильный ввод", "Создание записи", MessageBoxButtons.OK, MessageBoxIcon.Warning);  }  b.closeConnection();  }  private void buttonChange\_Click(object sender, EventArgs e)  {  Change();  ClearFields();  }  // Нажатие на клавишу "Отчистка"  private void button1\_Click(object sender, EventArgs e)  {  ClearFields();  }  }  } |
| Содержимое модуля “AddSuppliers.cs” |
| namespace Suppliers  {  public partial class AddSuppliers : Form  {  DataB database = new DataB();  public AddSuppliers()  {  InitializeComponent();  StartPosition = FormStartPosition.CenterScreen;  database.openConnection();  // Поиск Улицы из бд.  var qwery1 = $"select \* from Улица";  var command = new OleDbCommand(qwery1, database.getConnection());  OleDbDataReader reader = command.ExecuteReader();  while (reader.Read())  {  comboBox2.Items.Add(reader["Наименование"].ToString());  }  reader.Close();  // Поиск Банка из бд.  var qwery2 = $"select \* from Банк";  var command2 = new OleDbCommand(qwery2, database.getConnection());  OleDbDataReader reader2 = command2.ExecuteReader();  while (reader2.Read())  {  comboBox1.Items.Add(reader2["Наименование"].ToString());  }  reader2.Close();  database.closeConnection();  }  private void button1\_Click(object sender, EventArgs e)  {  database.openConnection();  var name = textBox1.Text;  int banks\_id = 0;  var rasch = textBox2.Text;  int streets\_id = 0;  int house;  int stroen;  int kvar;  var tel = textBox6.Text;  var inn = textBox7.Text;  // Поиск Улица\_ID.  var qwery1 = $"select ID from Улица where Наименование = '{comboBox2.Text}'";  var command = new OleDbCommand(qwery1, database.getConnection());  OleDbDataReader reader = command.ExecuteReader();  while (reader.Read())  {  streets\_id = reader.GetInt32(0);  }  reader.Close();  // Поиск Банк\_ID.  var qwery2 = $"select ID from Банк where Наименование = '{comboBox1.Text}'";  var command2 = new OleDbCommand(qwery2, database.getConnection());  OleDbDataReader reader2 = command2.ExecuteReader();  while (reader2.Read())  {  banks\_id = reader2.GetInt32(0);  }  reader2.Close();  // Проверка ввода.  bool isNumber1 = int.TryParse(textBox3.Text, out house);  bool isNumber2 = int.TryParse(textBox4.Text, out stroen);  bool isNumber3 = int.TryParse(textBox5.Text, out kvar);  string addQwery;  // Проверка на не пустоту строк и запрос на добавление новой строки в бд.  if (isNumber1 == true && isNumber3 == true && rasch.Length <=20 && tel.Length <= 12 && inn.Length <= 20 && house > 0 && kvar > 0 && name != "" && streets\_id > 0 && banks\_id > 0)  {  if (isNumber2 == false)  {  addQwery = $"insert into Поставщик (Название, Банк\_ID, РасчётныйСчёт, Улица\_ID, Дом, Строение, Квартира, Телефон, ИНН) values ('{name}', {banks\_id},'{rasch}',{streets\_id}, {house}, NULL, '{kvar}', '{tel}', '{inn}' )";  }  else  {  addQwery = $"insert into Поставщик (Название, Банк\_ID, РасчётныйСчёт, Улица\_ID, Дом, Строение, Квартира,Телефон, ИНН) values ('{name}', {banks\_id}, '{rasch}', {streets\_id}, {house}, {stroen}, '{kvar}', '{tel}', '{inn}' )";  }  var command4 = new OleDbCommand(addQwery, database.getConnection());  command4.ExecuteNonQuery();  MessageBox.Show("Запись успешно создана!", "Создание записи", MessageBoxButtons.OK, MessageBoxIcon.Information);  textBox1.Text = "";  textBox2.Text = "";  textBox3.Text = "";  textBox4.Text = "";  textBox5.Text = "";  textBox6.Text = "";  textBox7.Text = "";  }  else  {  MessageBox.Show("Неправильный ввод", "Создание записи", MessageBoxButtons.OK, MessageBoxIcon.Warning);  }  database.closeConnection();  }  }  } |

Таблица 10 - содержимое решения Suppliers

|  |
| --- |
| Содержимое модуля “Form1.cs” |
