

Министерство образования, науки и молодежной политики
Краснодарского края
Государственное бюджетное профессиональное образовательное учреждение
Краснодарского края «Ейский полипрофильный колледж»

Отделение педагогики
Специальность 09.02.07 Информационные системы и программирование

Студента И-32 группы
САМСЫКА РОДИОНА МИХАЙЛОВИЧА

РАЗРАБОТКА МОДУЛЕЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ
ОФОРМЛЕНИЯ ЗАКАЗОВ НА ПРЕДПРИЯТИИ ОБЩЕСТВЕННОГО
ПИТАНИЯ

Научный руководитель
преподаватель специальных
дисциплин
Фомин А. Т.

г. Ейск
2024 г.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	3
ГЛАВА 1 ПРОЕКТИРОВАНИЕ ПРИЛОЖЕНИЯ.....	5
1.1 Описание предметной области.....	5
1.2 Описание назначения ПС.....	5
1.3 Анализ требований к ПС.....	5
ГЛАВА 2 РАЗРАБОТКА ПРИЛОЖЕНИЯ.....	9
2.1 Разработка информационной модели.....	9
2.2 Разработка программных модулей.....	11
2.2.1 Интеграция программных модулей.....	17
2.3 Описание интерфейса.....	18
2.4 Инструкция пользователя.....	18
ЗАКЛЮЧЕНИЕ.....	21
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	22
ПРИЛОЖЕНИЕ А.....	24
ПРИЛОЖЕНИЕ Б.....	75

ВВЕДЕНИЕ

В больших мегаполисах существует острая конкуренция среди предприятий общественного питания. Среди востребованных в сфере общественного питания является ресторан, который осуществляет обслуживание предоставляя клиентам меню.

Тема проекта: разработка модулей информационной системы оформления заказов на предприятии общественного питания.

Актуальность

В ресторанах обслуживается большой поток клиентов и предоставляется обширное меню. Процесс должен быть быстрее а форма должна предлагать гибкие способы выбора позиции в меню. Поэтому автоматизация этого процесса будет актуальна.

Объект: предметная область — ресторан.

Предмет: автоматизация процесса оформления заказа.

Цель работы – разработка модулей информационной системы оформления заказа предметной области ресторан.

Задачи:

Для достижения поставленной цели курсового проекта необходимо решить следующие задачи:

1. Описание предметной области;
2. Описание назначение ПС;
3. Анализ требований к ПС;
4. Выбор технологии программирования;
5. Выбор среды программирования;
6. Создание информационной модели данных;
7. Разработка программных модулей;
8. Интеграция программных модулей;
9. Описание интерфейса приложения;

Структура работы: курсовой проект состоит из введения, двух глав, заключения, списка литературы. В первой главе описано проектирование системы для оформления заказов. Вторая глава посвящена описанию процесса разработки системы оформления заказов.

ГЛАВА 1 ПРОЕКТИРОВАНИЕ ПРИЛОЖЕНИЯ

1.1 Описание предметной области

Ресторан — это заведение, предоставляющее услуги питания с возможностью обслуживания клиентов на месте. Он играет важную роль в сфере общественного питания, предлагая разнообразные блюда и напитки, которые могут варьироваться от легких закусок до полноценного обеденного меню. Рестораны могут отличаться по стилю, концепции, уровню цен и качеству обслуживания, а также специализироваться на определенной кухне или блюдах.

1.2 Описание назначения ПС

ПС предназначена для официантов, чтобы сделать процесс оформления заказа клиента более удобным и интуитивно понятным для новых сотрудников. Это позволяет сократить время на обучение и повысить качество обслуживания.

1.3 Анализ требований к ПС

Функциональные требования:

— Программное обеспечение (ПС) должно обладать простым и интуитивно понятным интерфейсом, который позволит официантам быстро и легко оформлять заказы, минимизируя количество необходимых действий. Это обеспечит высокую эффективность работы и удовлетворит потребности клиентов.

— ПС должна предоставлять возможность просмотра актуального меню с описанием блюд, их ценами, чтобы официанты могли предоставлять актуальную информацию клиентам.

Требования к структуре ПС:

— Легкость в расширении функционала (например, добавление новых категорий блюд, интеграция с системами управления запасами).

Требования к надежности:

— В случае возникновения ошибок, система должна обеспечивать корректное информирование пользователей о произошедших проблемах и предоставлять рекомендации по их устранению;

— Обработка исключений с непрерывной работой.

Условия эксплуатации:

— Квалификация персонала: Минимум один сотрудник с базовыми навыками работы на компьютерах; рекомендуется предоставить обучение;

— Персонал: Ожидается, что от 2 до 10 сотрудников будут работать с системой одновременно в часы пик.

Требования к составу и параметрам технических средств:

- Минимальные системные требования к настольным компьютерам: операционная система Windows, процессор не ниже 2.0 ГГц, 4 ГБ ОЗУ, 500 МБ свободного места на жестком диске.

Требования к информационной и программной совместимости:

- Язык программирования: C#;
- Поддержка работы с реляционными СУБД.

Требования к транспортировке и хранению:

- Система должна быть предоставлена в виде программного продукта на физическом носителе или доступна для загрузки в интернете;
- Данные должны храниться на защищенном сервере с регулярным бэкапом.

Специальные требования:

- Интерфейс программы должен быть интуитивно понятным и доступным для обучающейся команды, что позволит новым сотрудникам быстро освоить программу без необходимости в длительном обучении;
- Система должна обеспечивать быструю обработку заказов, чтобы минимизировать время ожидания клиентов и улучшить общую производительность работы ресторана;
- Система должна обеспечивать быстрое время отклика на запросы пользователей, даже в условиях высокой нагрузки.

Технологии программирования – Среда: Visual Studio на Windows,
Язык: C#[12][8], **Компилятор:** Roslyn, **Инструменты:** Windows Forms, **Стиль программирования:** ООП [5].

Вывод

Была проанализирована предметная область, выбран процесс оформления заказов. Поставлена цель и задачи для разработки ПС.

ГЛАВА 2 РАЗРАБОТКА ПРИЛОЖЕНИЯ

2.1 Разработка информационной модели

После исследования предметной области и выбранного процесса, было принято решение об использовании базы данных основанной на схеме из (рисунка 1). Ключевым и связующим элементом является заказанная еда, так как таблица еды представляет список того, что можно приготовить, а заказ включает в себя уже заказанную еду.

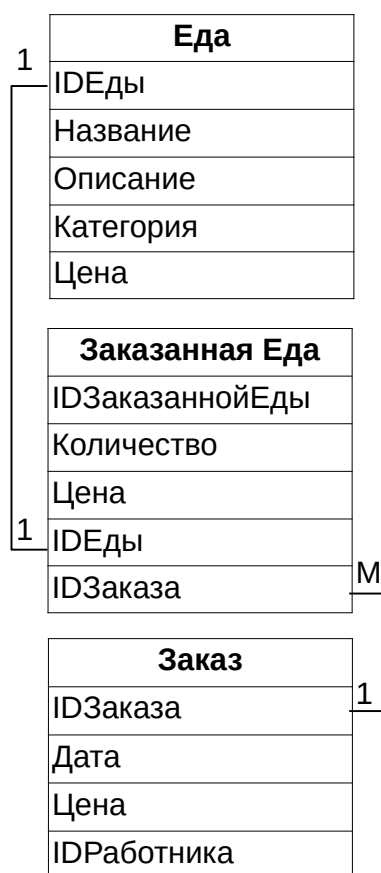


Рисунок 1 — Схема базы данных

Программа будет разработана по виду диаграммы классов[15] из (рисунка 2). В которой описаны классы и методы этих классов, а так же интерфейсы и реализующие их классы.

