|  |
| --- |
| МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  федеральное государственное АВТОНОМНОЕ образовательное учреждение высшего образования  «Национальный исследовательский ядерный университет «МИФИ» |
| **Димитровградский инженерно-технологический институт –**  филиал федерального государственного автономного образовательного учреждения высшего образования «Национальный исследовательский ядерный университет «МИФИ»  **(ДИТИ НИЯУ МИФИ)** |

**Специальность 09.02.07 Информационные системы и программирование**

**МДК.01.01. Разработка программных модулей**

Дифференцированный зачет

«База данных «Столовая»»

Выполнил студент 3 курса 332 группы

Серых Павел Андреевич

(Ф.И.О. студента)

Преподаватель ДИТИ НИЯУ МИФИ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_А.В.Надеждина

Работа сдана 21 апреля 2022г

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Димитровград

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc101448867)

[ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 4](#_Toc101448868)

[ПРАКТИЧЕСКАЯ ЧАСТЬ 6](#_Toc101448869)

[Задание 1. Создание дизайна для окна авторизации 6](#_Toc101448870)

[Задание 2. Подключение MySQL и создание базы данных 6](#_Toc101448871)

[Задание 3. Авторизация пользователя через базу данных 7](#_Toc101448872)

[Задание 4. Окно регистрации пользователя 8](#_Toc101448873)

[Задание 5. Регистрация пользователя в базе данных 8](#_Toc101448874)

[Задание 6. Переход между окнами в приложении 9](#_Toc101448875)

[Задание 7. Создать в БД таблицу, по вашей теме, чтобы она выводилась на форме "Главный экран", после авторизации. 9](#_Toc101448876)

[ЗАКЛЮЧЕНИЕ 19](#_Toc101448877)

[СПИСОК ИСПОЛЬЗОВАНЫХ ИСТОЧНИКОВ 20](#_Toc101448878)

# **ВВЕДЕНИЕ**

Язык программирования С# был создан в конце 1990-х годов и стал частью общей .NET-стратегии Microsoft. Впервые он увидел свет в качестве б-версии в середине 2000 года. Главным архитектором С# был Андерс Хейлсберг (Anders Hejlsberg) -- один из ведущих специалистов в области языков программирования, получивший признание во всем мире. Достаточно сказать, что в 1980-х он был автором весьма успешного продукта Turbo Pascal, изящная реализация которого установила стандарт для всех будущих компиляторов. С# непосредственно связан с С, C++ и Java. И это не случайно. Эти три языка -- самые популярные и самые любимые языки программирования в мире. Более того, почти все профессиональные программисты сегодня знают С и C++, и большинство знает Java. Поскольку С# построен на прочном, понятном фундаменте, то переход от этих "фундаментальных" языков к "надстройке" происходит без особых усилий со стороны программистов. Так как Андерс Хейлсберг не собирался изобретать свое "колесо", он сосредоточился на введении усовершенствований и новшеств.

"Дедушкой" С# является язык С. От С язык С# унаследовал синтаксис, многие ключевые слова и операторы. Кроме того, С# построен на улучшенной объектной модели, определенной в C++. Если вы знаете С или C++, то с С# вы сразу станете друзьями. С# и Java связаны между собой несколько сложнее. Как упоминалось выше, Java также является потомком С и C++. У него тоже общий с ними синтаксис и сходная объектная модель. Подобно Java C# предназначен для создания переносимого кода. Однако С# -- не потомок Java. Скорее С# и Java можно считать двоюродными братьями, имеющими общих предков, но получившими от родителей разные наборы "генов".

К достоинствам языка С# следует отнести то, что данный язык имеет 300 000 библиотек разных функций, которые работают с максимальным быстродействием. Алгоритмы и программы, которые представляют интерес для изучения и исследования, исчисляются десятками и сотнями. Одни из них решают конкретные задачи на основе известных методов, другие образуют «системную» основу для огромного большинства прикладных программ. К последней группе алгоритмов можно отнести методы обработки информации, представленной в виде одномерных и двухмерных массивов (матриц). Работа с матрицами — это одна из стандартных функций, возникающих в процессе решения задач самого различного характера.

# **ТЕОРЕТИЧЕСКАЯ ЧАСТЬ**

Windows Forms — это платформа пользовательского интерфейса для создания классических приложений Windows. Она обеспечивает один из самых эффективных способов создания классических приложений с помощью визуального конструктора в Visual Studio. Такие функции, как размещение визуальных элементов управления путем перетаскивания, упрощают создание классических приложений.

В Windows Forms можно разрабатывать графически сложные приложения, которые просто развертывать, обновлять, и с которыми удобно работать как в автономном режиме, так и в сети. Приложения Windows Forms могут получать доступ к локальному оборудованию и файловой системе компьютера, на котором работает приложение.

