Государственное бюджетное профессиональное образовательное учреждение

Самарской области

«Тольяттинский социально-экономический колледж»

**КУРСОВАЯ РАБОТА**

**РАЗРАБОТКА И ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ «УЧЁТ ТОВАРОВ НА СКЛАДЕ ПРИ ПРЕДПРИЯТИИ»**

**ПМ.05 ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ИНФОРМАЦИОННЫХ СИСТЕМ**

**МДК 05.02 Разработка кода информационных систем**

**09.02.07 Информационные системы и программирование**

|  |  |  |  |
| --- | --- | --- | --- |
| Студент |  | **/** | К.А. Куприянов |
|  | *подпись* |  | *И.О. Фамилия* |
| \_\_.\_\_.2022 г. |  |  |  |
| Оценка выполнения и защиты курсовой работы | | |  |
|  | | |  |
| Руководитель |  | **/** | Е.В. Плюснина |
|  | *подпись* |  | *И.О. Фамилия* |
| \_\_.\_\_.2022 г. |  |  |  |

Тольятти, 2022

Государственное бюджетное профессиональное образовательное учреждение  
Самарской области  
«Тольяттинский социально-экономический колледж»

|  |  |  |
| --- | --- | --- |
|  |  | Утверждаю:  Заместитель директора по УР  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ М.С. Киронова  *« » 202 г.* |

**ЗАДАНИЕ**

на курсовую работу

**по ПМ.05 Проектирование и разработка информационных систем модуля, выполняемой в рамках МДК.05.02 Разработка кода информационных систем**

студента группы ИСП-31

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Куприянова Кирилла Александровича\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Фамилия Имя отчество студента*

Тема курсовой работы: «Разработка информационной системы **«**Учёт товаров на складе при предприятии»

1. Содержание задания:

1.1 Разработать физическую структуру информационной системы.

1.2 Разработать запросы к информационной системе.

1.3 Разработать приложение.

1. Исходные данные:

Исходные данные для практической реализации автоматизированной информационной системы (АИС) берутся из различных информационных источников (Интернет-ресурсы, печатные издания, периодика и др.).

1. Содержание курсовой работы

Введение

1 Разработка базы данных информационной системы «Учёт товаров на складе при предприятии»

1.1 Разработка физической структуры БД

1.2 Разработка запросов ИС

2 Разработка интерфейса информационной системы «Учёт товаров на складе при предприятии»

2.1 Разработка интерфейса

2.2 Разработка руководства пользователя

Заключение

Список использованных источников

Приложения

Приложение А Листинг программы

Дата выдачи задания: «13» января 2022 г.

Дата сдачи работы на отделение: «\_\_»\_\_\_\_\_\_\_\_2022 г.

Руководитель курсового(ой) проекта(работы) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ Е.В. Плюснина

подпись расшифровка подписи

**КАЛЕНДАРНЫЙ ПЛАН**

выполнения курсовой работы

Студентом 3 курса группы ИСП-31

По теме Разработка и проектирование ИС «Учёт товаров на сладе при предприятии»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| №  этапа  работы | Содержание этапов работы | Плановый срок выполнения этапа | Планируемый объем выполнения  этапа, % | Отметка  о  выполнении  этапа |
| 1 | Выбор, обоснование темы и объекта исследования | Январь 2022 | 5% |  |
| 2 | Утверждение темы, согласование плана. Введение, библиография | Январь 2022 | 10% |  |
| 3 | Изучение и анализ информационных материалов по теме | Февраль 2022 | 15% |  |
| 4 | Обоснование актуальности выбранной темы применительно к профессиональной деятельности (введение) | Февраль 2022 | 20% |  |
| 5 | Изложение материала основной части по теме курсовой работы | Февраль 2022 | 20% |  |
| 6 | Подведение итогов проведенного анализа, формулировка выводов УИР применительно к профессиональной деятельности (заключение) | Март 2022 | 20% |  |
| 7 | Оформление работы и сдача на проверку | Март 2022 | 10% |  |
| 8 | Защита работы |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Студент |  | **/** | К.А. Куприянов |
|  | *подпись* |  | *И.О. Фамилия* |
| \_\_.\_\_.2022 г. |  |  |  |
| Руководитель |  | **/** | Е.В. Плюснина |
|  | *подпись* |  | *И.О. Фамилия* |
| \_\_.\_\_.2022 г. |  |  |  |

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 5](#__RefHeading___Toc94035240)

[1 РАЗРАБОТКА БАЗЫ ДАННЫХ ИНФОРМАЦИОННОЙ СИСТЕМЫ УЧЁТ ТОВАРОВ НА СКЛАДЕ ПРИ ПРЕДПРИЯТИИ 7](#__RefHeading___Toc94035241)

[1.1 Разработка физической структуры БД 7](#__RefHeading___Toc94035242)

[1.2 Разработка запросов ИС 9](#__RefHeading___Toc94035243)

[2 РАЗРАБОТКА ИНТЕРФЕЙСА ИНФОРМАЦИОННОЙ СИСТЕМЫ УЧЁТ ТОВАРОВ НА СКЛАДЕ ПРИ ПРЕДПРИЯТИИ 11](#__RefHeading___Toc94035244)

2.1 Разработка интерфейса ……………………………………………………. 11

2.2 Разработка руководства пользователя ………………………….……….... 20

[ЗАКЛЮЧЕНИЕ 24](#__RefHeading___Toc94035246)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 25](#__RefHeading___Toc94035247)

[ПРИЛОЖЕНИЕ А.](#__RefHeading___Toc94035248) [ЛИСТИНГ ПРОГРАММЫ 27](#__RefHeading___Toc94035249)

# ВВЕДЕНИЕ

Актуальность внедрения автоматизированной информационной системы заключается в инновационном подходе хранения документов в электронной базе данных, является более выгодным вариантом, нежели хранение документов в бумажном формате.

Основной целью использования ИС являются создание открытого информационного пространства для эффективного решения задач посредством сетевого взаимодействия предприятия на основе информационно коммуникационных технологий.

ИС предназначена для комплексной автоматизации учёта на предприятии.

# При изучении предметной области, выяснилось, что информационная система «Расчет стоимости услуг, предоставляемых клиентам» на первый взгляд простая, но для ее реализации необходимо обратится к некоторым разделам, таким как создание базы данных и создание приложения для взаимодействия с этой базой данных.

# Предметом исследования процесса будет ИС «Учёт товаров на складе при предприятии».

# Целью курсовой работы является разработка и проектирование информационной системы «Учёт товаров на складе при предприятии».

# Для достижения поставленной цели были определены следующие задачи:

# разработать базы данных информационной системы;

# разработать физическую структуру базы данных;

# разработать запросы ИС;

# разработать интерфейс ИС.

# Для разработки информационной системы будут применяться методы: анализ деятельности предприятия с выявлением его функции, а также функциональное моделирование системы.