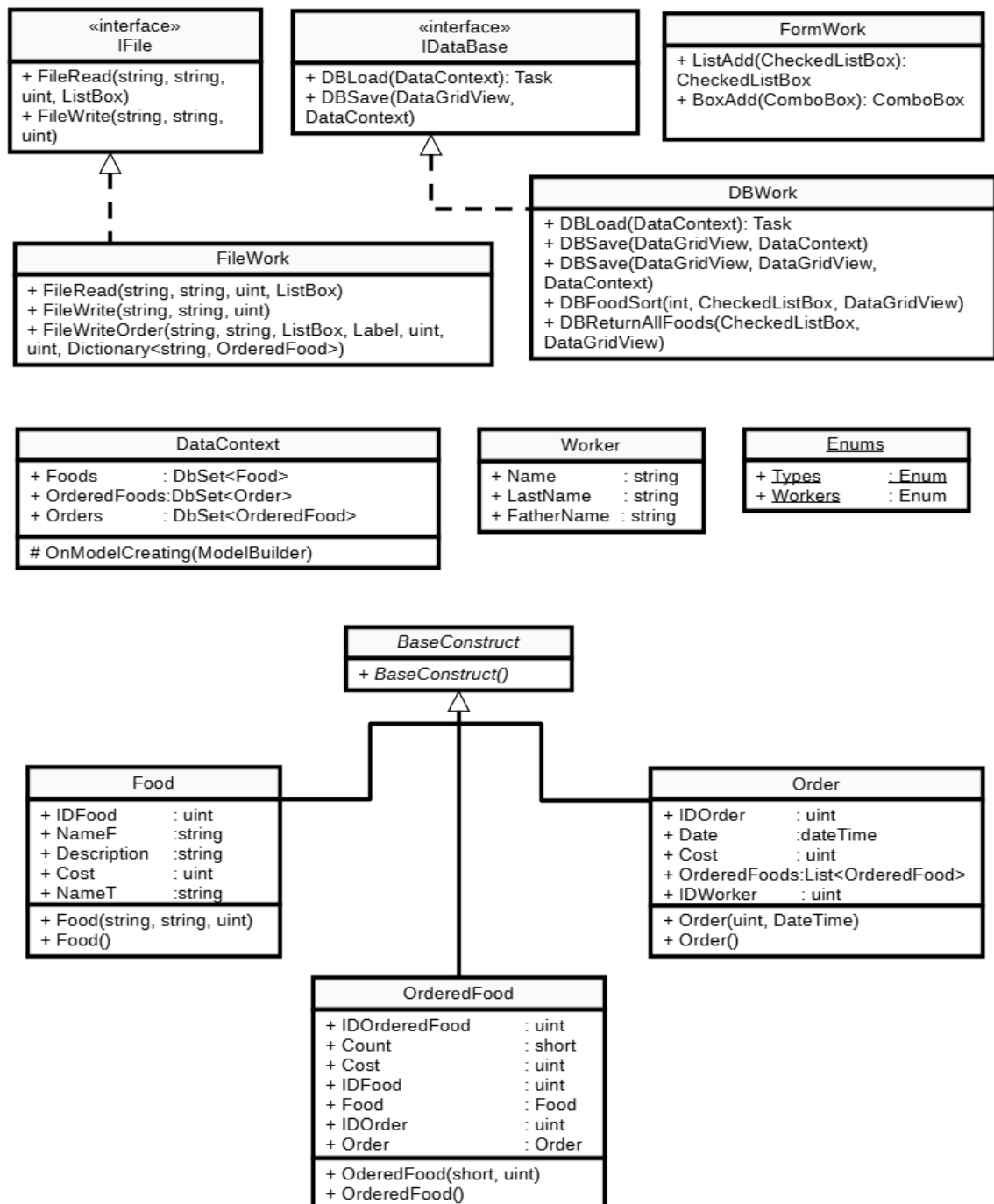


Рисунок 2 — Диаграмма классов

2.2 Разработка программных модулей

Метод DBFoodSort класса DBWork

Этот метод сортирует меню в зависимости от выбранного пользователем типа еды (Напиток, десерт, закуска и т. д.). Его принцип работы проиллюстрирован на (рисунке 3). Выбранный тип это элементы из `CheckedListBox` которые были выбраны. После берётся имя этих элементов и сравнивается с полем из базы данных, и если совпадает то эта запись из базы данных выводится, иначе пропускается.

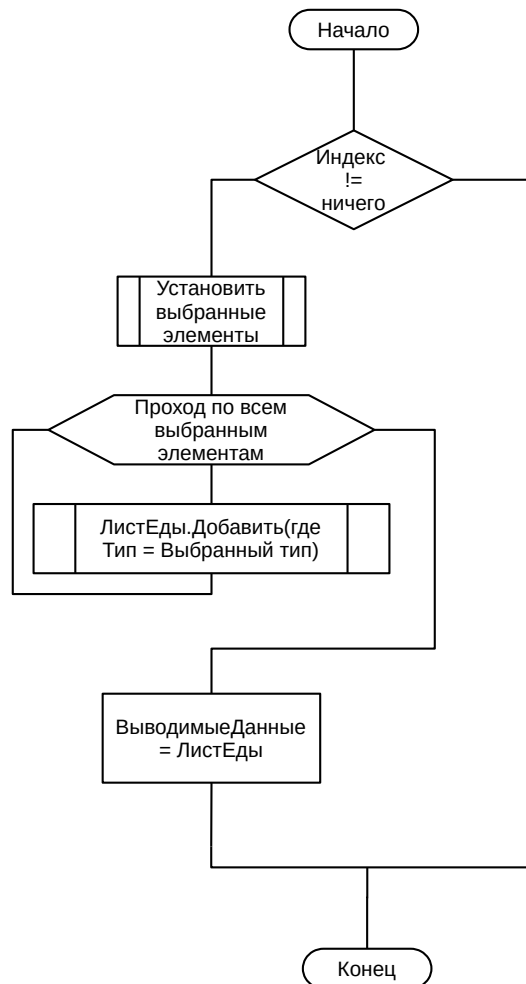


Рисунок 1 — Сортировка еды в списке

Метод DBReturnAllFoods класса DBWork

Этот метод противоположен DBFoodSort, он убирает все фильтры и выводит все записи из базы данных, то есть, отображает всё меню. Его работа проиллюстрирована на (рисунке 4).

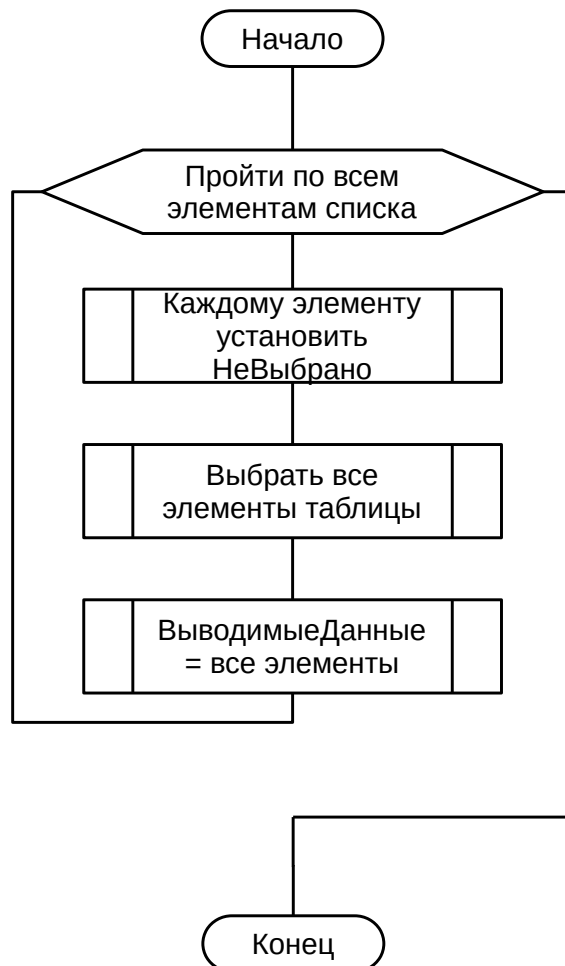


Рисунок 2 — Возврат всей еды в лист

Метод *FileWriteOrder* класса *FileWork*

Первая часть метода (рисунок 5) записывает информацию в базу данных. А именно заказ и заказанную еду. Вся информация для записи берётся из формы.

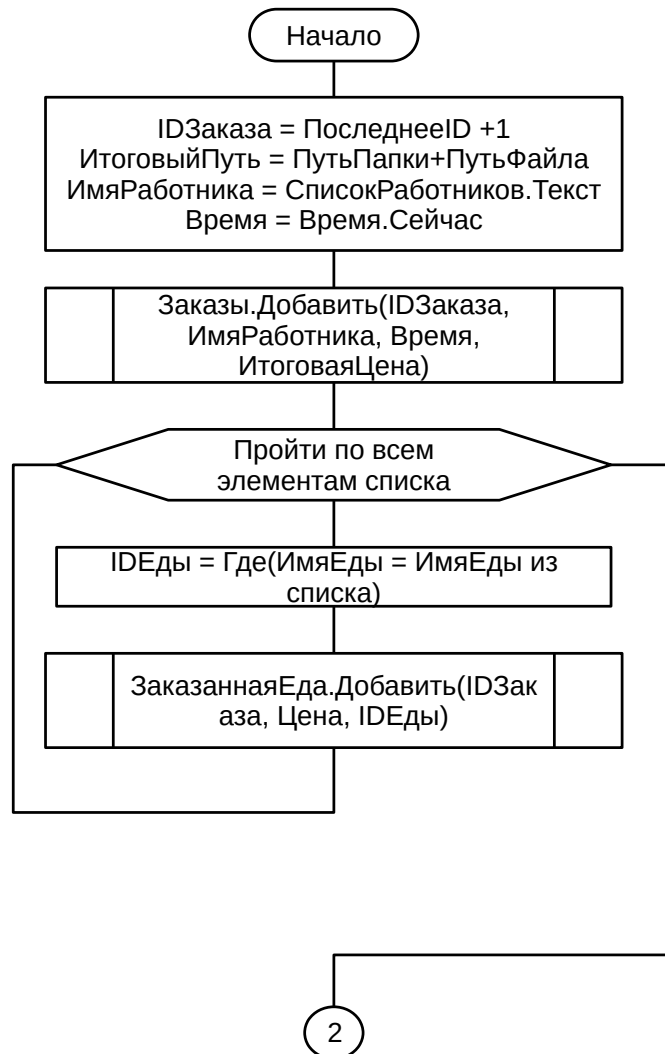


Рисунок 3 — Запись заказа в файл 1

Вторая часть этого метода (рисунок 6) записывает информацию о заказе в файл. Метод FileWrite схож с этим, но он берёт уже сформированную информацию из базы данных, а не формирует её из полей формы.

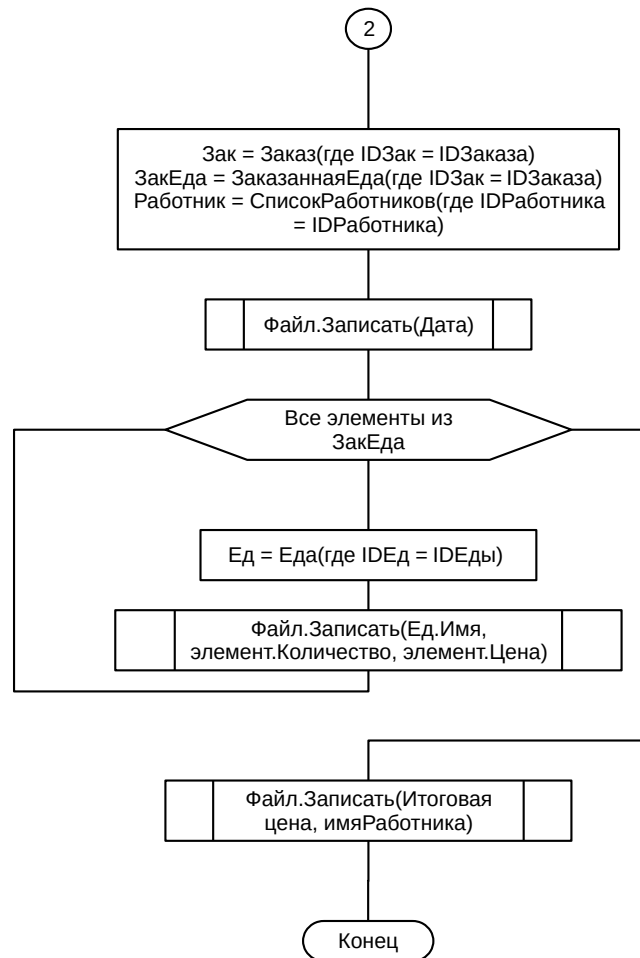


Рисунок 4 — Запись заказа в файл 2

Более кратко обо всех методах можно узнать из приведённой ниже таблицы 1. Остальные части программы представлены в таблицах 2 и 3.