На компьютере можно установить один или несколько экземпляров SQL Server . Каждый экземпляр SQL Server может содержать одну или несколько баз данных. В базе данных может содержаться одна или несколько групп объектов владения, которые называются схемами. В каждой схеме присутствуют объекты базы данных, такие как таблицы, представления и хранимые процедуры. Некоторые объекты, например сертификаты и асимметричные ключи, могут содержаться в базе данных, но при этом не находиться внутри схемы. Дополнительные сведения о создании таблиц см. в разделе Tables.

Базы данных SQL Server хранятся в файловой системе в виде файлов. Файлы могут быть объединены в группы файлов. Дополнительные сведения о файлах и файловых группах см. в разделе Database Files and Filegroups.

При получении доступа к экземпляру SQL Server пользователи идентифицируются согласно имени входа. При получении доступа к базе данных пользователи идентифицируются как пользователи базы данных. Имя пользователя базы данных может быть основано на имени входа. Если автономные базы данных включены, то пользователь базы данных может быть создан не на основе имени входа. Дополнительные сведения о пользователях см. в статье CREATE USER (Transact-SQL).

Пользователь, имеющий доступ к базе данных, может получить разрешения на доступ к объектам этой базы данных. Хотя разрешения и могут быть предоставлены отдельным пользователям, рекомендуется создавать роли базы данных, добавляя при этом пользователей базы данных к соответствующим ролям, а затем предоставлять разрешения ролям. Предоставление разрешений ролям, а не пользователям позволяет легко и понятно управлять процессом распределения разрешений, несмотря на постоянное изменение и рост числа пользователей. Дополнительные сведения о ролях и разрешениях см. в разделах CREATE ROLE (Transact-SQL) и Субъекты (ядро СУБД).

Git (произносится «гит»[7]) — распределённая система управления версиями. Проект был создан Линусом Торвальдсом для управления разработкой ядра Linux, первая версия выпущена 7 апреля 2005 года. На сегодняшний день его поддерживает Джунио Хамано.

Среди проектов, использующих Git — ядро Linux, Swift, Android, Drupal, Cairo, GNU Core Utilities, Mesa, Wine, Chromium, Compiz Fusion, FlightGear, jQuery, PHP, NASM, MediaWiki, DokuWiki, Qt, ряд дистрибутивов Linux.

Программа является свободной и выпущена под лицензией GNU GPL версии 2. По умолчанию используется TCP порт 9418.

# **ПРАКТИЧЕСКАЯ ЧАСТЬ**

## **Задание 1. Создание дизайна для окна авторизации**

Создал дизайн окна авторизации (рис. 1).