# 1 РАЗРАБОТКА БАЗЫ ДАННЫХ ИНФОРМАЦИОННОЙ СИСТЕМЫ УЧЁТ ТОВАРОВ НА СКЛАДЕ ПРИ ПРЕДПРИЯТИИ

## 1.1 Разработка физической структуры БД

Чтобы создать новую базу данных, откройте SQL Server Management Studio. Щелкните правой кнопкой мыши на узел Базы данных и выберите команду создать базу данных. В поле Новая база данных введите имя базы данных. Чтобы создать базу данных, приняв все значения по умолчанию, нажмите кнопку ОК.

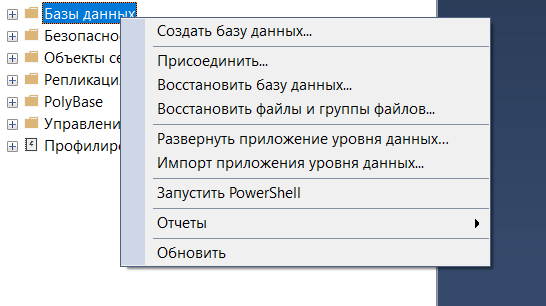
**

Рисунок 1.1 – Окно обозреватель объектов

В обозревателе объектов открываем контейнер «Базы данных», затем открываем нужную базу данных и щелкаем правой кнопкой мыши по пункту «Таблицы», после «Создать» и выбираем «Таблица…».

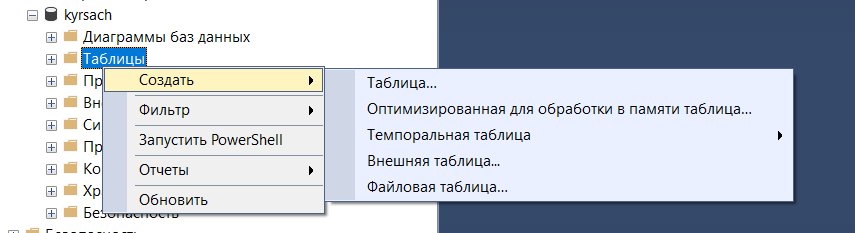
**

Рисунок 1.2 – Окно создания таблицы

Введите имена столбцов, выберите типы данных и определите для каждого столбца, могут ли в нем присутствовать значения NULL, как показано на иллюстрации ниже:

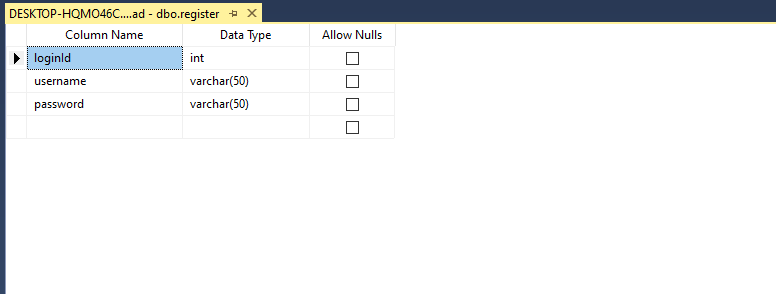


Рисунок 1.3 – Таблица «register»

Теперь можно заполнить нашу таблицу и создать в ней нужные нам строки со своими значениями, которые мы будем использовать в нашей программе:

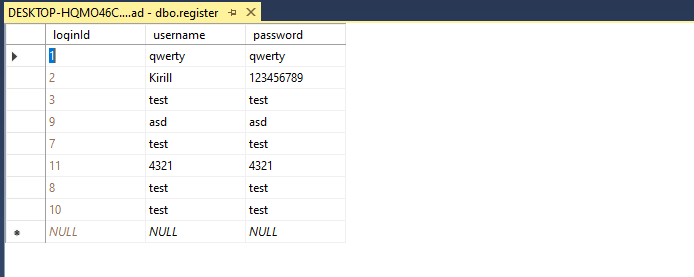


Рисунок 1.4 – Заполненная Таблица «register»

## 1.2 Разработка запросов ИС

Чтобы создать представление, в обозревателе объектов открываем контейнер «Базы данных», затем открываем нужную базу данных и щелкаем правой кнопкой мыши по пункту «Представления», после «Создать представление…».

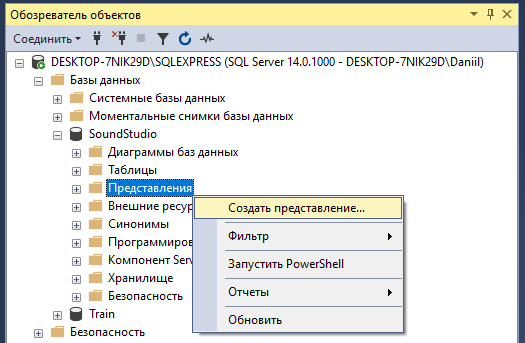


Рисунок 1.5 – Окно создания представления

В диалоговом окне «Добавление таблицы» выберите один или несколько элементов, которые необходимо включить в новое представление, на одной из следующих вкладок: «Таблицы», «Представления», «Функции» и «Синонимы». Щелкните «Добавить», а затем выберите Закрыть.

На Панели диаграмм выберите столбцы или другие элементы для включения в новое представление. На Панели критериев выберите дополнительные условия сортировки или фильтрации для столбцов.

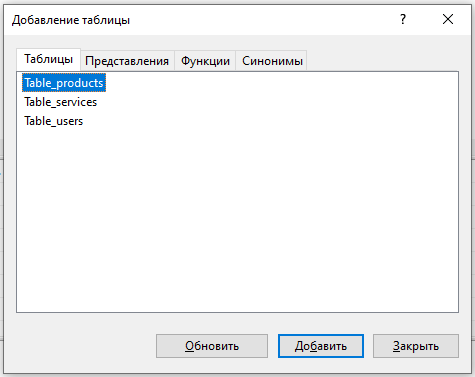


Рисунок 1.6 – Диалоговое окно добавление таблицы

В меню Файл выберите пункт «Сохранить» и введите название представления – «имя\_представления».

Затем создадим запрос для информационной системы путем выбора столбцов или других элементов представления. На панели критериев выберите дополнительные условия сортировки или фильтрации для столбцов.

Общая структура запроса выглядит следующим образом:

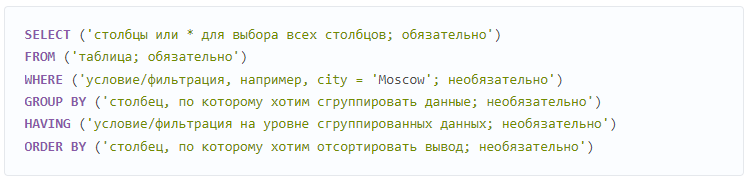


Рисунок 1.7 – Общая структура запроса

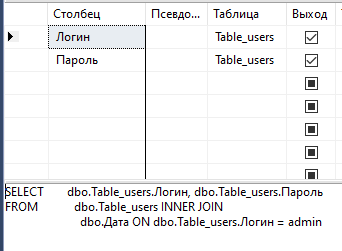


Рисунок 1.8 – Запрос Select для информационной системы

# Создание остальных запросов базы данных для информационной системы происходит аналогично примеру, описанному выше.

# 2 РАЗРАБОТКА ИНТЕРФЕЙСА ИНФОРМАЦИОННОЙ СИСТЕМЫ УЧЁТ ТОВАРОВ НА СКЛАДЕ ПРИ ПРЕДПРИЯТИИ

## 2.1 Разработка интерфейса

Первым шагом является создание инфраструктуры приложения. Создайте новый проект приложения WPF в Visual Basic или Visual C# и введите название. Откройте Visual Studio и выберите создать новый проект в меню начало работы. Откроется диалоговое окно «Создание нового проекта». В раскрывающемся списке язык выберите C#. Выберите шаблон приложения WPF (платформа .NET Framework) и нажмите кнопку далее.