Таблица 1 — Классы и методы в них

Класс	Метод	Описание	Приложение
FormWork	ListAdd()	Добавление в передаваемый лист элементы из перечисления	A 8
	BoxAdd()	Добавление в передаваемый всплывающий список элементы из перечисления	
FileWork	FileRead()	Чтение строк из файла и запись их в текстовое поле	A 7
	FileWrite()	Запись в файл информации из базы данных	
	FileWriteOrder()	Запись информации в базу данных и файл, извлекая её из полей формы	
DBWork	DBLoad()	Загрузка ресурсов базы данных для формы	A 6
	DBSave()	Сохранение изменений для базы данных, имеет перегрузку если изменения в двух таблицах	
	DBFoodSort()	Устанавливает фильтр для выводимой еды в виде её типов	
	DBReturnAllFoods()	Убирает все фильтры с выводимой еды	

Таблица 2 — Интерфейсы и классы родители

Класс/интерфейс	Поля/методы	Наследники	Приложение
IFile	FileWrite() FileRead()	FileWork	A 10
IDataBase	DBLoad() DBSave	DBWork	A 10
BaseConstruct	BaseConstruct()	Food, Order, OrderedFood	A 11

Таблица 3 — Остальные классы

Класс	Поля/методы	Описание	Приложение
DBContext	OnModel()	Заполнение полей таблиц заранее прописанными значениями	А 9
	Foods	Список для связи с таблицей еды	
	Orders	Список для связи с таблицей заказов	
	OrderedFoods	Список для связи с таблицей заказанной еды	
Enums	Types	Перечисление типов еды	А 11
	Workers	Перечисление сотрудников	
Worker	Name	Имя сотрудника	А 11
	LastName	Фамилия сотрудника	
	FatherName	Отчество сотрудника	
Food	IDFood	Номер записи в таблице Еда	А 11
	NameF	Название еды	
	Description	Описание еды	
	Cost	Цена за порцию	
	NameT	Название типа из перечисления	
	Food()	Конструктор класса	
Order	IDOrder	Номер записи в таблице Заказ	А 11
	Data	Дата оформления	
	Cost	Итоговая цена	
	OrderedFoods	Лист заказанной еды (для связи)	
	IDWorker	Номер сотрудника оформившего заказ	
	Order()	Конструктор класса	
OrderedFood	IDOrderedFood	Номер записи в таблице Заказанная Еда	А 11

	Count	Количество порций	
	Cost	Цена	
	IDFood	Номер еды	
	Food	Экземпляр класса еды (для связи)	
	IDOrder	Номер заказа	
	Order	Экземпляр класса заказа (для связи)	
	OrderedFood()	Конструктор класса	

2.2.1 Интеграция программных модулей

Linq [7]

используется для написания запросов к коллекциям

EntityFrameworkCore [9][13]

Используется для работы с базой данных используя код на с#

Collections.Generic

используется для доступа к коллекциям

Threading.Tasks [6] [10]

используется для оптимизации программы с помощью многозадачности

Windows.Forms [14]

Используется для разработки пользовательского интерфейса

System.IO [11]

Используется для работы с файлами

Microsoft.Data.Sqlite

Используется для написания запросов и работы с базой данных

System.Data

Используется для работы с датой и временем

2.3 Описание интерфейса

Интерфейс представляет собой пустую форму с меню инструментов сверху (Рисунок Б.1), при нажатии на кнопки которого открываются дочерние формы, что позволяет открыть сразу несколько окон. В панели присутствуют два пункта «официант» и «администратор», у каждого есть свои подпункты, что даёт чёткое разделение полномочий и выполняемых функций.

2.4 Инструкция пользователя

Для официанта:

1.Просмотр меню

Для открытия нужно выбрать пункт «Официант» и подпункт «Просмотреть меню» (Рисунок Б.2). Слева выбирается тип еды для сортировки записей в правом окне. При нажатии кнопки «Всё» снимутся все выделения и фильтры, после чего выведется полное меню.

2.Оформить заказ

Для открытия нужно выбрать пункт «Официант» и подпункт «Оформить заказ» (Рисунок Б.3). Для оформления нужно: нажать на нужное блюдо (для удобства представлен способ фильтрации по типу); ввести количество порций в текстовом поле; в выплывающем списке выбрать официанта который обслуживает в данный момент; нажать кнопку «Добавить к заказу»; если требуется удалить выбираем ненужный заказ из списка и жмём «Удалить из заказа»; Когда заказ готов к оформлению жмём кнопку «Оформить» справа внизу. Приложение предложит открыть файл с информацией о заказе, если выбрать «да» откроется файл.

Для администратора:

1.Редактирование меню

Для открытия нужно выбрать пункт «Администратор» и подпункт «Меню» (Рисунок Б.4). Почти на всё свободное место представлен список продуктов. Для добавления нужно пролистать в самый низ списка и нажать на свободную строку с символом «*» в начале, после чего ввести нужные данные и нажать кнопку «Сохранить изменения» (эта кнопка сохраняет все изменения в базу данных). Для удаления нужно выбрать ненужную запись и нажать кнопку «Удалить продукт». Для редактирования два раза кликнуть по редактируемому полю. После любых изменений сохранять нажатием на «Сохранить изменения»

2.Редактирование заказов

Для открытия нужно выбрать пункт «Администратор» и подпункт «Заказы» (Рисунок Б.5). Слева представлен список заказов а справа блюда, включённые в этот заказ. Кнопка «Удалить Заказ» удаляет выбранный заказ из списка, соответствующие ему продукты так же удаляются. Кнопка «Удалить Продукт» удаляет выбранный продукт из списка справа. Кнопка «Сохранить» сохраняет все изменения в базу данных. Кнопка «Вывести текст» выводит текст

в список справа из файла, соответствующего выбранному заказу. Кнопка «Сохранить чек» сохраняет или изменяет файл чека соответствующий выбранному заказу. Кнопка «Открыть папку с чеками» откроет в проводнике папку в которой находятся все файлы.

ЗАКЛЮЧЕНИЕ

Выполнение курсового проекта было направлено на разработку информационной системы для предметной области ресторан, что предполагало осуществление нескольких задач.

Для этого было необходимо рассмотреть особенности предметной области ресторан, провести анализ процесса оформления заказа, обеспечить создание необходимых функций.

Все этапы по реализации данных задач были выполнены, и поставленная цель была достигнута.

В ходе работы над курсовым проектом применялись знания и умения, полученные в ходе изучения учебных тем: Windows Forms, Entity Framework, SQLite, асинхронное программирование.

Разработанная информационная система отражает знания из предметной области, рассмотренной в курсовом проекте. Система содержит необходимые объекты, функциональность которых описана в пояснительной записке.

Таким образом, работа над созданием информационной системы предметной области ресторан была завершена, что является достижением поставленной цели.

Курсовой проект основывается на условном примере, который не отражает всех особенностей предметной области ресторан, поэтому в данной работе мы говорим о разработке модели информационной системы. Однако выполнение данного курсового проекта позволило ознакомиться и научиться применять необходимые инструменты разработки информационных систем, которые могут быть использованы для реальной предметной области.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Постановление Правительства Российской Федерации от 30.09.2020 N 1597 "Об утверждении требований к электронным образовательным ресурсам". – М., 2020.
2. Постановление Правительства Российской Федерации от 1 августа 2020 г. N 1169 "Об утверждении правил разработки и утверждения образовательных стандартов". – М., 2020.
3. Федеральный закон от 29.06.2015 N 162-ФЗ "Об информации, информационных технологиях и о защите информации". – М., 2015.
4. Федеральный закон от 27.07.2006 N 152-ФЗ "О персональных данных". – М., 2006.
5. А.Ю. Беляков. Объектно-ориентированное программирование. – Пермь: Изд-во ФГБОУ ВПО Пермская ГСХА, 2017. – 87 с.
6. Алекс Дэвис. Асинхронное программирование в C# 5.0. / Пер. с англ. Слинкин А. А. – М.: ДМК Пресс, 2013. – 120 с.
7. Фримен, Адам, Раттц-мл., Джозеф. язык интегрированных запросов в C# 2010 для профессионалов. – М. : ООО "И.Д. Вильямс", 2011. — 656 с.
8. Прайс, Марк. C# 9 и .NET 5. Разработка и оптимизация. — СПб.: Питер, 2022. — 832 с.
9. Microsoft. Entity Framework Core Documentation. – URL:
<https://docs.microsoft.com/en-us/ef/core/> (дата обращения: 27.10.2024).
10. Microsoft. Asynchronous Programming with Async and Await. – URL:
<https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/concepts/async/> (дата обращения: 27.10.2024).
11. Microsoft. File Класс. – URL:
<https://learn.microsoft.com/ru-ru/dotnet/api/system.io.file?view=net-8.0> (дата обращения: 27.10.2024).

12. Metanit. C# (C Sharp). – URL: <https://metanit.com/sharp/tutorial/> (дата обращения: 25.10.2024).
13. Metanit. Entity Framework Core. – URL: <https://metanit.com/sharp/entityframeworkcore/> (дата обращения: 27.10.2024).
14. Metanit. Windows Forms. – URL: <https://metanit.com/sharp/windowsforms/> (дата обращения: 25.10.2024).
15. Хабр. Использование диаграммы классов UML при проектировании и документировании программного обеспечения. – URL: <https://habr.com/ru/articles/572234/> (дата обращения: 14.11.2024).

ПРИЛОЖЕНИЕ А

Листинг 1 — Form1.cs

```
1  using Microsoft.EntityFrameworkCore;
2  using Restauran;
3  using Restauran.cs;
4  using System;
5  using System.Collections.Generic;
6  using System.ComponentModel;
7  using System.Data;
8  using System.Drawing;
9  using System.Linq;
10 using System.Text;
11 using System.Threading.Tasks;
12 using System.Windows.Forms;
13 using System.Diagnostics;
14
15 namespace restauran
16 {
17     public partial class Form1 : Form
18     {
19         Stopwatch stopwatch = new Stopwatch();
20         public Form1()
21         {
22             InitializeComponent();
23         }
24     }
```



```

25         private void
        менюToolStripMenuItem_Click(object sender, EventArgs
        e)
26     {
27         stopwatch.Restart();
28         ModerMenu newChild = new();
29         newChild.MdiParent = this;
30         newChild.Show();
31         stopwatch.Stop();
32         MessageBox.Show($"Time
        {stopwatch.ElapsedMilliseconds}ms");
33     }
34
35     private void Form1_Load(object sender,
        EventArgs e)
36     {
37
38     }
39     private DataContext? dbContext;
40     protected override void OnLoad(EventArgs e)
41     {
42         base.OnLoad(e);
43         this.dbContext = new DataContext();
44         // Uncomment the line below to start
        fresh with a new database.