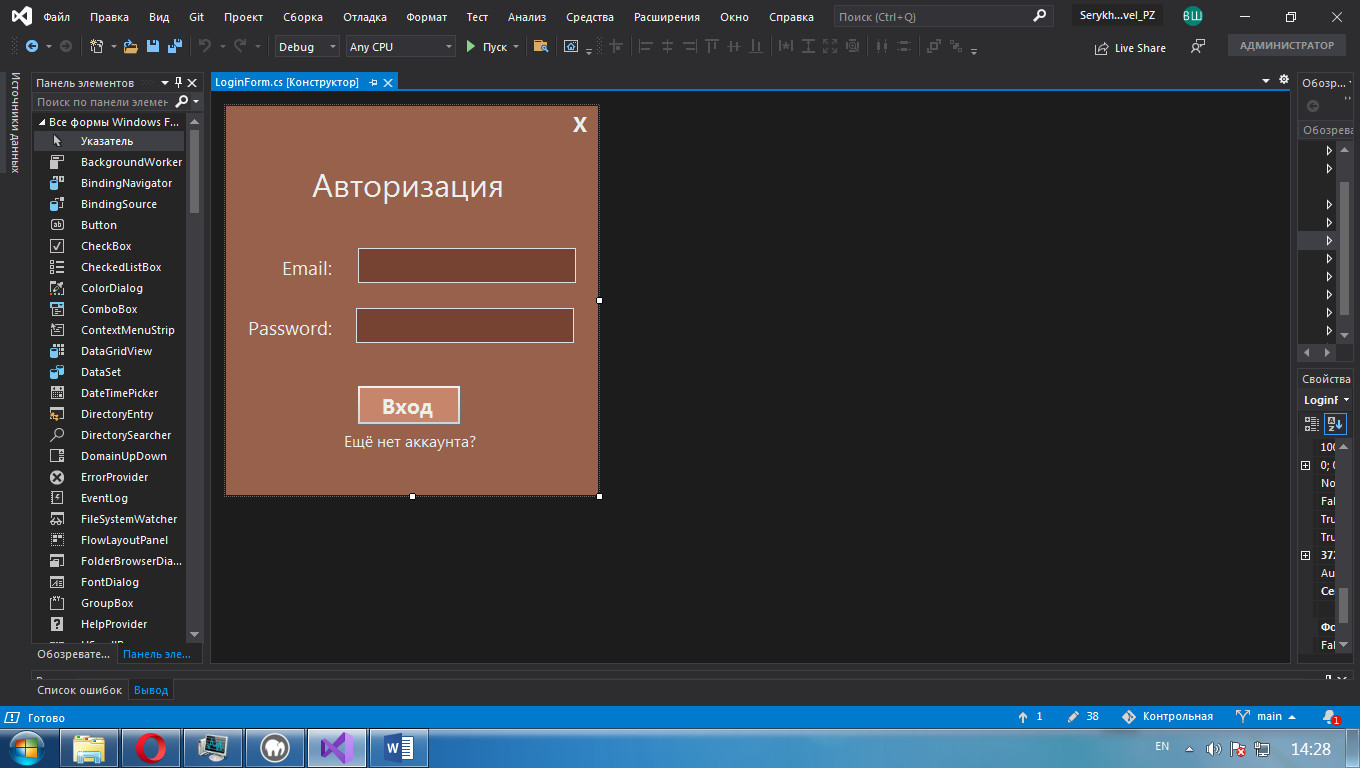


Рисунок 1 – Форма авторизации

## **Задание 2. Подключение MySQL и создание базы данных**

Создал БД с помощью phpMyAdmin (рис. 2).

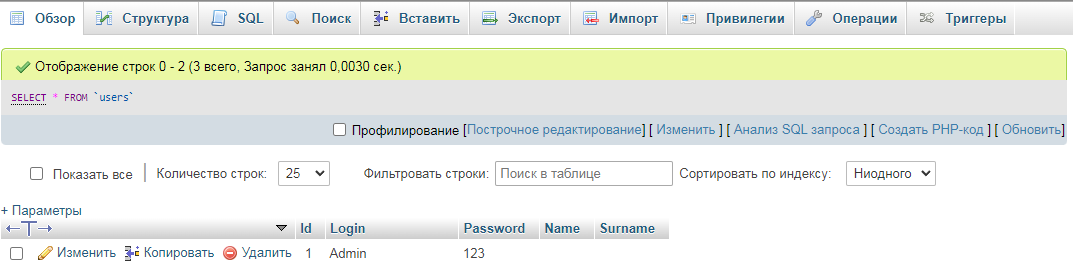


Рисунок 2 – Таблица users

Написал класс для взаимодействия с БД (рис. 3).

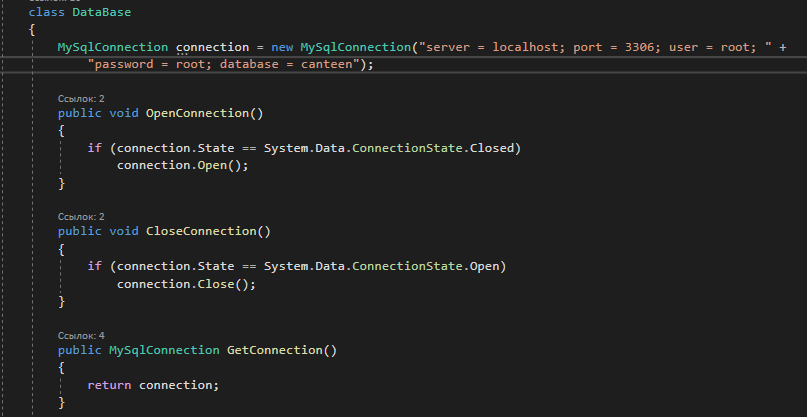


Рисунок 3 – Класс DataBase

## **Задание 3. Авторизация пользователя через базу данных**

Добавил код для события нажатия на кнопку «Войти» на форме «Авторизация», который проверят совподают ли введеные данные с данными в БД (рис. 4).



Рисунок 4 – Код события Click у кнопки Enter

## **Задание 4. Окно регистрации пользователя**

Создал окно авторизации (рис 5).

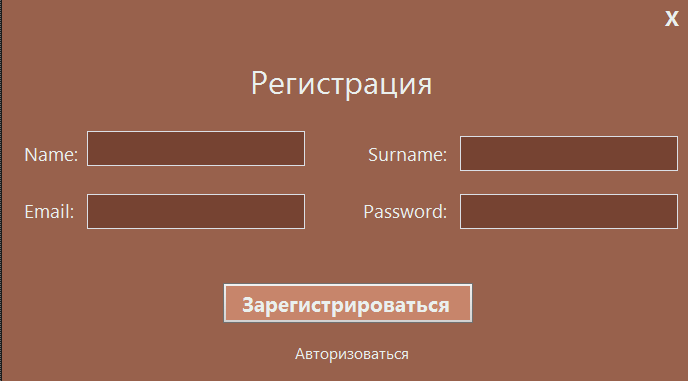


Рисунок 5 – Форма регистрации

## **Задание 5. Регистрация пользователя в базе данных**

Добавил возможность зарегестрировать пользователя, после нажатия на кнопку «Зарегистрироваться», в БД при заполнении всех полей и отсутсвии пользователя с такой же почтой (рис. 6).