| namespace KurovayaBD  {  public partial class Form1 : Form  {  // Таймер для постоянной проверки состояния формы.  System.Windows.Forms.Timer formTimer = new System.Windows.Forms.Timer();  DataB dataBase = new DataB();  public Form1()  {  InitializeComponent();  StartPosition = FormStartPosition.CenterScreen;  ActiveControl = UserName;  }  private void CloseButton\_Click(object sender, EventArgs e)  {  Close();  }  private void CloseButton\_MouseEnter(object sender, EventArgs e)  {  CloseButton.ForeColor = Color.Blue;  }  private void CloseButton\_MouseLeave(object sender, EventArgs e)  {  CloseButton.ForeColor = Color.Black;  }  private void Form1\_Load(object sender, EventArgs e)  {  formTimer.Interval = 500;  formTimer.Start();  formTimer.Tick += new EventHandler(FormTimer\_Tick);  }  private void FormTimer\_Tick(object sender, EventArgs e)  {  CapsLockFlagLabel.Text = (Console.CapsLock ? "Клавиша CapsLock нажата" : "");  if (InputLanguage.CurrentInputLanguage.LayoutName == "США")  CurrentLanguageLabel.Text = "Язык ввода Английский";  else if (InputLanguage.CurrentInputLanguage.LayoutName == "Русская")  CurrentLanguageLabel.Text = "Язык ввода Русский";  }  private void EnterButton\_Click(object sender, EventArgs e)  {  // Удаляем все права из временной таблицы Rights  dataBase.openConnection();  string qwery = $"DELETE FROM Rights";  OleDbCommand command5 = new OleDbCommand(qwery, dataBase.getConnection());  command5.ExecuteNonQuery();    // Считывание введенного логина и пароля.  var loginUser = UserName.Text;  var passUser = md5.hashPassword(Password.Text);  OleDbDataAdapter adapter = new OleDbDataAdapter();  DataTable table = new DataTable();  // Создание строки запроса.  string qwerystring = $"select Id, Логин, Пароль from Users where Логин = '{loginUser}' and Пароль = '{passUser}'";  OleDbCommand command = new OleDbCommand(qwerystring, dataBase.getConnection());  adapter.SelectCommand = command;  adapter.Fill(table);  if (table.Rows.Count == 1)  {  // Считываем id пользователя.  int user\_id = 0;  OleDbDataReader reader = command.ExecuteReader();  while (reader.Read())  {  user\_id = reader.GetInt32(0);  }  reader.Close();  dataBase.closeConnection();  Menu1 menu1 = new Menu1(this, user\_id);  menu1.Show();  Hide();  }  else  {  MessageBox.Show("Неверный логин или пароль", "Ошибка ввода", MessageBoxButtons.OK, MessageBoxIcon.Warning);  Password.Text = "";  }  }  private void CancelButton\_Click(object sender, EventArgs e)  {  Close();  }  }  } |
| Содержимое модуля “Menu1.cs” |
| using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Diagnostics.Contracts;  using System.Drawing;  using System.Linq;  using System.Reflection;  using System.Text;  using System.Threading.Tasks;  using System.Security.Cryptography;  using System.Text.Json.Serialization;  using System.Windows.Forms;  using static System.Windows.Forms.VisualStyles.VisualStyleElement;  using database;  using System.Data.OleDb;  namespace KurovayaBD  {  public partial class Menu1 : Form  {  Form1 form1 = new Form1();  DataB dataBase = new DataB();  public Menu1(Form1 form, int ID)  {  InitializeComponent();  // Переменные для хранения информации о вкладках для каждого уровня иерархии.  ToolStripMenuItem menu = new ToolStripMenuItem();  ToolStripMenuItem soMenu = new ToolStripMenuItem();  form1 = form;  DataTable MenuBD = new DataTable();  // добавление данных в основное меню  UPD();  // Создаём запрос на права пользователя.  dataBase.openConnection();  string qwerystring3 = $"select \* from AccessRights where ID\_User ={ID} order by ID\_Menu";  OleDbCommand command1 = new OleDbCommand(qwerystring3, dataBase.getConnection());  OleDbDataReader reader = command1.ExecuteReader();  // Коллекции типа ключ-значения, для хранения прав пользователя.  