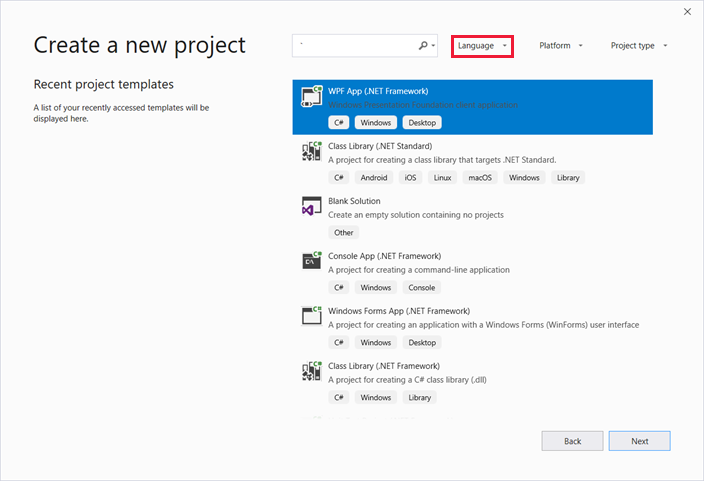


Рисунок 2.1 – Окно создания приложения WPF

Откроется диалоговое окно «Настройка нового проекта». Введите имя проекта и нажмите кнопку создать.

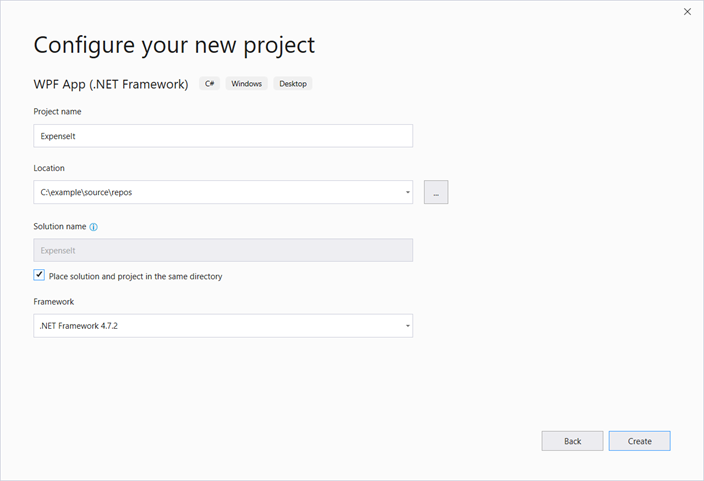


Рисунок 2.2 – Окно создания приложения WPF

Visual Studio создает проект и открывает конструктор для окна приложения по умолчанию с именем MainWindow.xaml.

Чтобы как-то взаимодействовать с пользователем, получать от пользователя ввод с клавиатуры или мыши и использовать введенные данные в программе, нам нужны элементы управления.

Создаем интерфейс окна авторизации, используя элементы управления Label, Image для визуального восприятия информации;

TextBox, PasswordBox для ввода логина и пароля; Button для закрытия окна и для входа в систему.

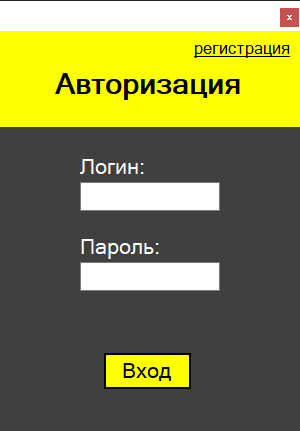


Рисунок 2.3 – Интерфейс окна «Авторизация»

Затем откроем App.xaml (C#). Этот XAML-файл определяет приложение WPF и все ресурсы приложения. Этот файл также используется для указания пользовательского интерфейса, в данном случае окна авторизации, которое автоматически отображается при запуске приложения.

После этого добавим проверку (Captcha) в случае неверного ввода логина или пароля в окно авторизации. Интерфейс окна авторизации с проверкой будет выглядеть следующим образом:

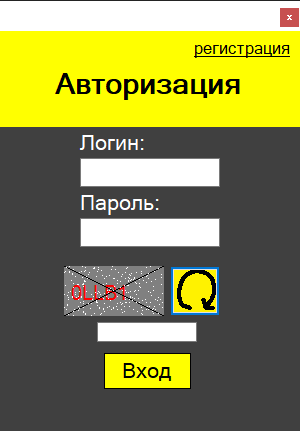


Рисунок 2.4 – Интерфейс окна «Авторизация» с проверкой ввода логина или пароля

Следом перейдем к созданию кода на языке C# для окна авторизации. В обозревателе решений нажмем правой кнопкой мыши по проекту, выберем пункт «Добавить» - «Создать элемент».

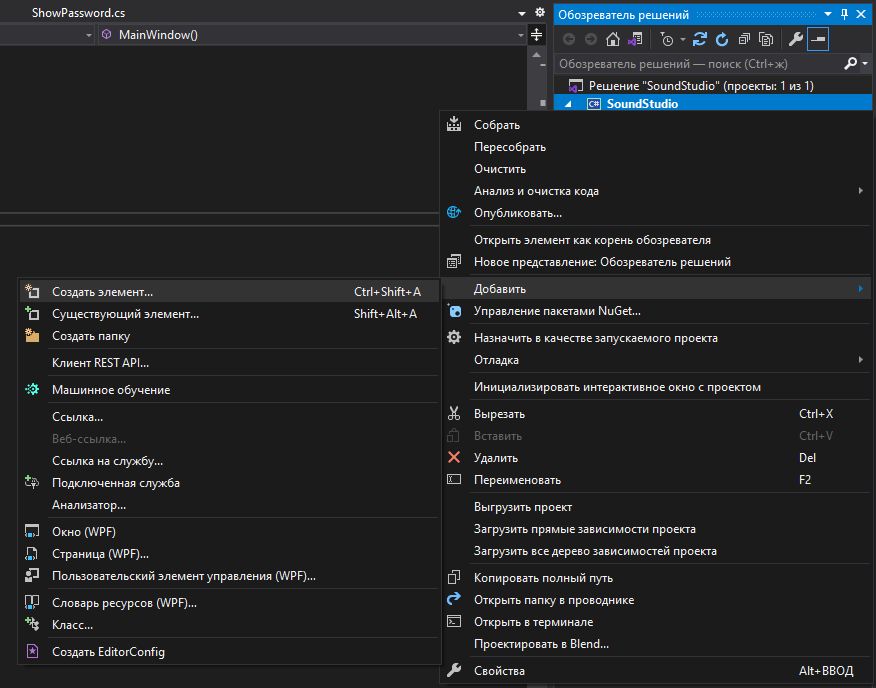


Рисунок 2.5 – Окно обозреватель решений

В открывшемся окне «Добавление нового элемента» выберем элемент «Класс Visual C#» и дадим ему название. Нажмем кнопку «Добавить» и перейдем к нему.