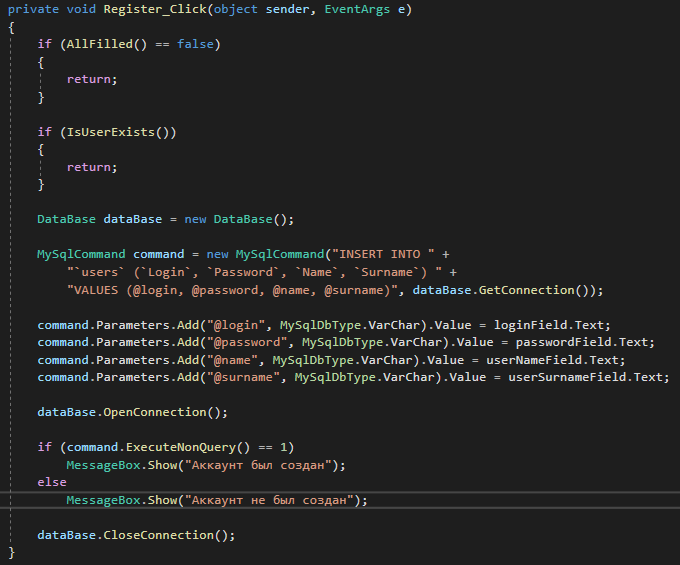


Рисунок 6 – Код события Click у кнопки Register

## **Задание 6. Переход между окнами в приложении**

Реалезовал переход между формами регистрация, авторизация и главной формой. (рис. 7).

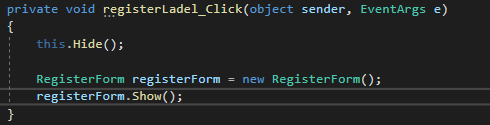


Рисунок 7 – Переход на страницу регистрации

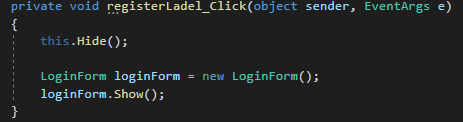


Рисунок 8 – Переход на страницу авторизации

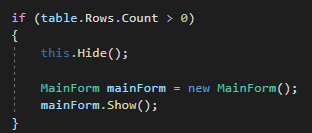


Рисунок 9 – Переход на главную форму

## **Задание 7. Создать в БД таблицу, по вашей теме, чтобы она выводилась на форме "Главный экран", после авторизации.**

Создал главную форму из который можно открыть 4 таблицы из моего задания (рис. 10).

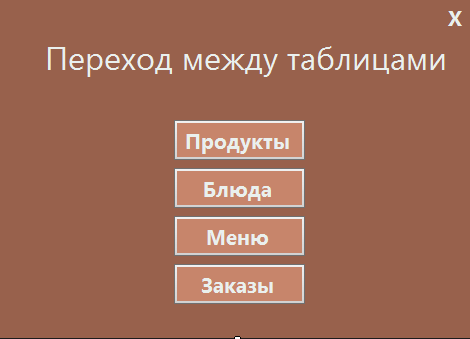


Рисунок 10 – Форма для перехода между таблицами

Написал код для кнопок перехода (рис. 11).

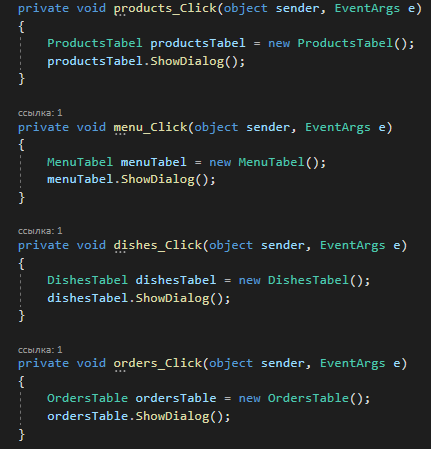


Рисунок 11 – Код кнопок перехода

Создал 4 формы с таблицими (рис. 12 - 15).