```

```

45      //this.dbContext.Database.EnsureCreated();
46          //this.dbContext.Foods.Load();
47          //this.dbContext.OrderedFoods.Load();
48          //this.dbContext.FoodTypes.Load();
49          //this.dbContext.Workers.Load();
50          //this.dbContext.Orders.Load();
51          //this.foodTypeBindingSource.DataSource
    = dbContext.FoodTypes.Local.ToBindingList();
52          //this.foodsBindingSource.DataSource =
    dbContext.Foods.Local.ToBindingList();
53      }
54
55      protected override void
    OnClosing(CancelEventArgs e)
56      {
57          base.OnClosing(e);
58          this.dbContext?.Dispose();
59          this.dbContext = null;
60      }
61
62      private void
    оформитьЗаказToolStripMenuItem_Click(object sender,
    EventArgs e)
63      {
64          OrderNew newChild = new OrderNew();

```

```

65             newChild.MdiParent = this;
66             newChild.Show();
67         }
68
69         private void
просмотретьменюToolStripMenuItem_Click(object
sender, EventArgs e)
70     {
71         Menu newChild = new Menu();
72         newChild.MdiParent = this;
73         newChild.Show();
74     }
75
76         private void
заказыToolStripMenuItem_Click(object sender,
EventArgs e)
77     {
78         ModerOrders newChild = new();
79         newChild.MdiParent = this;
80         newChild.Show();
81     }
82     }
83 }

```

Листинг 2 — Menu.cs

```

1 using Microsoft.Data.Sqlite;
2 using Microsoft.EntityFrameworkCore;

```

```

3  using Restauran.cs;
4  using Restauran.Methods;
5  using System;
6  using System.Collections;
7  using System.Collections.Generic;
8  using System.ComponentModel;
9  using System.Data;
10 using System.Diagnostics;
11 using System.Drawing;
12 using System.Linq;
13 using System.Text;
14 using System.Threading.Tasks;
15 using System.Windows.Forms;
16 using static
    Microsoft.EntityFrameworkCore.DbLoggerCategory;
17 using static Restauran.cs.Methods;
18
19 namespace restaurant
20 {
21     public partial class Menu : Form
22     {
23         Stopwatch stopwatch = new Stopwatch();
24         FormWork FoW = new FormWork();
25         DBWork DBW = new DBWork();
26         FileWork FiW = new FileWork();
27         public Menu()

```

```

28         {
29             //stopwatch.Restart();
30             InitializeComponent();
31             //stopwatch.Stop();
32             //MessageBox.Show($"Time
    {stopwatch.ElapsedMilliseconds}ms");
33         }
34
35     private async void Menu_Load(object sender,
    EventArgs e)
36     {
37         //stopwatch.Restart();
38         Type = await Task.Run(() =>
    FoW.ListAdd(this.Type));
39         //stopwatch.Stop();
40         //MessageBox.Show($"Time load
    {stopwatch.ElapsedMilliseconds}ms");
41     }
42
43     private DataContext? dbContext = new
    DataContext();
44     protected override async void
    OnLoad(EventArgs e)
45     {
46         stopwatch.Restart();
47         base.OnLoad(e);

```

```

48         await DBW.DBLoad(dbContext);
49         this.foodBindingSource.DataSource =
            dbContext.Foods.Local.ToBindingList();
50         stopwatch.Stop();
51         MessageBox.Show($"Time db load
            {stopwatch.ElapsedMilliseconds}ms");
52     }
53
54     protected override void
        OnClosing(CancelEventArgs e)
55     {
56         base.OnClosing(e);
57         this.dbContext?.Dispose();
58         this.dbContext = null;
59     }
60
61     private async void Type_MouseDown(object
        sender, MouseEventArgs e)
62     {
63         //stopwatch.Restart();
64         int index = Type.IndexFromPoint(e.X,
            e.Y);
65         DBW.DBFoodSort(index, this.Type,
            dataGridView1);
66         //stopwatch.Stop();

```

```

67             //MessageBox.Show($"Time type select
           {stopwatch.ElapsedMilliseconds}ms");
68         }
69
70     private void button1_Click_1(object sender,
           EventArgs e)
71     {
72         //stopwatch.Restart();
73         if (Type.SelectedIndex != -1)
74         {
75             DBW.DBReturnAllFoods(this.Type,
           this.dataGridView1);
76         }
77         //stopwatch.Stop();
78         //MessageBox.Show($"Time btn click
           {stopwatch.ElapsedMilliseconds}ms");
79     }
80 }
81 }

```

Листинг 3 — ModerMenu.cs

```

1  using Microsoft.EntityFrameworkCore;
2  using Restauran.cs;
3  using Restauran.Methods;
4  using System;
5  using System.Collections.Generic;

```

```

6 using System.ComponentModel;
7 using System.Data;
8 using System.Drawing;
9 using System.Linq;
10 using System.Text;
11 using System.Threading.Tasks;
12 using System.Windows.Forms;
13
14 namespace Restauran
15 {
16     public partial class ModerMenu : Form
17     {
18         public ModerMenu()
19         {
20             InitializeComponent();
21         }
22         private DataContext? dbContext = new
DataContext();
23         FormWork FoW = new FormWork();
24         DBWork DBW = new DBWork();
25         FileWork FiW = new FileWork();
26         protected async override void
OnLoad(EventArgs e)
27     {
28         base.OnLoad(e);
29         await DBW.DBLoad(dbContext!);

```



```

30         this.foodBindingSource.DataSource =
           dbContext.Foods.Local.ToBindingList();
31     }
32
33     protected override void
       OnClosing(CancelEventArgs e)
34     {
35         base.OnClosing(e);
36         this.dbContext?.Dispose();
37         this.dbContext = null;
38     }
39
40     private void button1_Click(object sender,
       EventArgs e)
41     {
42         DBW.DBSave(this.dataGridView2,
           dbContext!);
43     }
44
45
46     private void dataGridView2_CellClick(object
       sender, DataGridViewCellEventArgs e)
47     {
48         dataGridView2.Rows[e.RowIndex].Selected
           = true;
49     }

```

```

50
51
52     private void button3_Click(object sender,
        EventArgs e)
53     {
54         MessageBoxButtons buttons =
        MessageBoxButtons.YesNo;
55
56         if (MessageBox.Show( "Осторожнее в
        удалении", "Уверены?", buttons) ==
        System.Windows.Forms.DialogResult.Yes)
57         {
58
        dataGridView2.Rows.RemoveAt(dataGridView2.CurrentRow
        .Index);
59
        MessageBox.Show("Удалён продукт");
60     }
61 }
62 }
63 }

```

Листинг 4 — ModerOrders.cs

```

1  1
2  2using Microsoft.EntityFrameworkCore;
3  3 using Restauran.cs;
4  4 using System;
5  5 using System.Collections.Generic;

```

```

6 using System.ComponentModel;
7 using System.Data;
8 using System.Drawing;
9 using System.Linq;
10 using System.Text;
11 using System.Threading.Tasks;
12 using System.Windows.Forms;
13 using System.IO;
14 using System.Diagnostics;
15 using Restauran.Methods;
16
17 namespace Restauran
18 {
19     public partial class ModerOrders : Form
20     {
21         string directoryPath = @"..\Чеки";
22         string filePath = "чек";
23         FormWork FoW = new FormWork();
24         DBWork DBW = new DBWork();
25         FileWork FiW = new FileWork();
26
27         public ModerOrders()
28         {
29             InitializeComponent();
30         }

```

```

31         private DataContext? dbContext = new
           DataContext();
32         protected async override void
           OnLoad(EventArgs e)
33         {
34             base.OnLoad(e);
35             await DBW.DBLoad(dbContext!);
36             this.orderBindingSource.DataSource =
           dbContext.Orders.Local.ToBindingList();
37
           this.orderedFoodsBindingSource.DataSource =
           dbContext.OrderedFoods.Local.ToBindingList();
38         }
39
40         protected override void
           OnClosing(CancelEventArgs e)
41         {
42             base.OnClosing(e);
43             this.dbContext?.Dispose();
44             this.dbContext = null;
45         }
46
47         private void button3_Click(object sender,
           EventArgs e)
48         {

```

```

49             uint id =
                (uint)dataGridView1.SelectedRows[0].Cells[0].Value;
50             FiW.FileRead(directoryPath, filePath,
                id, this.listBox1);
51         }
52
53     private void dataGridView1_CellClick(object
        sender, DataGridViewCellEventArgs e)
54     {
55         dataGridView1.Rows[e.RowIndex].Selected
            = true;
56     }
57
58     private async void button4_Click(object
        sender, EventArgs e)
59     {
60         uint id =
            (uint)dataGridView1.SelectedRows[0].Cells[0].Value;
61         FiW.FileWrite(directoryPath, filePath,
            id);
62     }
63
64     private void button5_Click(object sender,
        EventArgs e)
65     {
66         try

```

```

67         {
68             Process.Start("explorer.exe",
        directoryPath);
69         }
70         catch (Exception)
        { MessageBox.Show("Возникли проблемы"); }
71     }
72
73     private void button2_Click(object sender,
        EventArgs e)
74     {
75         MessageBoxButtons buttons =
        MessageBoxButtons.YesNo;
76
77         if (MessageBox.Show("Осторожнее в
        удалении заказа", "Уверены?", buttons) ==
        System.Windows.Forms.DialogResult.Yes)
78         {
79
        dataGridView1.Rows.Remove(dataGridView1.SelectedRows
        [0]);
80
        MessageBox.Show("Удалён заказ и
        соответствующие ему продукты");
81     }
82 }
83