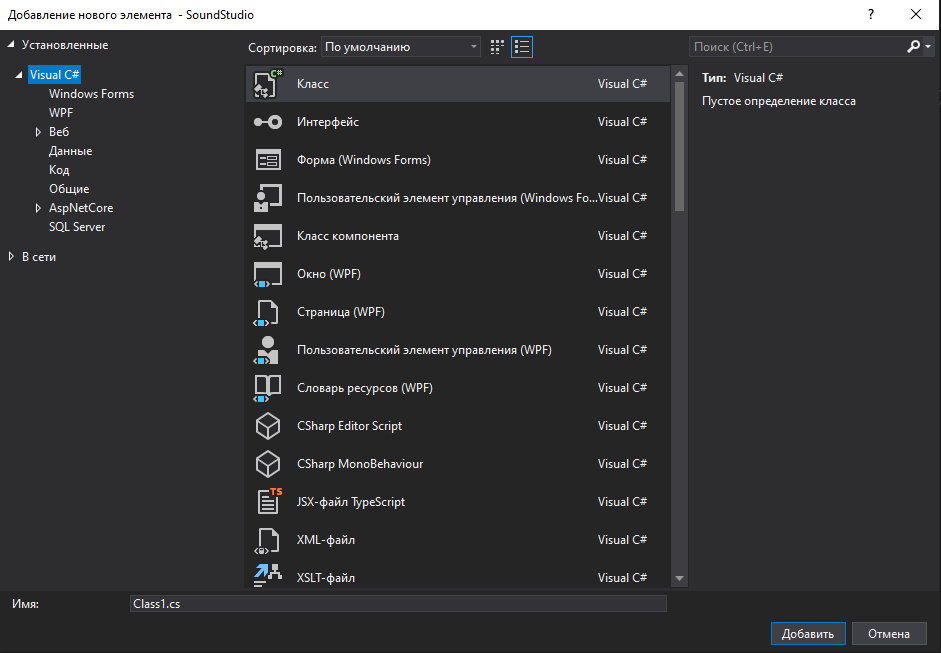


Рисунок 2.6 – Окно добавления нового класса

Создадим класс Connection, который будет брать данные о пользователе из sql-базы данных, используя метод Select.

class DB1

{

SqlConnection SqlConnection = new SqlConnection(@"Data Source=DESKTOP-HQMO46C;Initial Catalog=Sklad;Integrated Security=True");

public void openConenection()

{

if (SqlConnection.State == System.Data.ConnectionState.Closed)

{

SqlConnection.Open();

}

}

public void closeConenection()

{

if (SqlConnection.State == System.Data.ConnectionState.Open)

{

SqlConnection.Close();

}

}

public SqlConnection getConnection()

{

return SqlConnection;

}

}

Затем настроим кнопку, отвечающий за вход в систему. В этой кнопке будет обращение к классу DB1 для получения информации из sql-базы данных о пользователе. Поставим условия входа – при удачном вводе логина и пароля, пользователь попадает на главное окно. При ошибке ввода добавляется проверка (Captcha) с возможность обновления.

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

{

Form3 fr3 = new Form3();

Form2 fr2 = new Form2();

var loginUser = Login.Text;

var passUser = Password.Text;

SqlDataAdapter adapter = new SqlDataAdapter();

DataTable table = new DataTable();

string querysting = $"select loginId, username, password from register where username = '{loginUser}' and password = '{passUser}'";

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

adapter.SelectCommand = command;

adapter.Fill(table);

if (Login.Text != "")

{

if (Password.Text != "")

{

if ((Login.Text == "admin") && (Password.Text == "admin"))

{

fr3.Show();

Hide();

}

else if ((Login.Text == "admin") && (Password.Text != "admin")) //возможно лишнее условие

{

fr2.Show();

Hide();

}

else if (table.Rows.Count == 1)

{

fr3.Show();

Hide();

}

else

{

fr2.Show();

Hide();

}

}

else

{

MessageBox.Show("Не введён пороль");

}

}

else

{

MessageBox.Show("Не введён логин");

}

}

Для проверки неверного ввода логина или пароля создадим класс CreateImage, который будет генерировать проверку (Captcha).

private Bitmap CreateImage(int Width, int Height)

{

Random rnd = new Random();

Bitmap result = new Bitmap(Width, Height);

int Xpos = rnd.Next(0, Width - 50);

int Ypos = rnd.Next(15, Height - 15);

Brush[] colors = { Brushes.Black,

Brushes.Red,

Brushes.RoyalBlue,

Brushes.Green };

Graphics g = Graphics.FromImage((Image)result);

g.Clear(Color.Gray);

text = String.Empty;

string ALF = "1234567890QWERTYUIOPASDFGHJKLZXCVBNM";

for (int i = 0; i < 5; ++i)

text += ALF[rnd.Next(ALF.Length)];

g.DrawString(text,

new Font("Arial", 15),

colors[rnd.Next(colors.Length)],

new PointF(Xpos, Ypos));

g.DrawLine(Pens.Black,

new Point(0, 0),

new Point(Width - 1, Height - 1));

g.DrawLine(Pens.Black,

new Point(0, Height - 1),

new Point(Width - 1, 0));

for (int i = 0; i < Width; ++i)

for (int j = 0; j < Height; ++j)

if (rnd.Next() % 20 == 0)

result.SetPixel(i, j, Color.White);

return result;

}

Чтобы была возможность обновить проверку (Captcha), нам понадобится кнопка, код которой приведён ниже.

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

{

pictureBox1.Image = this.CreateImage(pictureBox1.Width, pictureBox1.Height);

}

Конечный код для окна авторизации, в котором будет проверяться правильность введенных логина пароля и (Captcha), будет выглядеть следующим образом:

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

{

Form3 fr3 = new Form3();

var loginUser = Login.Text;

var passUser = Password.Text;

SqlDataAdapter adapter = new SqlDataAdapter();

DataTable table = new DataTable();

string querysting = $"select loginId, username, password from register where username = '{loginUser}' and password = '{passUser}'";

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

adapter.SelectCommand = command;

adapter.Fill(table);

if (Login.Text != "")

{

if (Password.Text != "")

{

if ((Login.Text == "admin") && (Password.Text == "admin"))

{

if (textBox1.Text == this.text)

{

fr3.Show();

Hide();

}

else

{

MessageBox.Show("Неверно введена капча");

}

}

else if ((Login.Text == "admin") && (Password.Text != "admin"))

{

MessageBox.Show("Неверно введён пароль");

}

else if (table.Rows.Count == 1)

{

if (textBox1.Text == this.text)

{

fr3.Show();

Hide();

}

else

{

MessageBox.Show("Неверно введена капча");

}

}

else

{

MessageBox.Show("Неверно введён логин или пороль");

}

}

else

{

MessageBox.Show("Не введён пороль");

}

}

else

{

MessageBox.Show("Не введён логин");

}

}

## Следом добавим главное окно проекта. Чтобы осуществить открытие нового окна приложения, для начала нужно создать окно. Для этого правой кнопкой мыши (ПКМ) нажимаем на название проекта в «Обозреватель решений» — пункт «Добавить» — «Окно».