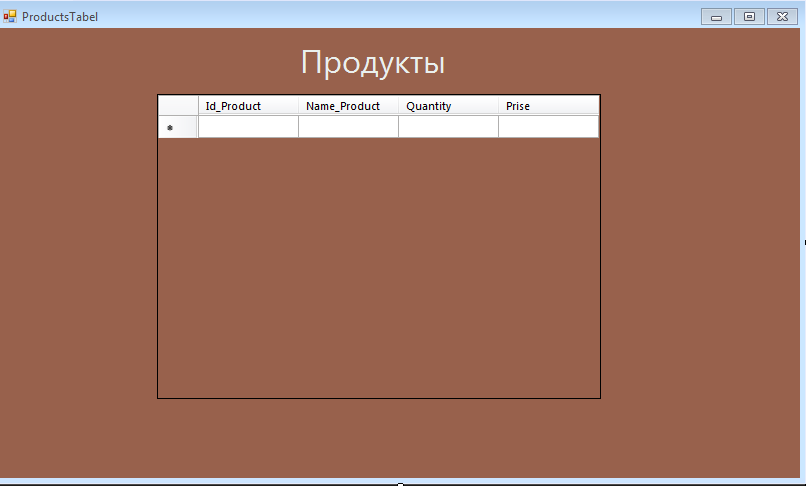


Рисунок 12 – Таблица «Продукты»

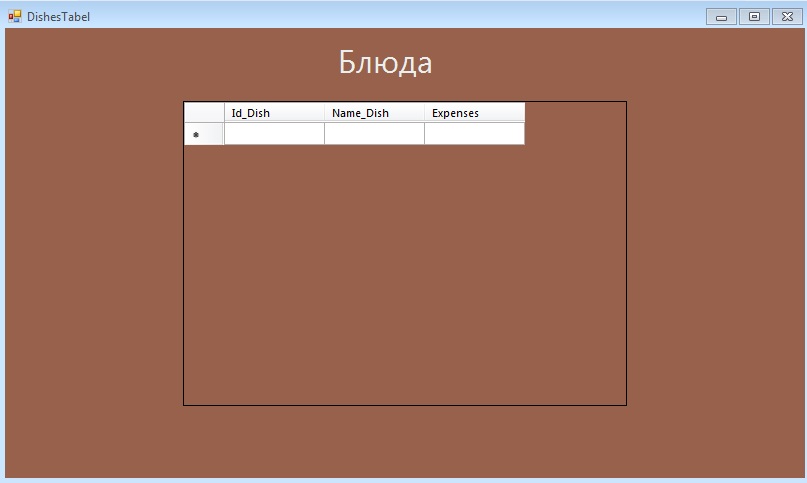


Рисунок 13 – Таблица «Блюда»

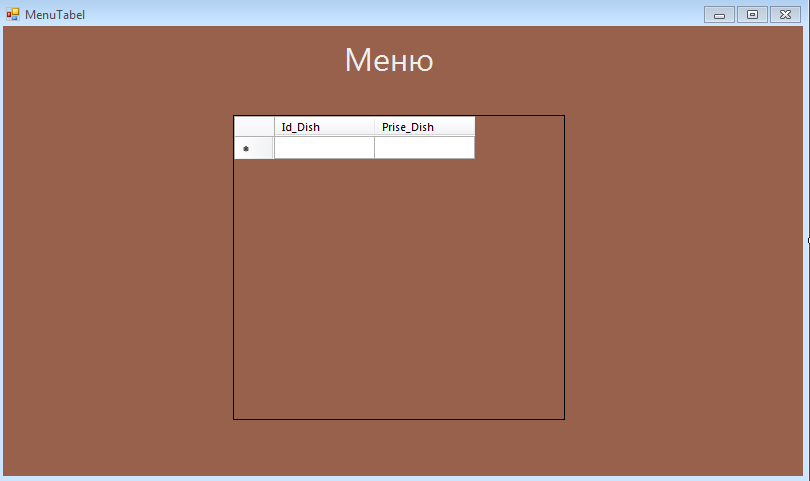


Рисунок 14 – Таблица «Меню»

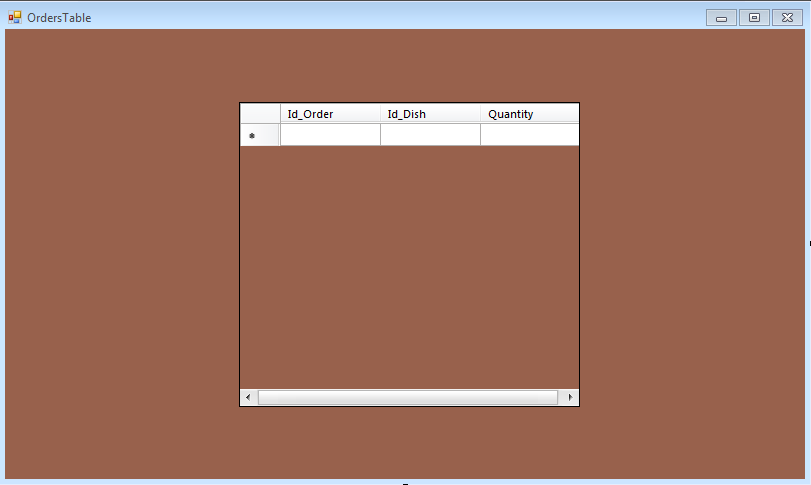


Рисунок 15 – Таблица «Заказы»

Написал метод LoadData в классе DataBase для заполнения этих таблиц который в качестве параметров получает текст запроса и объект DataGridView (рис. 16).