List<int> MENU = new List<int>();  Dictionary<int, bool> R = new Dictionary<int, bool>();  Dictionary<int, bool> W = new Dictionary<int, bool>();  Dictionary<int, bool> E = new Dictionary<int, bool>();  Dictionary<int, bool> D = new Dictionary<int, bool>();  while (reader.Read())  {  MENU.Add(reader.GetInt32(1));  R.Add(reader.GetInt32(1), reader.GetBoolean(2));  W.Add(reader.GetInt32(1), reader.GetBoolean(3));  E.Add(reader.GetInt32(1), reader.GetBoolean(4));  D.Add(reader.GetInt32(1), reader.GetBoolean(5));  }  reader.Close();  // Добавление прав конкретного пользователя в таблицу Rights.  for (int i = 0; i < MENU.Count; i++)  {  var addQwery = $"insert into Rights values ({MENU[i]},{R[MENU[i]]},{W[MENU[i]]}, {E[MENU[i]]}, {D[MENU[i]]}, {ID})";  var command4 = new OleDbCommand(addQwery, dataBase.getConnection());  command4.ExecuteNonQuery();  }  // Создание запроса на получение данных о меню.  string qwerystring2 = $"select \* from Menu order by ID, Порядок";  OleDbCommand command = new OleDbCommand(qwerystring2, dataBase.getConnection());  reader = command.ExecuteReader();  // Данные из таблицы Menu.  int ID\_menu;  int ID\_parent;  string name;  string name\_dll = "";  string nameF = "";  // Количество вкладок на панеле.  int countMenus;  // Количество подвкладок на панеле.  int countSoMenus;  while (reader.Read())  {  ID\_menu = reader.GetInt32(0);  ID\_parent = reader.GetInt32(1);  name = reader.GetString(2).Trim();  try  {  name\_dll = reader.GetString(3).Trim() + ".dll";  nameF = reader.GetString(4).Trim();  }  catch (Exception) { name\_dll = ""; nameF = ""; }  // Добавление вкладки в панель если меню не имеет предков.  if (ID\_parent == 0)  {  menu = new ToolStripMenuItem();  menu.Text += name;  if (R[ID\_menu] == false)  {  menu.Enabled = false;  }  // Добавляем вкладку на панель.  TopMenu.Items.Add(menu);  countMenus = TopMenu.Items.Count - 1;  if (name\_dll != "")  {  try  {  // Загружаем dll.  Assembly asm = Assembly.LoadFrom(name\_dll);  // Получаем класс.  Type? t = asm.GetType("MyProj.Class1");  // Получаем метод обработки запросов.  MethodInfo? getMethod = t.GetMethod(nameF, BindingFlags.NonPublic | BindingFlags.Static);  // Вызываем метод из dll и передаём данные о правах пользователя.  EventHandler? result = getMethod?.Invoke(null, null) as EventHandler;  TopMenu.Items[countMenus].Click += result;  }  catch (Exception) { }  }  }  // Если меню - это подменю, то создаён подменю.  if (ID\_parent > 0)  {  soMenu = new ToolStripMenuItem();  soMenu.Text += name;  if (R[ID\_menu] == false)  {  soMenu.Enabled = false;  }  menu.DropDownItems.Add(soMenu);  countSoMenus = menu.DropDownItems.Count - 1;  try  {  Assembly asm = Assembly.LoadFrom(name\_dll);  Type? t = asm.GetType("MyProj.Class1");  MethodInfo? getMethod = t.GetMethod(nameF, BindingFlags.Static | BindingFlags.NonPublic);  EventHandler? result = getMethod?.Invoke(null, null) as EventHandler;  menu.DropDownItems[countSoMenus].Click += result;  }  catch (Exception) { }  }  }  reader.Close();  dataBase.closeConnection();  }  private void Menu1\_FormClosing(object sender, FormClosingEventArgs e)  {  form1.Close();  }  // Закрытие программы  private void button1\_Click(object sender, EventArgs e)  {  DialogResult result = MessageBox.Show("Вы действительно хотите выйти?", "Выход", MessageBoxButtons.YesNo, MessageBoxIcon.Question);  switch (result)  {  case DialogResult.Yes:  {  form1.Close();  break;  }  case DialogResult.No:  break;  }  }  // Кнопка для обновления основного меню.  private void button2\_Click(object sender, EventArgs e)  {  UPD();  }  // Обновление данных в основном меню.  private void UPD()  {  dataBase.openConnection();  // Кол-во поставщиков.  