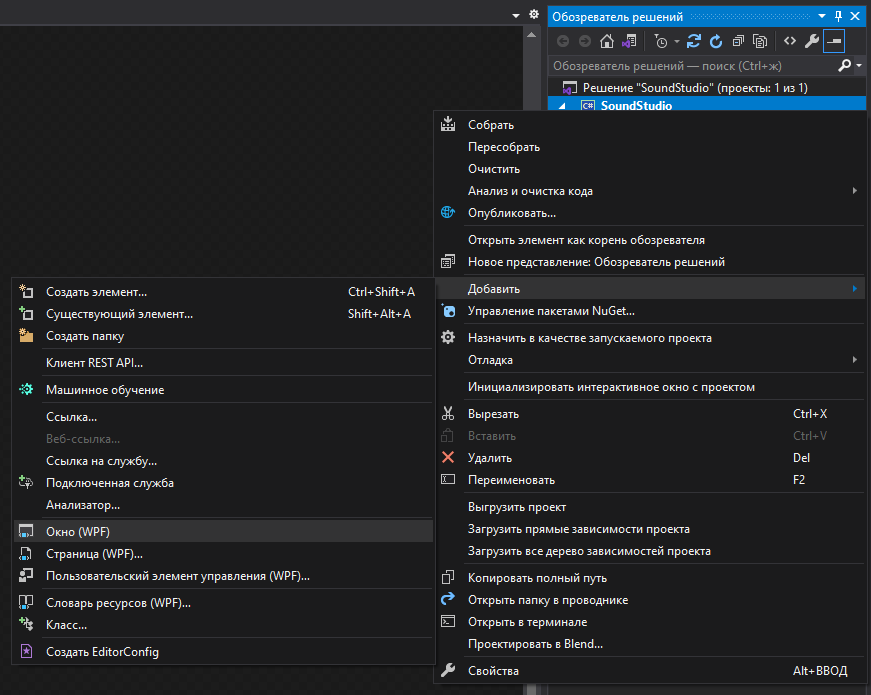


Рисунок 2.7 – Добавление главного окна проекта

После нужно указать название и нажать кнопку «Далее».

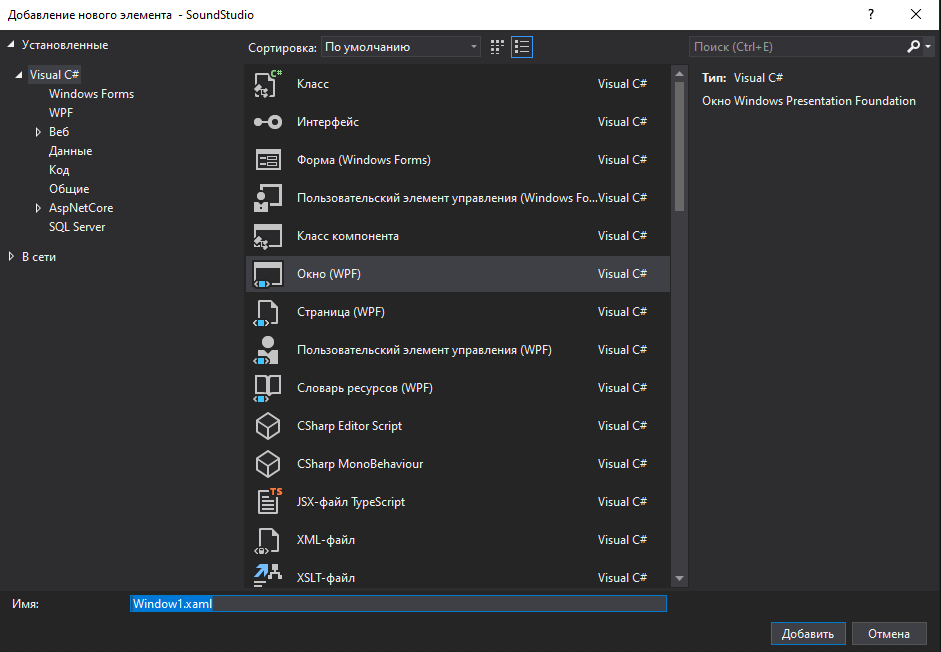


Рисунок 2.8 – Добавление имени главного окна проекта

Интерфейс главного окна и программный код разрабатываются аналогично описанному выше примеру.

## 2.2 Разработка руководства пользователя

Информационная система «Учёт товаров на складе» представляет собой программный комплекс для учёта, внесения, изменения и удаления информации в базу данных.

Данное руководство предназначено для пользователей приложения, установленного на устройствах, работающих под управлением операционной системы Windows 10.

Для установки приложения компьютер конечного пользователя должен удовлетворять требованиям.

Минимальная конфигурация сервера:

* + - тип процессора – intel core i3 и выше или совместимый с ним;
    - объем оперативного запоминающего устройства 8Гб и более;
    - жесткий диск 60 Гб;
    - модем, для выхода в Интернет;
    - монитор, клавиатура, мышь.

Требования, предъявляемые к конфигурации клиентских станций:

* + - процессор, с тактовой частотой не менее 1200 MHz,
    - 2 GB оперативной памяти;
    - монитор – SVGA;
    - клавиатура - 101/102 клавиши;
    - манипулятор типа «мышь».

Для реализации данной системы требуется установка следующего ПО: Microsoft Visual Studio 2021 и СУБД Microsoft SQL Server 2018 Express Edition.

Для авторизации введите логин и пароль в открывшемся окне при входе в приложение, затем нажмите кнопку «Вход». Если вы не введёте Логин или Пароль, то выйдет соответствующее оповещение: «Вы не ввели логин» или «Вы не ввели пороль». Если вы неправильно введёте Логин или Пароль то вам придётся ввести (Captcha).

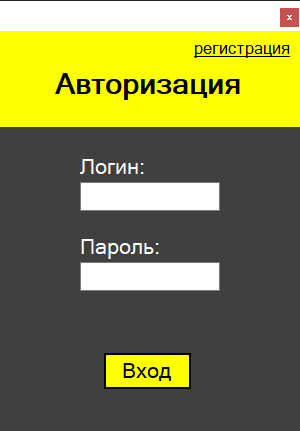
 

Рисунок 2.9 – Окно Авторизации

Рисунок 2.10 – Окно Авторизации с (Captcha)

Когда вы введёте логин, пороль и (Captcha) вас перекинет на главное окно программы.