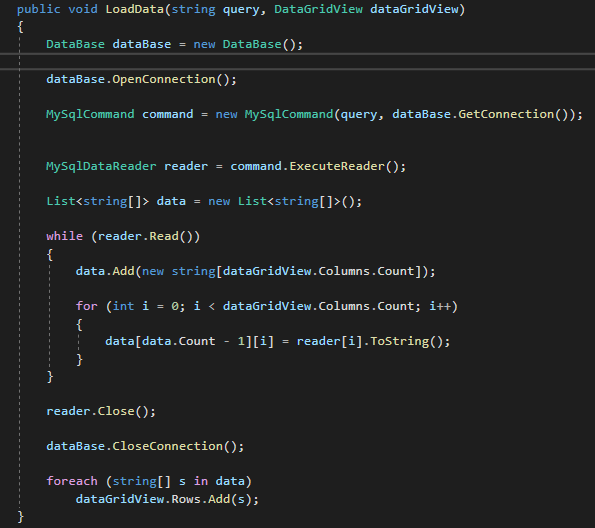


Рисунок 16 – Метод LoadData

Проверил программу на работаспособность (рис. 16).

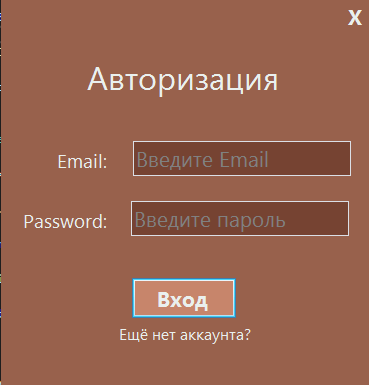


Рисунок 17 – Форма авторизации

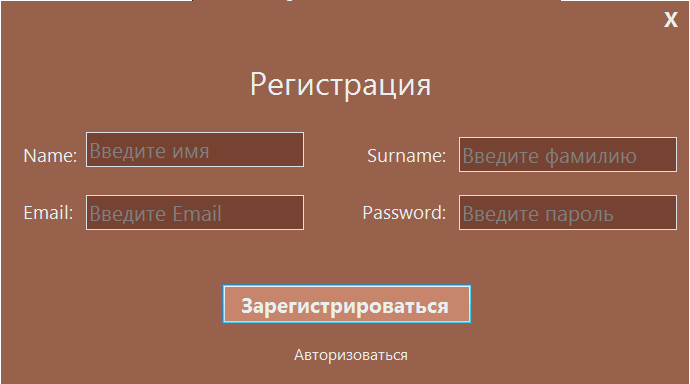


Рисунок 18 – Форма регистрации

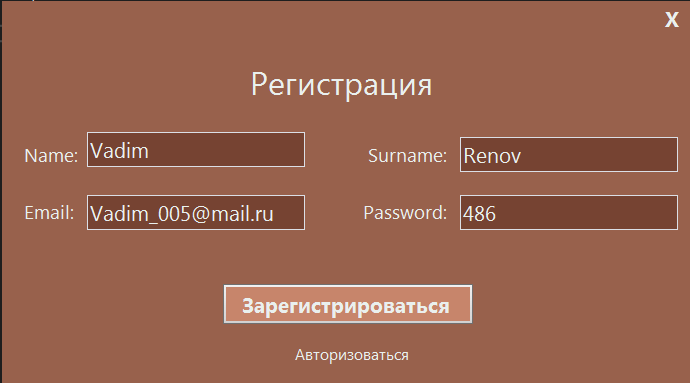


Рисунок 19 – Регистрациия в БД(1)

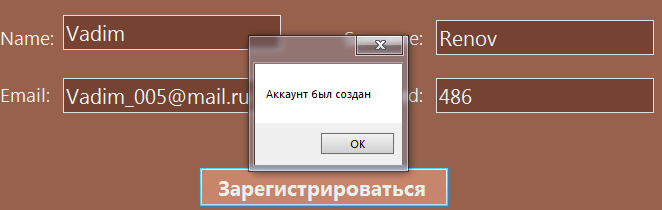


Рисунок 20 – Регистрациия в БД(2)



Рисунок 21 – Регистрациия в БД(3)