string qwerystring = $"select count(\*) from Поставщик";  OleDbCommand command = new OleDbCommand(qwerystring, dataBase.getConnection());  OleDbDataReader reader = command.ExecuteReader();  while (reader.Read())  {  label4.Text = reader.GetInt32(0).ToString();  }  reader.Close();  // Кол-во покупателей.  string qwerystring2 = $"select count(\*) from Покупатель";  OleDbCommand command2 = new OleDbCommand(qwerystring2, dataBase.getConnection());  OleDbDataReader reader2 = command2.ExecuteReader();  while (reader2.Read())  {  label5.Text = reader2.GetInt32(0).ToString();  }  reader2.Close();  // Кол-во активных заказов.  string qwerystring3 = $"select \* from `Счёт-фактуры` order by Заказ\_ID";  OleDbCommand command3 = new OleDbCommand(qwerystring3, dataBase.getConnection());  OleDbDataReader reader3 = command3.ExecuteReader();  int count = 0;  int last\_id = 0;  while (reader3.Read())  {  if (last\_id != reader3.GetInt32(3) && reader3.GetString(4).Trim() == "Активен")  {  count++;  }  last\_id = reader3.GetInt32(3);  }  reader2.Close();  label6.Text = count.ToString();  dataBase.closeConnection();  }  }  } |

Таблица 11 - содержимое решения Tech2

|  |
| --- |
| Содержимое модуля “Class1.cs” |
| namespace MyProj  {  public class Class1  {  static EventHandler WaybillF()  {  return WaybillF\_Click;  }  private static void WaybillF\_Click(object sender, EventArgs e)  {  WaybillForm form = new WaybillForm();  form.Show();  }  }  } |
| Содержимое модуля “WayBillForm.cs” |
| using database;  namespace Waybill  {  public partial class WaybillForm : Form  {  bool W;  DataB b = new DataB();  public WaybillForm()  {  InitializeComponent();  StartPosition = FormStartPosition.CenterScreen;  RightsFunction();  AddItems();  if (W == false)  {  button1.Enabled = false;  }  }  // Авторизация прав пользователя на меню Накладная.  private void RightsFunction()  {  string qwertystring = $"select \* from Rights where ID\_Menu = 7";  OleDbCommand command = new OleDbCommand(qwertystring, b.getConnection());  b.openConnection();  OleDbDataReader reader = command.ExecuteReader();  while (reader.Read())  {  W = Convert.ToBoolean(reader.GetBoolean(2));  }  reader.Close();  b.closeConnection();  }  private void AddItems()  {  // Поиск Поставщиков и добавление вкладок.  var qwery1 = $"select \* from Поставщик";  var command = new OleDbCommand(qwery1, b.getConnection());  b.openConnection();  OleDbDataReader reader = command.ExecuteReader();  while (reader.Read())  {  comboBox1.Items.Add(reader["Название"].ToString().Trim());  }  reader.Close();  b.closeConnection();  }  private void button1\_Click(object sender, EventArgs e)  {  if(comboBox1.Text.Length > 0)  {  b.openConnection();  int post\_id = 0;  int postavstchik\_id = 0;  // Поиск Поставщик\_ID.  var qwery1 = $"select ID from Поставщик where Название = '{comboBox1.Text}'";  var command = new OleDbCommand(qwery1, b.getConnection());  OleDbDataReader reader = command.ExecuteReader();  while (reader.Read())  {  postavstchik\_id = reader.GetInt32(0);  }  reader.Close();  string addpost = $"insert into Поставка (Поставщик\_ID, Дата ) values ({postavstchik\_id}, '{dateTimePicker1.Value}')";  var command1 = new OleDbCommand(addpost, b.getConnection());  command1.ExecuteNonQuery();  // Получение ID созданной поставки.  string qwerty = $"SELECT \* FROM Поставка WHERE Поставщик\_ID = {postavstchik\_id} AND ID = (SELECT MAX(ID) FROM Поставка)";  OleDbCommand command2 = new OleDbCommand(qwerty, b.getConnection());  OleDbDataReader reader2 = command2.ExecuteReader();  while (reader2.Read())  {  post\_id = Convert.ToInt32(reader2.GetInt32(0));  }  reader2.Close();  b.closeConnection();  Close();  NakladForm frm = new NakladForm(post\_id);  frm.Show();  }  else  {  MessageBox.Show("Не выбран поставщик", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Warning);  }  }  }  } |