```

```

84         private void button6_Click(object sender,
           EventArgs e)
85         {
86             MessageBoxButtons buttons =
           MessageBoxButtons.YesNo;
87
88             if (MessageBox.Show("Осторожнее в
           удалении", "Уверены?", buttons) ==
           System.Windows.Forms.DialogResult.Yes)
89             {
90
           dataGridViewView2.Rows.RemoveAt(dataGridView2.CurrentRow
           .Index);
91             MessageBox.Show("Удалён заказанный
           продукт");
92         }
93     }
94
95     private void button1_Click(object sender,
           EventArgs e)
96     {
97         DBW.DBSave(this.dataGridView1,
           this.dataGridView1, dbContext!);
98     }
99 }
100 }

```

Листинг 5 — OrderNew.cs

```
1 using Microsoft.EntityFrameworkCore;
2 using Restauran.cs;
3 using System;
4 using System.Collections.Generic;
5 using System.ComponentModel;
6 using System.Data;
7 using System.Diagnostics;
8 using System.Drawing;
9 using System.Linq;
10 using System.Text;
11 using System.Threading.Tasks;
12 using System.Windows.Forms;
13 using Restauran.Methods;
14
15 namespace Restauran
16 {
17     public partial class OrderNew : Form
18     {
19         //public List<string> OrderedFoodName =
20         new();
21         //public List<uint> OrderedFoodCount =
22         new();
23         //public List<uint> OrderedFoodCost = new();
```



```

22         Dictionary<string, OrderedFood> OrderedFoods
    = new();
23         uint TotalCost = 0;
24         uint idWorker = 0;
25         string directoryPath = @"..\Чеки";
26         string filePath = "чек";
27         FormWork FoW = new FormWork();
28         DBWork DBW = new DBWork();
29         FileWork FiW = new FileWork();
30
31         public OrderNew()
32         {
33             InitializeComponent();
34         }
35         private DataContext? dbContext = new
    DataContext();
36         protected async override void
    OnLoad(EventArgs e)
37         {
38             base.OnLoad(e);
39             dbContext = new DataContext();
40             await DBW.DBLoad(dbContext);
41             this.foodBindingSource.DataSource =
    dbContext.Foods.Local.ToBindingList();
42         }
43

```

```

44         protected override void
        OnClosing(CancelEventArgs e)
45     {
46         base.OnClosing(e);
47         this.dbContext?.Dispose();
48         this.dbContext = null;
49     }
50     private void button1_Click(object sender,
        EventArgs e)
51     {
52         short count = 0;
53         try { count =
        short.Parse(textBox1.Text); }
54         catch (Exception)
        { MessageBox.Show("Заполните корректными данными");
        return; }
55
56         string selected = "";
57         try { selected =
        dataGridView2.SelectedRows[0].Cells[1].Value.ToString(); }
58         catch (Exception)
        { MessageBox.Show("Выберите что то из меню");
        return; }
59

```

```

60             uint cost = (uint)
                (dataGridView2.SelectedRows[0].Cells[2].Value);
61             if (OrderedFoods.ContainsKey(selected))
62             {
63                 OrderedFoods[selected].Count +=
count;
64                 OrderedFoods[selected].Cost = (uint)
                (cost * OrderedFoods[selected].Count);
65                 listBox1.Items.Clear();
66                 TotalCost = 0;
67                 foreach (var el in OrderedFoods)
68                 {
69                     listBox1.Items.Add($"{el.Key}
{el.Value.Count}    {el.Value.Cost:c}");
70                     TotalCost +=
OrderedFoods[selected].Cost;
71                 }
72             }
73             else
74             {
75                 OrderedFoods.Add(selected, new
OrderedFood((count), (uint)(count * cost)));
76
                listBox1.Items.Add($"{OrderedFoods.Last().Key}
{OrderedFoods.Last().Value.Count}
{OrderedFoods.Last().Value.Cost:c}");

```

```

77             TotalCost +=
OrderedFoods[selected].Cost;
78         }
79         dataGridView2.ClearSelection();
80         label3.Text = $"Итого: {TotalCost}";
81     }
82
83     private void dataGridView2_CellClick(object
sender, DataGridViewCellEventArgs e)
84     {
85         dataGridView2.Rows[e.RowIndex].Selected
= true;
86     }
87
88     private void button2_Click(object sender,
EventArgs e)
89     {
90         string key = "";
91         try { key =
listBox1.SelectedItem!.ToString()!.Split(" ")
[0]; }
92         catch (Exception) { MessageBox.Show("Вы
не выбрали что удалить"); return; }
93
94         listBox1.Items.Remove(listBox1.SelectedItem);

```

```

95             OrderedFoods.Remove(key);
96         }
97
98         private async void OrderNew_Load(object
sender, EventArgs e)
99     {
100         comboBox1 = await Task.Run(() =>
FoW.BoxAdd(this.comboBox1));
101         Type = await Task.Run(() =>
FoW.ListAdd(this.Type));
102     }
103
104     private void
comboBox1_SelectedValueChanged(object sender,
EventArgs e)
105     {
106         idWorker =
(uint)comboBox1.SelectedIndex + 1;
107         using (DataContext db = new
DataContext())
108         {
109             var wor = Enums.Workers.Where(e
=> e.Key == idWorker).Select(e => new
{ e.Value.LastName,
e.Value.Name }).FirstOrDefault();

```

```

110                label4.Text = $"{wor!.LastName}
                {wor.Name}";
111            }
112        }
113
114        private void button3_Click(object
            sender, EventArgs e)
115        {
116            FiW.FileWriteOrder(directoryPath,
            filePath, this.listBox1, this.label4, TotalCost,
            idWorker, OrderedFoods);
117            //MessageBox.Show("Заказ оформлен");
118
119        }
120
121        private void Type_MouseDown(object
            sender, MouseEventArgs e)
122        {
123            int index = Type.IndexFromPoint(e.X,
            e.Y);
124            DBW.DBFoodSort(index, this.Type,
            this.dataGridView2);
125        }
126    }
127 }

```

Листинг 6 — DBWork.cs

```
1  using Microsoft.EntityFrameworkCore;
2  using Restauran.cs;
3  using System;
4  using System.Collections.Generic;
5  using System.Linq;
6  using System.Text;
7  using System.Threading.Tasks;
8
9  namespace Restauran.Methods
10 {
11     internal class DBWork : Interfaces.IDataBase
12     {
13         public async Task DBLoad(DataContext
14             dbContext)
15         {
16             await
17                 dbContext.Database.EnsureCreatedAsync();
18             await dbContext.Foods.LoadAsync();
19             await
20                 dbContext.OrderedFoods.LoadAsync();
21             await dbContext.Orders.LoadAsync();
22         }
23         ///////////////
24         public async void DBFoodSort(int index,
25             CheckedListBox Type, DataGridView dgv)
```

```

22      {
23          if (index != ListBox.NoMatches)
24          {
25              Type.SetItemChecked(index, !
Type.GetItemChecked(index));
26
27              var selectedItems =
Type.CheckedItems.Cast<string>().ToList();
28              if (selectedItems.Any())
29              {
30                  //MessageBox.Show("hi");
31                  using (DataContext db = new
DataContext())
32                  {
33                      List<Food> data = new
List<Food>();
34                      foreach (string el in
Type.CheckedItems)
35                      {
36                          data.AddRange(await
db.Foods.Where(food => food.NameType ==
el).ToListAsync());
37                      }
38                      dgv.DataSource = data;
39                  }
40              }

```



```

41         }
42     }
43     ////////////
44     public async void
        DBReturnAllFoods(CheckedListBox list, DataGridView
        dgv)
45     {
46         for (int i = 0; i < list.Items.Count; i+
            +)
47         {
48             list.SetItemChecked(i, false);
49         }
50         using (DataContext db = new
            DataContext())
51         {
52             var query = db.Foods.AsQueryable();
53             var data = await
                query.ToListAsync();
54             dgv.DataSource = data;
55         }
56         list.SetSelected(list.SelectedIndex,
            false);
57         dgv.Refresh();
58     }
59     ////////////

```

```

60         public void DBSave(DataGridView dgv,
           DataContext dbContext)
61     {
62         dbContext.SaveChanges();
63         dgv.Refresh();
64         MessageBox.Show("Сохранено");
65     }
66     public void DBSave(DataGridView dgv,
           DataGridView dgv2, DataContext dbContext)
67     {
68         dbContext.SaveChanges();
69         dgv.Refresh();
70         dgv2.Refresh();
71         MessageBox.Show("Сохранено");
72     }
73     ////////////
74
75     }
76 }

```

Листинг 7 — FileWork.cs

```

1  using Microsoft.EntityFrameworkCore;
2  using Restauran.cs;
3  using System;
4  using System.Collections.Generic;
5  using System.Diagnostics;

```

```

6 using System.Linq;
7 using System.Text;
8 using System.Threading.Tasks;
9
10 namespace Restaurant.Methods
11 {
12     internal class FileWork : Interfaces.IFile
13     {
14         public void FileRead(string directoryPath,
15             string filePath, uint id, ListBox listBox1)
16         {
17             string totalPath = $"{directoryPath}\
18             {filePath}{id}.txt";
19             if (!Directory.Exists(directoryPath))
20             {
21                 Directory.CreateDirectory(directoryPath);
22             }
23             try
24             {
25                 using (StreamReader str =
26                     File.OpenText(totalPath))
27                 {
28                     listBox1.Items.Clear();
29                 }
30             }
31             catch { }
32         }
33     }
34 }

```

```

26             if (str.ReadLine() == null ||
                str.ReadLine() == "") { MessageBox.Show("Файл пустой
                ._."); return; }
27             string[] strings =
                str.ReadToEnd().Split('\n');
28             foreach (var el in strings)
29             {
30                 listBox1.Items.Add(el);
31             }
32         }
33     }
34     catch (Exception)
35     {
36         MessageBox.Show("Проблема с
                файлом");
37         return;
38     }
39 }
40 ///////////////
41 public async void FileWrite(string
    directoryPath, string filePath, uint id)
42 {
43     string totalPath = $"{directoryPath}\
        {filePath}{id}.txt";
44     if (!Directory.Exists(directoryPath))
45     {