На главном окне находится три панели, впервой, представлены кнопки для внесения, удаления, изменения и сохранения всех изменений в базу данных.

В панели ниже расположены строки использующиеся для отображения информации о выбранной строке в таблице базы данных и дальнейшего их изменения.

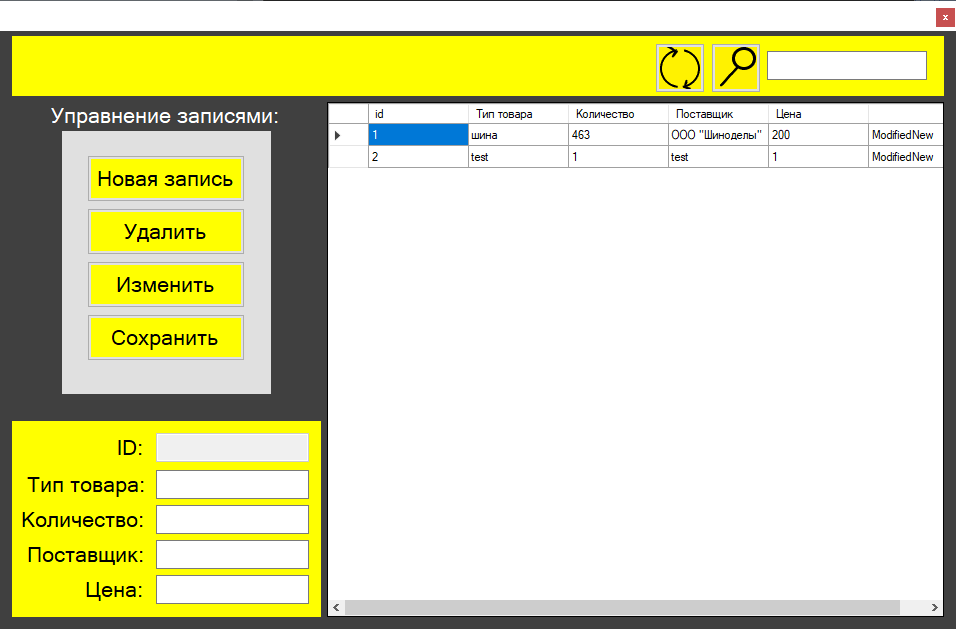


Рисунок 2.11 – Главное окно

А основную часть окна занимает таблица, в которой отображаются данные строк из таблиц базы данных. Эту таблицу можно отсортировать с помощью строки, в верхней панели введя туда нужные значения или обновить её для отображения проведённых изменений.

При нажатие на кнопку «Удалить» программа удалит из таблицы выделенную строку (она удалится в том числе и из базы данных).

При нажатие на кнопку «Изменить» можно спомощью панели снизу изменить параметры выделенной строки из таблицы, введя туда нужные вам параметры и значения, после чего нужно будет нажать на кнопку «Сохранить», чтобы ваши изменения корректно отобразились в таблице и сохранились в базе данных.

При нажатие на кнопку «Новая запись» открывается окно в котором нужно ввести: тип, количество, поставщика и цену товара.

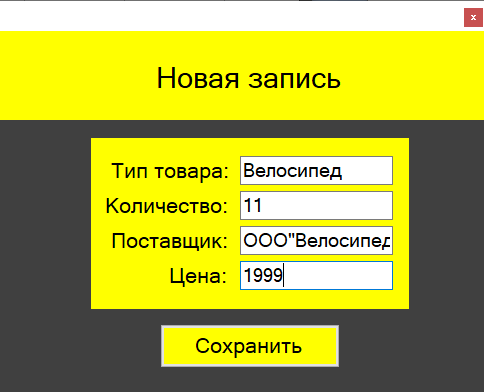


Рисунок 2.12 – Окно добавления новой записи

После чего нажимаем кнопку «Сохранить» и закрываем это окно с помощью крестика сверху-справа. Для отображения изменений в таблице нажимаем на значок «круглых стрелок» в верхней панели.

# ЗАКЛЮЧЕНИЕ

В первой части курсовой работы была произведена разработка базы данных информационной системы «Учёт товаров на складе при предприятии». Была разработана физическая структура базы данных для информационной системы. Также, был рассмотрен процесс разработки запросов ИС «Учёт товаров на складе при предприятии». В дальнейшем создана база данных системы в программной среде MS SQL Server Management Studio 18.

Во второй части курсовой работы на основе рассмотренных процессов отдела «Учёт товаров на складе при предприятии» был разработан и подробно описан интерфейс информационной системы в программе MS Visual Studio 19, а также было разработано руководство пользователя.

Затем был разработан программный модуль автоматизированной информационной системы «Учёт товаров на складе при предприятии» с собственным интерфейсом и запросами при помощи программ Microsoft Visual Studio 2019 и MS SQL Server Management Studio 18.

Подробно описан и проиллюстрирован функционал и структура самого модуля, а также проведено тестирование готового программного продукта. В последнюю очередь была составлена сопутствующая документация.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ГОСТ 34.601 – 90. Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания.
2. ГОСТ 34.602 - 2020. Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы.
3. ГОСТ 19.201 - 78 ЕСПД. Техническое задание. Требования к содержанию и оформлению.
4. ГОСТ 19.202 - 78 ЕСПД. Спецификация. Требования к содержанию и оформлению.
5. ГОСТ Р ИСО/МЭК 12207. Процессы жизненного цикла программных средств.
6. Золотов С.Ю. Проектирование информационных систем: Учебно-методическое пособие. Методические рекомендации для выполнения курсового проекта, лабораторных работ и практических занятий по дисциплине «Проектирование информационных систем» - Томск: ТУСУР, 2013. - 34 с.
7. Шнайдер, Роберт Microsoft SQL Server 6.5. Проектирование высокопроизводительных баз данных; М.: Лори, 2010. - 361 c
8. Петкович, Душан Microsoft SQL Server 2012. Руководство для начинающих / Душан Петкович. - М.: БХВ-Петербург, 2012. - 460 c.
9. Тейлор, Аллен SQL для чайников / Аллен Тейлор. - М.: Вильямс, 2014. - 416 c.
10. Браст, Э.Дж. Разработка приложений на основе Microsoft SQL Server 2008 / Э.Дж. Браст. - М.: Русская Редакция, 2010. - 751 c.
11. Хетагуров, Я. А. Проектирование автоматизированных систем обработки информации и управления (АСОИУ). Учебник / Я.А. Хетагуров. - М.: Бином. Лаборатория знаний, 2015. - 240 c.
12. Кристофер, Д. Маннинг Введение в информационный поиск / Кристофер Д. Маннинг, ПрабхакарРагхаван ,ХайнрихШютце. - М.: Вильямс, 2014. - 528 c.
13. Бишоп Дж. С# в кратком изложении; Бином. Лаборатория знаний - М., 2015. - 234 c.
14. ГриффитсИэн Программирование на C# 5.0; Эксмо - М., 2014. - 580 c.
15. Гуриков С. Р. Введение в программирование на языке Visual C#; ИЛ - Москва, 2013. - 448 c.
16. Шилдт Герберт C# 4.0. Полное руководство; Вильямс - М., 2015. - 291 c.
17. Эндрю Троелсен Язык программирования C# 5.0 и платформа .NET 4.5; Диалектика / Вильямс - М., 2015. - 126 c.