| Содержимое модуля “NakladForm.cs” |
| namespace Waybill  {  public partial class NakladForm : Form  {  // Перечисление для состояния строки.  enum RowState  {  Existed,  New,  Modifide,  ModifideNew,  Deleted  }  DataB b = new DataB();  // Переменная для работы с DataGridView.  int selectedRow;  public bool W, E, D;  public int id\_postavki;  public NakladForm(int id)  {  InitializeComponent();  StartPosition = FormStartPosition.CenterScreen;  id\_postavki = id;  RightsFunction();  if (W == false)  {  buttonNew.Enabled = false;  }  if (E == false)  {  buttonChange.Enabled = false;  }  if (D == false)  {  buttonDel.Enabled = false;  }  }  // Авторизация прав пользователя на меню Производитель.  private void RightsFunction()  {  string qwertystring = $"select \* from Rights where ID\_Menu = 6";  OleDbCommand command = new OleDbCommand(qwertystring, b.getConnection());  b.openConnection();  OleDbDataReader reader = command.ExecuteReader();  while (reader.Read())  {  W = Convert.ToBoolean(reader.GetBoolean(2));  E = Convert.ToBoolean(reader.GetBoolean(3));  D = Convert.ToBoolean(reader.GetBoolean(4));  }  reader.Close();  b.closeConnection();  }  private void CreateColums()  {  dataGridView1.Columns.Add("ID", "ID");  dataGridView1.Columns.Add("lekar", "Лекарство");  dataGridView1.Columns.Add("kol", "Количество");  dataGridView1.Columns.Add("price", "Цена");  dataGridView1.Columns.Add("post\_id", "Поставка\_ID");  // Статус строки.  dataGridView1.Columns.Add("IsNew", String.Empty);  dataGridView1.Columns[0].Visible = false;  dataGridView1.Columns[4].Visible = false;  dataGridView1.Columns[5].Visible = false;  }  // Отчистка таблицы "Запись".  private void ClearFields()  {  textBox1.Text = "";  numericUpDown1.Value = 1;  textBox2.Text = "";  }  // Добавление Строки из базы данных в DataGridView.  private void ReadSingleRow(DataGridView dgw, IDataRecord record)  {  dgw.Rows.Add(record.GetInt32(0), record.GetInt32(1), record.GetInt32(2), record.GetDecimal(3), record.GetInt32(4), RowState.ModifideNew);  }  // Поиск данных в базе данных.  private void RefreshDataGrid(DataGridView dgw)  {  dgw.Rows.Clear();  string qwertystring = $"select \* from Накладная where Поставка\_ID = {id\_postavki}";  OleDbCommand command = new OleDbCommand(qwertystring, b.getConnection());  b.openConnection();  OleDbDataReader reader = command.ExecuteReader();  while (reader.Read())  {  ReadSingleRow(dgw, reader);  }  reader.Close();  b.closeConnection();  }  // Нажатие на клавишу "Обновить"  private void button2\_Click(object sender, EventArgs e)  {  DialogResult result = MessageBox.Show("Сохранить изменения?", "Сохрание", MessageBoxButtons.YesNo, MessageBoxIcon.Question);  switch (result)  {  case DialogResult.Yes:  {  Update();  RefreshDataGrid(dataGridView1);  ClearFields();  break;  }  case DialogResult.No:  {  RefreshDataGrid(dataGridView1);  ClearFields();  break;  }  } }  // Событие "Добавление записи".  private void buttonNew\_Click(object sender, EventArgs e)  {  RefreshDataGrid(dataGridView1);  AddLekNak addfrm = new AddLekNak(id\_postavki, dataGridView1);  addfrm.Show();  }  private void NakladForm\_Load\_1(object sender, EventArgs e)  {  CreateColums();  }  // Запись данных в таблицу "Запись".  private void dataGridView1\_CellClick(object sender, DataGridViewCellEventArgs e)  {  selectedRow = e.RowIndex;  if (e.RowIndex >= 0)  {  b.openConnection();  DataGridViewRow row = dataGridView1.Rows[selectedRow];  numericUpDown1.Value = Convert.ToInt32(row.Cells[2].Value.ToString().Trim());  textBox1.Text = row.Cells[3].Value.ToString().Trim();  var id = row.Cells[1].Value;  // Поиск Лекарства по id для получения названия и количества.  