```

```

46
    Directory.CreateDirectory(directoryPath);
47        }
48        using (StreamWriter str =
    File.CreateText(totalPath))
49        {
50            using (DataContext db = new
    DataContext())
51            {
52                //var
    Process.Start("explorer.exe", folderPath);
53                var el = await db.Orders.Where(e
    => e.IDOrder == id).FirstOrDefaultAsync();
54                var el2 = await
    db.OrderedFoods.Where(e => e.IDOrder ==
    el!.IDOrder).ToListAsync();
55                var el4 = Enums.Workers.Where(e
    => e.Key == el!.IDWorker).FirstOrDefault();
56
    str.WriteLine($"Data\t\t\t\t{el!.Date}");
57                str.WriteLine($"Gusto Testo");
58                str.WriteLine($"№{el.IDOrder}
    \t\t\t\tПариж Ленина 19");
59
    str.WriteLine($"-----
    -----");

```

```

60                foreach (var l in el2)
61                {
62                    var el3 = await
db.Foods.Where(e => e.IDFood ==
l.IDFood).FirstOrDefaultAsync();
63                    short length =
(short)el3.NameF!.Split(' ').Length;
64                    short l0 =
(short)el3.NameF!.Split(' ')[0].Length;
65                    short l01 = 0;
66                    if (length > 1)
67                        l01 = Convert.ToInt16(l0
+ el3.NameF!.Split(' ')[1].Length);
68
69                        //if (length > 2)
70
//str.WriteLine($"{el3!.NameF,-15}{l!.Count}шт
*{el3.Cost}={l.Cost:c}");
71                    //else
72                    await
str.WriteLineAsync($"{el3!.NameF,-30}{l!.Count}шт
*{el3.Cost}={l.Cost:c}");
73                }
74
str.WriteLine($"-----
-----");

```

```

75                str.WriteLine($"Итого =\t\t\t\t\t
{el.Cost:c}");
76                str.WriteLine($"Обслужил
{el4!.Value.LastName} {el4!.Value.Name}
{el4!.Value.FatherName}");
77            }
78        }
79        MessageBox.Show("Чек сохранён");
80    }
81    ////////////
82    public async void FileWriteOrder(string
directoryPath, string filePath, ListBox listBox1,
System.Windows.Forms.Label label4, uint TotalCost,
uint idWorker, Dictionary<string, OrderedFood>
OrderedFoods)
83    {
84        string totalPath = "";
85        if (!Directory.Exists(directoryPath))
86        {
87            Directory.CreateDirectory(directoryPath);
88        }
89        if (listBox1.Items.Count == 0)
90        {
            MessageBox.Show("А у вас заказ пустой"); return; }
            using (DataContext db = new
DataContext())

```

```

91          {
92              //uint id0 =
          (uint)dataGridView1.SelectedRows[0].Cells[0].Value;
          // исправить ( исправил >=) )
93              uint id0 = db.Orders.Max(e =>
          e.IDOrder) + 1;
94              totalPath = $"{directoryPath}\
          {filePath}{id0}.txt";
95              string WName = label4.Text;
96              if (WName == "не выбрано")
          { MessageBox.Show("Кто вы?"); return; }
97              DateTime dateTime = DateTime.Now;
98
99              db.Orders.Add(new Order { Date =
          dateTime, Cost = TotalCost, IDWorker = idWorker });
100              await db.SaveChangesAsync();
101              foreach (var el in OrderedFoods)
102              {
103                  uint idF = await
          db.Foods.Where(e => e.NameF == el.Key).Select(e =>
          e.IDFood).FirstOrDefaultAsync();
104                  db.OrderedFoods.Add(new
          OrderedFood { Count = el.Value.Count, Cost =
          el.Value.Cost, IDFood = idF, IDOrder = id0 });

```



```

105                //MessageBox.Show($"Count =
                {el.Value.Count}, Cost = {el.Value.Cost}, IDFood =
                {idF}, IDOrder = {idO}");
106                await db.SaveChangesAsync();
107            }
108            ///////////
109            using (StreamWriter str =
                File.CreateText(totalPath))
110            {
111                var el = await
                db.Orders.Where(e => e.IDOrder ==
                idO).FirstOrDefaultAsync();
112                var el2 = await
                db.OrderedFoods.Where(e => e.IDOrder ==
                el!.IDOrder).ToListAsync();
113                var el4 =
                Enums.Workers.Where(e => e.Key ==
                el!.IDWorker).FirstOrDefault();
114
                str.WriteLine($"Дата\t\t\t{el!.Date}");
115                str.WriteLine($"Gusto
                Testo");
116                str.WriteLine($"№
                {el.IDOrder}\t\t\tПариж Ленина 19");
117
                str.WriteLine($"-----");

```

```

118                foreach (var l in el2)
119                {
120                    var el3 =
                        db.Foods.Where(e => e.IDFood ==
                        l.IDFood).FirstOrDefault();
121                        //if
                        (el3!.NameF!.Split(' ').Length > 1 ||
                        el3!.NameF!.Split(' ')[0].Length > 7)
122                        //    await
                        str.WriteLineAsync($"{el3!.NameF}\t\t{l!.Count}шт
                        *{el3.Cost}={l.Cost:c}");
123                        //else
124                        //    await
                        str.WriteLineAsync($"{el3!.NameF}\t\t\t{l!.Count}шт
                        *{el3.Cost}={l.Cost:c}");
125                        await
                        str.WriteLineAsync($"{el3!.NameF,-30}{l!.Count}шт
                        *{el3.Cost}={l.Cost:c}");
126                }
127
                        str.WriteLine($"-----");
128                        str.WriteLine($"Итого =\t\t\t
                        {el.Cost:c}");
129                        str.WriteLine($"Обслужил
                        {el4!.Value.LastName} {el4!.Value.Name}
                        {el4!.Value.FatherName}");

```

```

130                }
131            }
132            MessageBoxButtons buttons =
                MessageBoxButtons.YesNo;
133
134            if (MessageBox.Show("Открыть файл
                чека?", "Заказ оформлен", buttons) ==
                System.Windows.Forms.DialogResult.Yes)
135            {
136                Process.Start("notepad.exe",
                totalPath);
137            }
138        }
139    }
140 }

```

Листинг 8 — FormWork.cs

```

1  using Restauran.cs;
2  using System;
3  using System.Collections.Generic;
4  using System.Linq;
5  using System.Text;
6  using System.Threading.Tasks;
7
8  namespace Restauran.Methods
9  {
10     internal class FormWork

```

```

11      {
12          public CheckedListBox ListAdd(CheckedListBox
list)
13      {
14          foreach (var el in Enums.Types)
15          {
16              list.Items.Add(el.Value);
17          }
18          return list;
19      }
20          ////////////
21      public System.Windows.Forms.ComboBox
BoxAdd(System.Windows.Forms.ComboBox cb1)
22      {
23          foreach (var el in Enums.Workers.Values)
24          {
25              cb1.Items.Add(el.LastName!);
26          }
27          return cb1;
28      }
29          ////////////
30
31      }
32 }

```

Листинг 9 — DataContext.cs

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6 using Microsoft.EntityFrameworkCore;
7
8 namespace Restaurant.cs
9 {
10     internal class DataContext : DbContext
11     {
12         public DbSet<Food> Foods { get; set; }
13         public DbSet<Order> Orders { get; set; }
14         public DbSet<OrderedFood> OrderedFoods
15         { get; set; }
16         public DataContext() =>
17             Database.EnsureCreated();
18         protected override void
19             OnConfiguring(DbContextOptionsBuilder
20             optionsBuilder) => optionsBuilder.UseSqlite(@"Data
21             Source=..\DataBase.db");
22
23         protected override void
24             OnModelCreating(ModelBuilder modelBuilder)
25         {
26         }
```

```

20         modelBuilder.Entity<Food>().HasData(
21             // Десерты
22             new Food("Тирамису", "Сочетает
кремовую маскарпоне с согревающим ароматом кофе и
шоколада", 200) { NameType = "Десерт", IDFood = 1 },
23             new Food("Желе", "Нежное желе из
фруктового и ягодного сока", 100) { NameType =
"Десерт", IDFood = 2 },
24             new Food("Шоколадный торт", "Влажный
пирог с насыщенным шоколадным вкусом", 300)
{ NameType = "Десерт", IDFood = 3 },
25             new Food("Павлова", "Легкий безе с
кремом и свежими фруктами", 250) { NameType =
"Десерт", IDFood = 4 },
26             new Food("Фруктовый салат", "Ассорти
из свежих сезонных фруктов", 150) { NameType =
"Десерт", IDFood = 5 },
27
28             // Напитки
29             new Food("Чай", "Чёрный чай из
лепестков дуба", 50) { NameType = "Напиток", IDFood
= 6 },
30             new Food("Красное вино",
"Выдержанное красное вино", 1000) { NameType =
"Напиток", IDFood = 7 },

```

```

31         new Food("Белое вино", "Легкое и
освежающее белое вино", 950) { NameType = "Напиток",
IDFood = 8 },
32         new Food("Коктейль Мохито",
"Освежающий коктейль с мятой и лаймом", 400)
{ NameType = "Напиток", IDFood = 9 },
33         new Food("Кофе", "Ароматный кофе с
нотками шоколада", 80) { NameType = "Напиток",
IDFood = 10 },
34
35         // Закуски высокой кухни
36         new Food("Канapé с икрой", "Канapé с
черной солью и нежной красной икрой", 250)
{ NameType = "Закуска высокой кухни", IDFood = 11 },
37         new Food("Тартар из лосося", "Свежий
лосось, мелко нарезанный с пряностями", 400)
{ NameType = "Закуска высокой кухни", IDFood = 12 },
38         new Food("Фуа-гра", "Изысканная
печень утки, подается с тостами", 1200) { NameType =
"Закуска высокой кухни", IDFood = 13 },
39         new Food("Устрицы", "Свежие устрицы
с лимонным соком", 800) { NameType = "Закуска
высокой кухни", IDFood = 14 },
40         new Food("Креветки в чесночном
соусе", "Сочные креветки в ароматном чесночном

```