# ПРИЛОЖЕНИЕ А

# ЛИСТИНГ ПРОГРАММЫ

**Класс Базы данных:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data.SqlClient;

namespace Отчёт\_товаров\_на\_складе

{

class DB1

{

SqlConnection SqlConnection = new SqlConnection(@"Data Source=DESKTOP-HQMO46C;Initial Catalog=Sklad;Integrated Security=True");

public void openConenection()

{

if (SqlConnection.State == System.Data.ConnectionState.Closed)

{

SqlConnection.Open();

}

}

public void closeConenection()

{

if (SqlConnection.State == System.Data.ConnectionState.Open)

{

SqlConnection.Close();

}

}

public SqlConnection getConnection()

{

return SqlConnection;

}

}

}

**Код окна Авторизации:**

using System;

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.SqlClient;

namespace Отчёт\_товаров\_на\_складе

{

public partial class Form1 : Form

{

DB1 database = new DB1();

public Form1()

{

InitializeComponent();

}

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

{

Form3 fr3 = new Form3();

Form2 fr2 = new Form2();

var loginUser = Login.Text;

var passUser = Password.Text;

SqlDataAdapter adapter = new SqlDataAdapter();

DataTable table = new DataTable();

string querysting = $"select loginId, username, password from register where username = '{loginUser}' and password = '{passUser}'";

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

adapter.SelectCommand = command;

adapter.Fill(table);

if (Login.Text != "")

{

if (Password.Text != "")

{

if ((Login.Text == "admin") && (Password.Text == "admin"))

{

fr3.Show();

Hide();

}

else if ((Login.Text == "admin") && (Password.Text != "admin"))

{

fr2.Show();

Hide();

}

else if (table.Rows.Count == 1)

{

fr3.Show();

Hide();

}

else

{

fr2.Show();

Hide();

}

}

else

{

MessageBox.Show("Не введён пороль");

}

}

else

{

MessageBox.Show("Не введён логин");

}

}

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

{

Form4 fr4 = new Form4();

fr4.Show();

Hide();

}

}

}

**Код окна Авторизации с (Captcha):**

using System;

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.SqlClient;

namespace Отчёт\_товаров\_на\_складе

{

public partial class Form2 : Form

{

DB1 database = new DB1();

public Form2()

{

InitializeComponent();

}

private void Form2\_Load(object sender, EventArgs e)

{

pictureBox1.Image = this.CreateImage(pictureBox1.Width, pictureBox1.Height);

}

private string text = String.Empty;

private Bitmap CreateImage(int Width, int Height)

{

Random rnd = new Random();

Bitmap result = new Bitmap(Width, Height);

int Xpos = rnd.Next(0, Width - 50);

int Ypos = rnd.Next(15, Height - 15);

Brush[] colors = { Brushes.Black,

Brushes.Red,

Brushes.RoyalBlue,

Brushes.Green };

Graphics g = Graphics.FromImage((Image)result);

g.Clear(Color.Gray);

text = String.Empty;

string ALF = "1234567890QWERTYUIOPASDFGHJKLZXCVBNM";

for (int i = 0; i < 5; ++i)

text += ALF[rnd.Next(ALF.Length)];

g.DrawString(text,

new Font("Arial", 15),

colors[rnd.Next(colors.Length)],

new PointF(Xpos, Ypos));

g.DrawLine(Pens.Black,

new Point(0, 0),

new Point(Width - 1, Height - 1));

g.DrawLine(Pens.Black,

new Point(0, Height - 1),

new Point(Width - 1, 0));

for (int i = 0; i < Width; ++i)

for (int j = 0; j < Height; ++j)

if (rnd.Next() % 20 == 0)

result.SetPixel(i, j, Color.White);

return result;

}

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

{

pictureBox1.Image = this.CreateImage(pictureBox1.Width, pictureBox1.Height);

}

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

{

Form3 fr3 = new Form3();

var loginUser = Login.Text;

var passUser = Password.Text;

SqlDataAdapter adapter = new SqlDataAdapter();

DataTable table = new DataTable();

string querysting = $"select loginId, username, password from register where username = '{loginUser}' and password = '{passUser}'";

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

adapter.SelectCommand = command;

adapter.Fill(table);

if (Login.Text != "")

{

if (Password.Text != "")

{

if ((Login.Text == "admin") && (Password.Text == "admin"))

{

if (textBox1.Text == this.text)

{

fr3.Show();

Hide();

}

else

{

MessageBox.Show("Неверно введена капча");

}

}

else if ((Login.Text == "admin") && (Password.Text != "admin"))

{

MessageBox.Show("Неверно введён пароль");

}

else if (table.Rows.Count == 1)

{

if (textBox1.Text == this.text)

{

fr3.Show();

Hide();

}

else

{

MessageBox.Show("Неверно введена капча");

}

}

else

{

MessageBox.Show("Неверно введён логин или пороль");

}

}

else

{

MessageBox.Show("Не введён пороль");

}

}

else

{

MessageBox.Show("Не введён логин");

}

}

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

{

Form4 fr4 = new Form4();

fr4.Show();

Hide();

}

}

}

**Код окна Регистрации:**

using System;

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.SqlClient;

namespace Отчёт\_товаров\_на\_складе

{

public partial class Form4 : Form

{

DB1 dataBase = new DB1();

public Form4()

{

InitializeComponent();

}

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

{

Form2 fr2 = new Form2();

fr2.Show();

Hide();

}

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

{

var login = Login.Text;

var password = Password.Text;

var loginUser = Login.Text;

SqlDataAdapter adapter = new SqlDataAdapter();

DataTable table = new DataTable();

string proverka = $"select loginId, username, password from register where username = '{loginUser}'";

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

adapter.SelectCommand = command;

adapter.Fill(table);

dataBase.openConenection();

if (Login.Text != "")

{

if (Password.Text != "")

{

if (Password.Text == Password2.Text)

{

if (table.Rows.Count > 0)

{

MessageBox.Show("Такой аккаунт уже существует");

}

else

{

string querysting = $"insert into register(username, password) values('{login}', '{password}')";

SqlCommand command2 = new SqlCommand(querysting, dataBase.getConnection());

command2.ExecuteNonQuery();

MessageBox.Show("Аккаунт зарегестрирован.");

}

}

else

{

MessageBox.Show("Пароли не совпадают.");

}

}

else

{

MessageBox.Show("Не введён пороль");

}

}

else

{

MessageBox.Show("Не введён логин");

}

dataBase.closeConenection();

}

}

}

**Код Основного окна:**

using System;