var qwery1 = $"select \* from Лекарство where ID = {id}";  var command = new OleDbCommand(qwery1, b.getConnection());  OleDbDataReader reader = command.ExecuteReader();  while (reader.Read())  {  textBox2.Text = reader["Название"].ToString();  }  reader.Close();  b.closeConnection();  }  }  // Метод обнавляет, удаляет или ничего не делает с данными в бд в соответствии со статусом каждой строки в DataGridView.  private void Update()  {  b.openConnection();  for (int i = 0; i < dataGridView1.Rows.Count; i++)  {  var rowState = (RowState)dataGridView1.Rows[i].Cells[5].Value;  if (rowState == RowState.Existed) continue;  if (rowState == RowState.Deleted)  {  var id = dataGridView1.Rows[i].Cells[0].Value.ToString().Trim();  var deleteQwery = $"delete from Накладная where ID = {id}";  var command = new OleDbCommand(deleteQwery, b.getConnection());  command.ExecuteNonQuery();  }  if (rowState == RowState.Modifide)  {  var id = dataGridView1.Rows[i].Cells[0].Value.ToString().Trim();  var kol = dataGridView1.Rows[i].Cells[2].Value.ToString().Trim();  var price = dataGridView1.Rows[i].Cells[3].Value.ToString().Trim();  price = price.Replace(",", ".");  // Обновление данных  string changeQwery = $"update Накладная set Количество = {kol}, Цена = {price} where ID ={id}";  var command2 = new OleDbCommand(changeQwery, b.getConnection());  command2.ExecuteNonQuery();  }  }  b.closeConnection();  }  // Метод делает строку невидимой и присваивает ей статус "Удален"  private void deleteRow()  {  int index = dataGridView1.CurrentCell.RowIndex;  dataGridView1.Rows[index].Visible = false;  if (Convert.ToInt32(dataGridView1.Rows[index].Cells[0].Value) >= 0)  {  dataGridView1.Rows[index].Cells[5].Value = RowState.Deleted;  }  }  // Событие "Удаление записи".  private void buttonDel\_Click(object sender, EventArgs e)  {  deleteRow();  ClearFields();  }  // Событие "Отчистка" .  private void button1\_Click(object sender, EventArgs e)  {  ClearFields();  }  private void buttonChange\_Click(object sender, EventArgs e)  {  Change();  ClearFields();  }  private void button3\_Click(object sender, EventArgs e)  {  RefreshDataGrid(dataGridView1);  MessageBox.Show("Накладная заполнена! Теперь вы можете загрузить Excel-файл.", "Заказ", MessageBoxButtons.OK, MessageBoxIcon.Information);  button4.Enabled = true;    }  // Экспорт файла  private void button4\_Click(object sender, EventArgs e)  {  // Создание объекта для запуска Excel.  Microsoft.Office.Interop.Excel.Application ExcelApp = new Microsoft.Office.Interop.Excel.Application();  // Создаём рабочую книгу.  ExcelApp.Application.Workbooks.Add(Type.Missing);  ExcelApp.Cells[1, 1] = "ID накладной";  ExcelApp.Cells[2, 1] = "Поставщик";  ExcelApp.Cells[3, 1] = "Дата";  ExcelApp.Cells[5, 1] = "Лекарство";  ExcelApp.Cells[5, 2] = "Количество";  ExcelApp.Cells[5, 3] = "Цена";  ExcelApp.Columns.ColumnWidth = 15;  int id\_Postawschik = 0;  string qwertystring = $"select \* from Поставка where ID = {id\_postavki}";  OleDbCommand command = new OleDbCommand(qwertystring, b.getConnection());  b.openConnection();  OleDbDataReader reader = command.ExecuteReader();  while (reader.Read())  {  ExcelApp.Cells[1, 2] = reader.GetInt32(0).ToString();  id\_Postawschik = reader.GetInt32(1);  ExcelApp.Cells[3, 2] = reader.GetDateTime(2).ToString();  }  reader.Close();  // Название поставщика по ID  string qwertystring2 = $"select Название from Поставщик where ID = {id\_Postawschik}";  OleDbCommand command2 = new OleDbCommand(qwertystring2, b.getConnection());  OleDbDataReader reader2 = command2.ExecuteReader();  while (reader2.Read())  {  ExcelApp.Cells[2, 2] = reader2.GetString(0).ToString();  }  reader2.Close();  b.closeConnection();  for (int i = 0; i < dataGridView1.RowCount; i++)  {  // Название Лекарства по ID  string qwertystring3 = $"select Название