```

    course", 650) { NameType = "Закуска высокой кухни",
    IDFood = 15 },
41
42         // Основные блюда
43         new Food("Стейк", "Сочный говяжий
    стейк, приготовленный до средней степени прожарки",
    1500) { NameType = "Основное блюдо", IDFood = 16 },
44         new Food("Лосось на гриле",
    "Запеченный лосось с лимоном и пряностями", 1200)
    { NameType = "Основное блюдо", IDFood = 17 },
45         new Food("Паста с креветками",
    "Паста с нежными креветками и томатным соусом", 800)
    { NameType = "Основное блюдо", IDFood = 18 },
46         new Food("Филе цыпленка", "Филе
    курицы с овощами на гриле", 700) { NameType =
    "Основное блюдо", IDFood = 19 },
47         new Food("Вегетарианская пицца",
    "Пицца с свежими овощами и сырами", 600) { NameType
    = "Основное блюдо", IDFood = 20 },
48         new Food("Говяжий бургер", "Сочный
    бургер с говядиной, свежими овощами и соусом", 900)
    { NameType = "Основное блюдо", IDFood = 21 },
49         new Food("Цезарь с курицей", "Салат
    Цезарь с гриль-курицей и пармезаном", 500)
    { NameType = "Основное блюдо", IDFood = 22 },

```



```
50         new Food("Ризотто с грибами",  
    "Кремовое ризотто с лесными грибами и трюфельным  
    маслом", 750) { NameType = "Основное блюдо", IDFood  
    = 23 },  
51         new Food("Куриные крылышки",  
    "Крылышки, запеченные в остром соусе", 450)  
    { NameType = "Основное блюдо", IDFood = 24 },  
52         new Food("Паста альфредо", "Паста с  
    кремовым соусом альфредо и пармезаном", 850)  
    { NameType = "Основное блюдо", IDFood = 25 },  
53         new Food("Кебаб", "Нежные кусочки  
    мяса, запеченные на шампуре", 1000) { NameType =  
    "Основное блюдо", IDFood = 26 },  
54         new Food("Рыба на пару", "Нежная  
    рыба, приготовленная на пару с лимоном и зеленью",  
    650) { NameType = "Основное блюдо", IDFood = 27 },  
55         new Food("Чили кон карне", "Острая  
    мексиканская фасоль с мясом и специями", 800)  
    { NameType = "Основное блюдо", IDFood = 28 },  
56         new Food("Лазанья", "Лазанья с  
    мясом, сыром и томатным соусом", 950) { NameType =  
    "Основное блюдо", IDFood = 29 },  
57         new Food("Запеченные овощи",  
    "Ассорти запеченных сезонных овощей с травами", 500)  
    { NameType = "Основное блюдо", IDFood = 30 },  
58
```

```

59          // Закуски
60          new Food("Брускетта", "Гренка с
            помидорами и базиликом", 100) { NameType =
            "Закуска", IDFood = 31 },
61          new Food("Сырная тарелка", "Ассорти
            из различных сыров с медом и орехами", 500)
            { NameType = "Закуска", IDFood = 32 },
62          new Food("Овощи гриль", "Ассорти
            овощей, приготовленных на гриле", 300) { NameType =
            "Закуска", IDFood = 33 },
63          new Food("Куриные крылышки", "Со
            spicy соусом", 450) { NameType = "Закуска", IDFood =
            34 },
64          new Food("Чипсы с соусом гуакамоле",
            "Кукурузные чипсы с нежным соусом", 200) { NameType
            = "Закуска", IDFood = 35 }
65      );
66
67      modelBuilder.Entity<Order>().HasData(
68          new Order(300, DateTime.Now)
            { IDOrder = 1, IDWorker = 1 },
69          new Order(450,
            DateTime.Now.AddMinutes(15)) { IDOrder = 2, IDWorker
            = 2 },

```

```

70             new Order(200,
DateTime.Now.AddMinutes(30)) { IDOrder = 3, IDWorker
= 3 },
71             new Order(500,
DateTime.Now.AddMinutes(45)) { IDOrder = 4, IDWorker
= 1 },
72             new Order(350,
DateTime.Now.AddMinutes(60)) { IDOrder = 5, IDWorker
= 2 },
73             new Order(400,
DateTime.Now.AddMinutes(75)) { IDOrder = 6, IDWorker
= 3 }, // Новый заказ
74             new Order(600,
DateTime.Now.AddMinutes(90)) { IDOrder = 7, IDWorker
= 1 } // Новый заказ
75         );
76
77
modelBuilder.Entity<OrderedFood>().HasData(
78             new OrderedFood(1, 100)
{ IDOrderedFood = 1, IDFood = 3, IDOrder = 1 }, //
Шоколадный торт
79             new OrderedFood(2, 200)
{ IDOrderedFood = 2, IDFood = 5, IDOrder = 1 }, //
Фруктовый салат

```

```
80          new OrderedFood(3, 300)
    { IDOrderedFood = 3, IDFood = 11, IDOrder = 2 }, //
    Канапе с икрой
81          new OrderedFood(4, 150)
    { IDOrderedFood = 4, IDFood = 12, IDOrder = 2 }, //
    Тартар из лосося
82          new OrderedFood(5, 120)
    { IDOrderedFood = 5, IDFood = 7, IDOrder = 3 }, //
    Красное вино
83          new OrderedFood(6, 100)
    { IDOrderedFood = 6, IDFood = 16, IDOrder = 4 }, //
    Стейк
84          new OrderedFood(7, 80)
    { IDOrderedFood = 7, IDFood = 20, IDOrder = 5 }, //
    Вегетарианская пицца
85          new OrderedFood(8, 60)
    { IDOrderedFood = 8, IDFood = 22, IDOrder = 5 }, //
    Сырная тарелка
86          new OrderedFood(9, 250)
    { IDOrderedFood = 9, IDFood = 26, IDOrder = 6 }, //
    Кебаб
87          new OrderedFood(10, 300)
    { IDOrderedFood = 10, IDFood = 29, IDOrder = 6 }, //
    Лазанья
```

```

88             new OrderedFood(11, 200)
           { IDOrderedFood = 11, IDFood = 12, IDOrder = 7 }, //
           Тартар из лосося
89             new OrderedFood(12, 150)
           { IDOrderedFood = 12, IDFood = 23, IDOrder = 7 } //
           Ризотто с грибами
90         );
91
92
93     }
94 }
95 }

```

Листинг 10 — Interfaces.cs

```

1  1
2  2using System;
3  3using System.Collections.Generic;
4  4using System.Linq;
5  5using System.Text;
6  6using System.Threading.Tasks;
7  7
8  8namespace Restauran.cs
9  9{
10 10    internal class Interfaces
11 11    {
12 12        public interface IFile
13 13        {

```

```

14         public void FileRead(string
        directoryPath, string filePath, uint id, ListBox
        listBox1);
15         public void FileWrite(string
        directoryPath, string filePath, uint id);
16     }
17     public interface IDatabase
18     {
19         public Task DBLoad(DataContext
        dbContext);
20         public void DBSave(DataGridView dgv,
        DataContext dbContext);
21     }
22 }
23 }

```

Листинг 11 — Objects.cs

```

1 using System;
2 using System.Collections.Generic;
3 using System.ComponentModel.DataAnnotations;
4 using System.ComponentModel.DataAnnotations.Schema;
5 using System.Linq;
6 using System.Text;
7 using System.Threading.Tasks;
8 using Microsoft.EntityFrameworkCore;
9 using Microsoft.VisualBasic.ApplicationServices;

```

```

10 using static
    Microsoft.EntityFrameworkCore.DbLoggerCategory.Datab
    ase;
11
12
13 namespace Restauran.cs
14 {
15     public abstract class BaseConstruct
16     {
17         protected BaseConstruct() { }
18     }
19     public static class Enums
20     {
21         public static Dictionary<int, string> Types
            = new Dictionary<int, string> {
22             { 1, "Десерт" },
23             { 2, "Напиток" },
24             { 3, "Закуска высокой кухни" },
25             { 4, "Основное блюдо" },
26             { 5, "Закуска" }
27         };
28
29         public static Dictionary<int, Worker>
            Workers = new Dictionary<int, Worker> {
30             {1, new Worker{Name = "Анатолий",
                LastName = "Чернов", FatherName = "Степанович" } },

```

```

31          {2, new Worker{Name = "Мария", LastName
    = "Иванова", FatherName = "Петровна" } },
32          {3, new Worker{Name = "Дмитрий",
    LastName = "Сидоров", FatherName = "Александрович" }
    }
33      };
34  }
35  public class Food : BaseConstruct
36  {
37
38      [Key] public uint IDFood { get; set; }
39      public string? NameF { get; set; }
40      public string Description { get; set; }
41      public uint Cost { get; set; }
42      public string NameType { get; set; }
43      public Food(string? Name, string Des, uint
    Cost)
44      {
45          NameF = Name;
46          Description = Des;
47          this.Cost = Cost;
48      }
49      public Food():base() { }
50  }
51  ////////////
52  public class OrderedFood : BaseConstruct

```



```

53     {
54         [Key] public uint IDOrderedFood { get;
           set; }
55         public short Count { get; set; }
56         public uint Cost { get; set; }
57         public uint IDFood { get; set; }
58         [ForeignKey("IDFood")] public Food? Food
           { get; set; }
59         public uint IDOrder { get; set; }
60         [ForeignKey("IDOrder")] public Order? Order
           { get; set; }
61         public OrderedFood(short Cou, uint Cos)
62         {
63             Count = Cou;
64             Cost = Cos;
65         }
66         public OrderedFood():base() { }
67     }
68     //////////////////////////////////
69     public class Order : BaseConstruct
70     {
71         [Key] public uint IDOrder { get; set; }
72         public DateTime Date { get; set; }
73         public uint Cost { get; set; }
74         public List<OrderedFood> OrderedFoods { get;
           set; } = new();

```