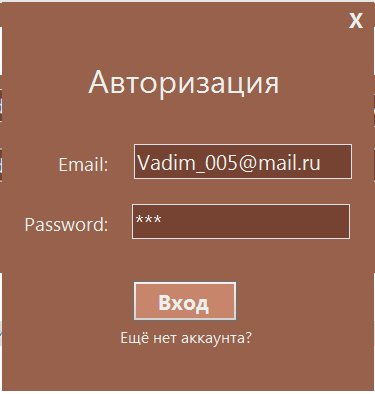


Рисунок 22 – Авторизация

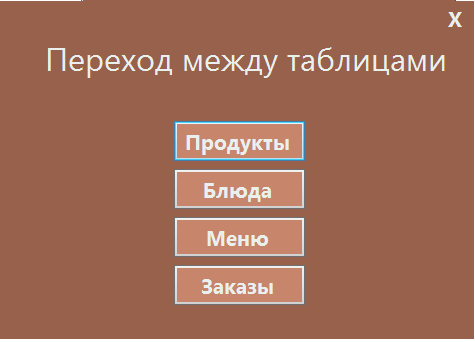


Рисунок 23 – Главная форма

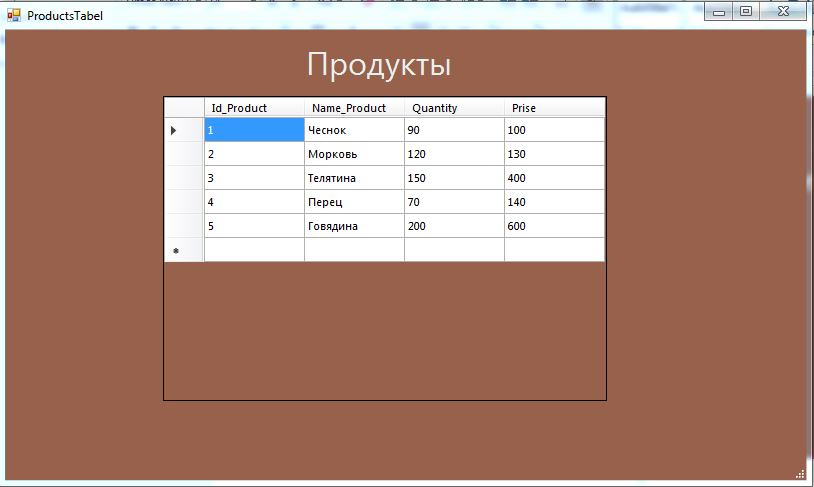


Рисунок 24 – Таблица «Продукты»

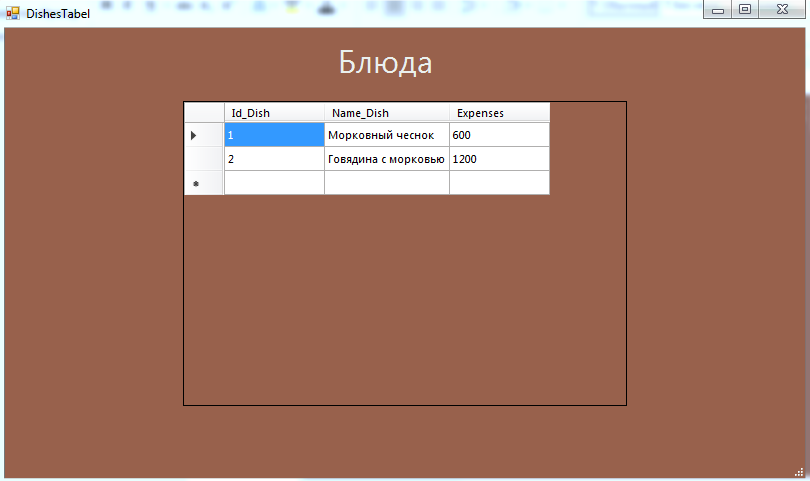


Рисунок 25 – Таблица «Блюда»

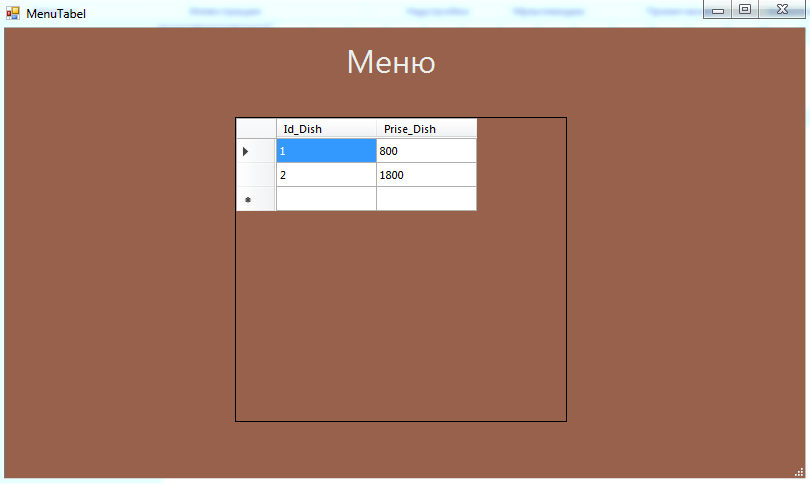


Рисунок 26 – Таблица «Меню»