from Лекарство where ID = {dataGridView1[1, i].Value}";  b.openConnection();  OleDbCommand command3 = new OleDbCommand(qwertystring3, b.getConnection());  OleDbDataReader reader3 = command3.ExecuteReader();  while (reader3.Read())  {  ExcelApp.Cells[i + 6, 1] = reader3.GetString(0).ToString();  }  reader3.Close();  b.closeConnection();    ExcelApp.Cells[i + 6, 2] = (dataGridView1[2, i].Value).ToString();  ExcelApp.Cells[i + 6, 3] = (dataGridView1[3, i].Value).ToString();  }  ExcelApp.Visible = true;  b.closeConnection();  MessageBox.Show("Таблица успешно создана", "Создание таблицы", MessageBoxButtons.OK, MessageBoxIcon.Information);  }  // Событие "Сохраниение".  private void buttonSave\_Click(object sender, EventArgs e)  {  Update();  }  private void Change()  {  int selectedRowindex = dataGridView1.CurrentCell.RowIndex;  int kol = Convert.ToInt32(numericUpDown1.Value);  decimal price = Convert.ToDecimal(textBox1.Text);  // Проверка ввода.  if (price>=0)  {  dataGridView1.Rows[selectedRowindex].Cells[2].Value = kol;  dataGridView1.Rows[selectedRowindex].Cells[3].Value = price;  dataGridView1.Rows[selectedRowindex].Cells[5].Value = RowState.Modifide;  }  else  {  MessageBox.Show("Неправильный ввод", "Создание записи", MessageBoxButtons.OK, MessageBoxIcon.Warning);  }  }  }  } |
| Содержимое модуля “AddLekNak.cs” |
| namespace Waybill  {  public partial class AddLekNak : Form  {  DataB database = new DataB();  public int id\_Post;  public AddLekNak(int id\_post, DataGridView dgw)  {  id\_Post = id\_post;  InitializeComponent();  StartPosition = FormStartPosition.CenterScreen;  AddItems(dgw);  }  // Добавление лекарств в комбобокс.  private void AddItems(DataGridView dgw)  {  List<int> Name\_in\_dgw = new List<int>();  for (int i = 0; i < dgw.Rows.Count; i++)  {  Name\_in\_dgw.Add(Convert.ToInt32(dgw.Rows[i].Cells[1].Value));  }  database.openConnection();  // Поиск Лекарств из бд.  var qwery1 = $"select \* from Лекарство";  var command = new OleDbCommand(qwery1, database.getConnection());  OleDbDataReader reader = command.ExecuteReader();  while (reader.Read())  {  // Если Лекарство уже есть в Заказе, то не добавляем его в combobox.  if (Name\_in\_dgw.Contains(Convert.ToInt32(reader.GetInt32(0))))  continue;  else  comboBox1.Items.Add(reader["Название"].ToString().Trim());  }  reader.Close();  database.closeConnection();  }  private void button1\_Click(object sender, EventArgs e)  {  database.openConnection();  var kol = numericUpDown1.Value;  int lek\_id = 0;  decimal price;  bool isNumber1 = decimal.TryParse(textBox1.Text, out price);  // Поиск Лекарство\_ID.  var qwery1 = $"select \* from Лекарство where Название = '{comboBox1.Text}'";  var command = new OleDbCommand(qwery1, database.getConnection());  OleDbDataReader reader = command.ExecuteReader();  while (reader.Read())  {  lek\_id = reader.GetInt32(0);  }  reader.Close();  string addQwery;  // Проверка на не пустоту строк и запрос на добавление новой строки в бд.  if (lek\_id >= 0)  {  if (isNumber1 == true)  {  addQwery = $"insert into Накладная (Лекарство\_ID, Количество, Цена, Поставка\_ID) values ({lek\_id}, {kol}, {price}, {id\_Post})";  var command4 = new OleDbCommand(addQwery, database.getConnection());  command4.ExecuteNonQuery();  MessageBox.Show("Запись успешно создана!", "Создание записи", MessageBoxButtons.OK, MessageBoxIcon.Information);  comboBox1.Text = "";  numericUpDown1.Value = 1;  Close();  }  else  {  MessageBox.Show("Неверный ввод цены!", "Создание записи", MessageBoxButtons.OK, MessageBoxIcon.Warning);  }  }  else  {  MessageBox.Show("Не выбран пункт!", "Создание записи", MessageBoxButtons.OK, MessageBoxIcon.Warning);  }  database.closeConnection();  }  }  } |

Таблица 12 - содержимое решения WayBill