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.SqlClient;

namespace Отчёт\_товаров\_на\_складе

{

enum RowState

{

Existed,

New,

Modified,

ModifiedNew,

Deleted

}

public partial class Form3 : Form

{

DB1 database = new DB1();

int selectedRow;

public Form3()

{

InitializeComponent();

}

private void CreateColumns()

{

dataGridView1.Columns.Add("id", "id");

dataGridView1.Columns.Add("type\_of", "Тип товара");

dataGridView1.Columns.Add("count\_of", "Количество");

dataGridView1.Columns.Add("postavka", "Поставщик");

dataGridView1.Columns.Add("price", "Цена");

dataGridView1.Columns.Add("IsNew", String.Empty);

}

private void clearFields()

{

textBoxID.Text = "";

textBoxType.Text = "";

textBoxCount.Text = "";

textBoxPostav.Text = "";

textBoxPrice.Text = "";

}

private void ReadSingleRow(DataGridView dgw, IDataRecord record)

{

dgw.Rows.Add(record.GetInt32(0), record.GetString(1), record.GetInt32(2), record.GetString(3), record.GetInt32(4), RowState.ModifiedNew);

}

private void refreshDataGrid(DataGridView dgw)

{

dgw.Rows.Clear();

string queryString = $"select \* from Tovar";

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

database.openConenection();

SqlDataReader reader = command.ExecuteReader();

while (reader.Read())

{

ReadSingleRow(dgw, reader);

}

reader.Close();

}

private void Form3\_Load(object sender, EventArgs e)

{

CreateColumns();

refreshDataGrid(dataGridView1);

}

private void dataGridView1\_CellClick(object sender, DataGridViewCellEventArgs e)

{

selectedRow = e.RowIndex;

if (e.RowIndex >= 0)

{

DataGridViewRow row = dataGridView1.Rows[selectedRow];

textBoxID.Text = row.Cells[0].Value.ToString();

textBoxType.Text = row.Cells[1].Value.ToString();

textBoxCount.Text = row.Cells[2].Value.ToString();

textBoxPostav.Text = row.Cells[3].Value.ToString();

textBoxPrice.Text = row.Cells[4].Value.ToString();

}

}

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

{

refreshDataGrid(dataGridView1);

}

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

{

Form5 fr5 = new Form5();

fr5.Show();

}

private void Searsh(DataGridView dgw)

{

dgw.Rows.Clear();

string searshString = $"select \* from Tovar where concat (id, type\_of, count\_of, postavka, price) like '%" + textBoxSearch.Text + "%'";

SqlCommand com = new SqlCommand(searshString, database.getConnection());

database.openConenection();

SqlDataReader read = com.ExecuteReader();

while (read.Read())

{

ReadSingleRow(dgw, read);

}

read.Close();

}

private void textBoxSearch\_TextChanged(object sender, EventArgs e)

{

Searsh(dataGridView1);

}

private void deleteRow()

{

int index = dataGridView1.CurrentCell.RowIndex;

dataGridView1.Rows[index].Visible = false;

int id = Convert.ToInt32(dataGridView1.Rows[index].Cells[0].Value);

var deleteQuery = $"delete from Tovar where id = {id}";

var command = new SqlCommand(deleteQuery, database.getConnection());

command.ExecuteNonQuery();

}

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

{

deleteRow();

clearFields();

}

private new void Update()

{

database.openConenection();

for (int index = 0; index < dataGridView1.Rows.Count; index++)

{

var rowState = (RowState)dataGridView1.Rows[index].Cells[0].Value;

if (rowState == RowState.Existed)

{

continue;

}

if (rowState == RowState.Modified)

{

var id = dataGridView1.Rows[index].Cells[0].Value.ToString();

var type = dataGridView1.Rows[index].Cells[1].Value.ToString();

var count = dataGridView1.Rows[index].Cells[2].Value.ToString();

var postavka = dataGridView1.Rows[index].Cells[3].Value.ToString();

var price = dataGridView1.Rows[index].Cells[4].Value.ToString();

var changeQuery = $"update Tovar set type\_of = '{type}', count\_of = '{count}', postavka = '{postavka}', price = '{price}' where id = '{id}'";

var command = new SqlCommand(changeQuery, database.getConnection());

command.ExecuteNonQuery();

}

}

database.closeConenection();

}

private void button4\_Click(object sender, EventArgs e)

{

Update();

}

private void Change()

{

var selectedRowIndex = dataGridView1.CurrentCell.RowIndex;

var id = textBoxID.Text;

var type = textBoxType.Text;

int count = Convert.ToInt32(textBoxCount.Text);

var postav = textBoxPostav.Text;

int price = Convert.ToInt32(textBoxPrice.Text);

if (dataGridView1.Rows[selectedRowIndex].Cells[0].Value.ToString() != string.Empty)

{

dataGridView1.Rows[selectedRowIndex].SetValues(id, type, count, postav, price);

dataGridView1.Rows[selectedRowIndex].Cells[5].Value = RowState.Modified;

}

}

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

{

Change();

clearFields();

}

private void textBoxCount\_KeyPress(object sender, KeyPressEventArgs e)

{

char ch = e.KeyChar;

if (!Char.IsDigit(ch) && ch != 8)

{

e.Handled = true;

}

}

private void textBoxPrice\_KeyPress(object sender, KeyPressEventArgs e)

{

char ch = e.KeyChar;

if (!Char.IsDigit(ch) && ch != 8)

{

e.Handled = true;

}

}

}

}

**Код окна добавления новой записи:**

using System;

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.SqlClient;

namespace Отчёт\_товаров\_на\_складе

{

public partial class Form5 : Form

{

DB1 database = new DB1();

public Form5()

{

InitializeComponent();

}

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

{

database.openConenection();

var type = textBoxType.Text;

int count = Convert.ToInt32(textBoxCount.Text);

var postav = textBoxPostav.Text;

int price = Convert.ToInt32(textBoxPrice.Text);

if (textBoxType.Text != "")

{

if (textBoxCount.Text != "")

{

if (textBoxPostav.Text != "")

{

if (textBoxPrice.Text != "")

{

var addQuery = $"insert into Tovar(type\_of, count\_of, postavka, price) values('{type}', '{count}', '{postav}', '{price}')";

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

command.ExecuteNonQuery();

MessageBox.Show("Запись успешно создана.");

}

else

{

MessageBox.Show("Вы не ввели цену.");

}

}

else

{

MessageBox.Show("Вы не указали поставщика.");

}

}

else

{

MessageBox.Show("Вы не ввели количество.");

}

}

else

{

MessageBox.Show("Вы не ввели тип товара.");

}

database.closeConenection();

}

private void textBoxPrice\_KeyPress(object sender, KeyPressEventArgs e)

{

char ch = e.KeyChar;

if (!Char.IsDigit(ch) && ch != 8)

{

e.Handled = true;

}

}

private void textBoxCount\_KeyPress(object sender, KeyPressEventArgs e)

{

char ch = e.KeyChar;

if (!Char.IsDigit(ch) && ch != 8)

{

e.Handled = true;

}

}

}

}