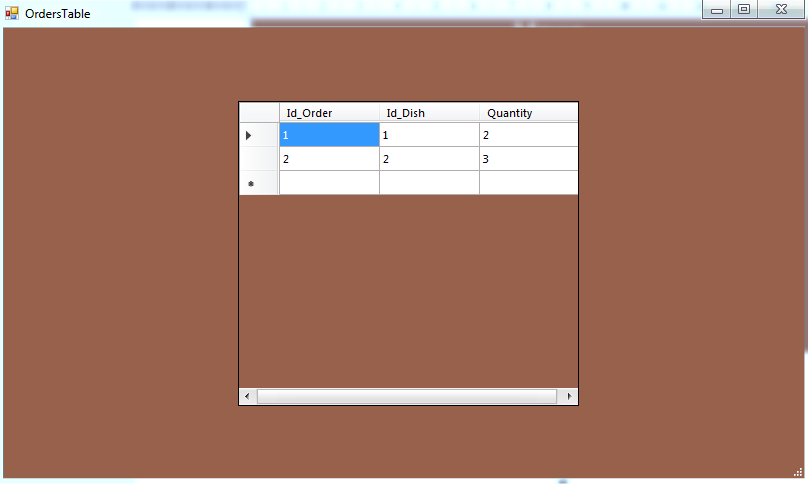


Рисунок 27 – Таблица «Заказы»

# **ЗАКЛЮЧЕНИЕ**

В процессе выполнения работы была изучена система управления базами данных MySQL, ее основные особенности.

Изучая теоретический материал, мы выявили особенности СУБД MySQL, относящиеся к различным аспектам функционирования системы: физическая организация данных, синтаксис языка, особенности используемых операторов. Кроме того, мы определили все виды целостности данных - как семантической, так и ссылочной - и отметили, каким образом в СУБД MySQL реализуются те или иные ограничения. В целом, рассматривая система управления базами данных представляет собой развитое, многообразное сетевое средство создания и использования БД, различных по своему составу, объему и функциональному назначению. В СУБД MySQL реализованы механизмы, необходимые для обеспечения целостности данных: установление формата данных в атрибутах, определение характера связей в реляционных отношениях, назначение атрибута первичным и внешним ключом.

В практической части работы на примере небольшого фрагмента информационной системы «Салон магии» нами были поставлены информационные задачи, возникающие при использовании ИС, на их основе сформулированы все виды ограничений целостности, реализованные впоследствии средствами СУБД MySQL.

Так же был изучен ЯП C#, который является хорошим выбором для быстрого конструирования различных компонентов - от высокоуровневой бизнес логики до системных приложений, использующих низкоуровневый код. Также следует отметить, что C# является и Web ориентированным - используя простые встроенные конструкции языка ваши компоненты могут быть легко превращены в Web сервисы, к которым можно будет обращаться из Internet посредством любого языка на любой операционной системе. Дополнительные возможности и преимущества перед другими языками приносит в C# использование передовых Web технологий, таких как: XML (Extensible Markup Language) и SOAP (Simple Object Access Protocol). Среда разработки Web сервисов позволяет программисту смотреть на существующие сегодня Web приложения, как на родные C# объекты, что дает возможность разработчикам соотнести имеющиеся Web сервисы с их познаниями в объектно-ориентированном программировании.

# **СПИСОК ИСПОЛЬЗОВАНЫХ ИСТОЧНИКОВ**

1. Вывод данных в DataGridView из БД на C#: сайт. – 2015. – URL: https://vscode.ru/prog-lessons/vyivod-dannyih-v-datagridview-c-sharp.html (дата обращения: 10.04.2021).
2. Создание базы данных mysql в phpmyadmin: сайт. – 2014. – URL: https://sozdatisite.ru/mysql/sozdanie-bazy-dannyh-mysql-v-phpmyadmin.html (дата обращения: 01.09.2020).
3. Введение - Установка Git: сайт. – 2014. –URL https://git-scm.com/book/ru/v2/Введение-Установка-Git (дата обращения: 10.04.2021).
4. MySQL connection with C# through PHPMyAdmin created database [closed]: сайт. – 2022. – URL: https://stackoverflow.com/questions/24120497/mysql-connection-with-c-sharp-through-phpmyadmin-created-database (дата обращения: 10.04.2021).
5. Управление таблицами в PHPMyAdmin: сайт. – 2010. – URL: https://myrusakov.ru/phpmyadmin-table.html (дата обращения: 10.04.2021).