```

75         public uint IDWorker { get; set; }
76         public Order(uint Cos, DateTime Date)
77         {
78             Cost = Cos;
79             this.Date = Date;
80         }
81         public Order() : base() { }
82     }
83     //////////////////////////////////
84     public class Worker
85     {
86         public String? Name { get; set; }
87         public String? LastName { get; set; }
88         public String? FatherName { get; set; }
89     }
90 }

```

ПРИЛОЖЕНИЕ Б

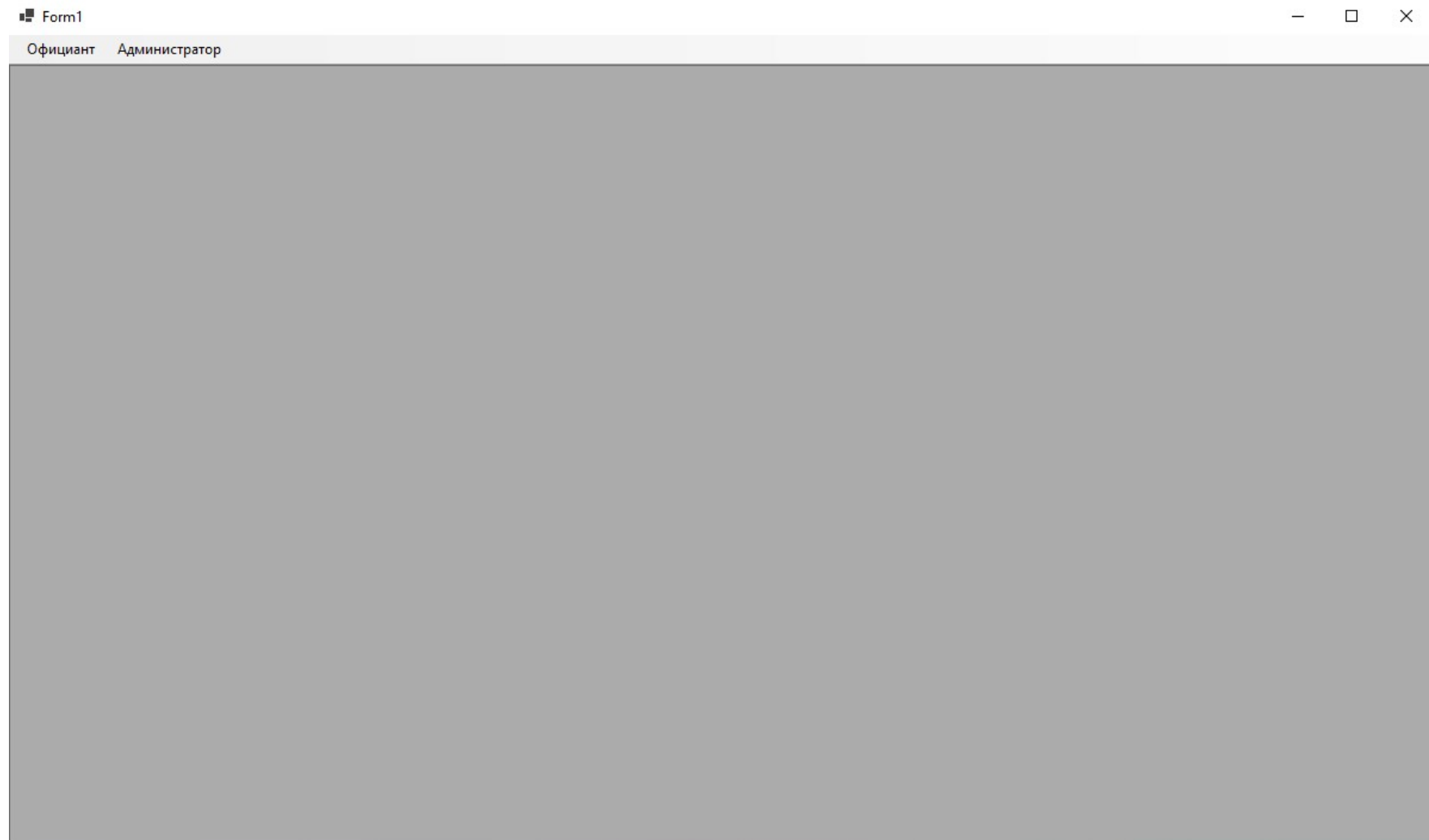


Рисунок 1 — Основная форма

Form1

Официант Администратор

Menu

☐ Десерт
☒ Напиток
☐ Закуска высокой к...
☐ Основное блюдо
☒ Закуска

Всё

Меню

	NameF	Cost	Description	NameType
▶	Чай	50	Чёрный чай из лепестков дуба	Напиток
	Красное ви...	1000	Выдержанное красное вино	Напиток
	Белое вино	950	Легкое и освежающее белое вино	Напиток
	Коктейль ...	400	Освежающий коктейль с мятой и лаймом	Напиток
	Кофе	80	Ароматный кофе с нотками шоколада	Напиток
	Брускетта	100	Гренка с помидорами и базиликом	Закуска
	Сырная та...	500	Ассорти из различных сыров с медом и орехами	Закуска
	Овощи гри...	300	Ассорти овощей, приготовленных на гриле	Закуска
	Куриные к...	450	Со spicy соусом	Закуска
	Чипсы с со...	200	Кукурузные чипсы с нежным соусом	Закуска

Рисунок 2 — Официант-просмотр

OrderNew

☐ Десерт

☐ Напиток

☒ **Закуска высокой кухни**

☐ Основное блюдо

☐ Закуска

Количество

Добавить к заказу

Удалить из заказа

Иванова

Иванова Мария

	NameF	Cost	Description
	Канapé с и...	250	Канapé с черной солью и нежной красной икрой
	Тартар из л...	400	Свежий лосось, мелко нарезанный с пряностями
►	Фуа-гра	1200	Изысканная печень утки, подается с тостами
	Устрицы	800	Свежие устрицы с лимонным соком
	Креветки в ...	650	Сочные креветки в ароматном чесночном соусе

Заказ

Итог: 1800

Шоколадный торт 2 600,00 Р

Фуа-гра 1 1 200,00 Р

Оформить

Рисунок 3 — Официант-оформить

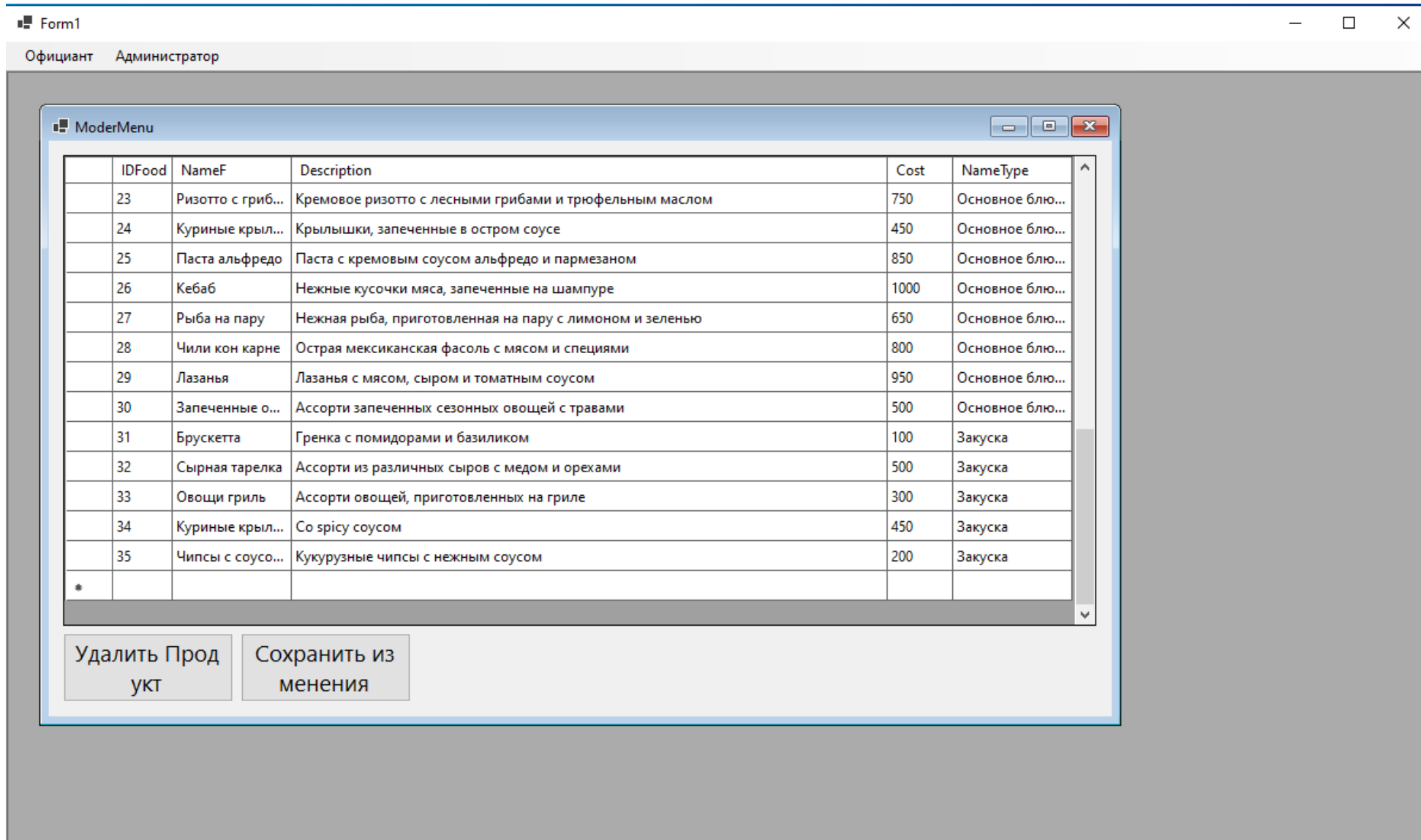


Рисунок 4 — Администратор-меню

Form1

Официант Администратор

МодерOrders

	IDOrder	Cost
▶	1	300
	2	450
	3	200
	4	500
	5	350
	6	400
	7	600
*		

Удалить Заказ Удалить Прод укт

Сохранить

Вывести текст Сохранить че к

Открыть папку с чеками

	IDOrderedFood	Count	Cost	IDFood	IDOrder
▶	1	1	100	3	1
	2	2	200	5	1
*					

Чек

Дата [дата]
[Название заведения]
[номер чека] адрес
еда1 1шт*200=200
еда2 1шт*200=200
ИТОГ =400
Обслужил [ФИО]

Рисунок 5 — Администратор-заказы

чек3.txt – Блокнот

Файл	Правка	Формат	Вид	Справка
Дата				05.11.2024 22:28:22
Gusto Testo				
№3				Париж Ленина 19

Оливки			1шт	*100=100,00 Р
Сыр			1шт	*50=50,00 Р
Тирамису			1шт	*200=200,00 Р
Желе			1шт	*100=100,00 Р
Чай			1шт	*50=50,00 Р
Красное вино			1шт	*1000=1 000,00 Р

Итого =				1 500,00 Р
Обслужил Чернов Анатолий Степанович				

Рисунок 6 — Файл