|  |  |  |
| --- | --- | --- |
| Министерство науки и высшего образования РФ  Федеральное государственное автономное образовательное учреждение высшего образования  «Пермский государственный национальный исследовательский университет» | | |
|  | Институт компьютерных наук и технологий | |
| **ОТЧЁТ**  по лабораторной работе №1  по дисциплине «Языки программирования»  Вариант 1 | | |
|  | | Работу выполнил  студент группы ПМИ-1,2-2023 2 курса  Туктамышев Р. Д.  «\_\_\_» \_\_\_\_\_\_\_\_2024 г. |
| Работу проверил  Ракина В.Д.  «\_\_\_» \_\_\_\_\_\_\_\_\_ 2024 г. |
| Пермь 2024 | | |

СОДЕРЖАНИЕ

[Структура проекта 3](#_Toc191521415)

[Классы-сущности: 3](#_Toc191521416)

[Класс для чтения данных: 3](#_Toc191521417)

[Класс для управления данными: 4](#_Toc191521418)

[Класс для записи данных: 5](#_Toc191521419)

[Класс для логирования: 5](#_Toc191521420)

# Код <https://github.com/Tuktamish/lab5>

# Структура проекта

Данный проект представляет собой систему управления базой данных для клиентов, заказов, блюд и составов заказов. Он реализован на языке C# и использует библиотеку NPOI для работы с Excel-файлами. Система позволяет выполнять операции CRUD (создание, чтение, обновление и удаление) для различных сущностей, а также выполнять запросы для получения специфической информации.

Проект состоит из нескольких классов, каждый из которых отвечает за определенную функциональность:

# Классы-сущности:

* **Client**: Класс, представляющий клиента с полями для идентификатора, фамилии, имени, отчества и адреса. Содержит конструктор для инициализации и метод ToString() для отображения информации о клиенте.
* **Dish**: Класс, представляющий блюдо с полями для идентификатора, названия и цены. Также имеет конструктор и метод ToString() для отображения информации о блюде.
* **Order**: Класс, представляющий заказ с полями для идентификатора, даты заказа, идентификатора клиента, цены доставки и статуса доставки. Включает конструктор и метод ToString().
* **OrderItem**: Класс, представляющий элемент заказа с полями для идентификатора элемента, идентификатора заказа, идентификатора блюда и количества. Содержит конструктор и метод ToString().

# Класс для чтения данных:

**DatabaseReader**: Класс, отвечающий за чтение данных из Excel-файлов. Содержит методы для чтения клиентов, заказов, составов заказов и блюд. Каждый метод открывает файл, считывает данные и возвращает соответствующий список объектов.

Он включает следующие методы:

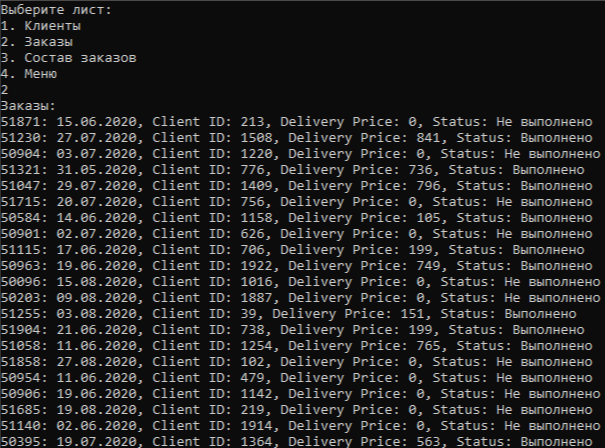
1. **ReadClients**: Читает данные клиентов из листа "Клиенты" и возвращает список объектов Client.
2. **ReadOrders**: Читает данные заказов из листа "Заказы" и возвращает список объектов Order, включая обработку дат и цен.
3. **ReadOrderItems**: Читает данные элементов заказов из листа "Состав заказов" и возвращает список объектов OrderItem.
4. **ReadDishes**: Читает данные блюд из листа "Меню" и возвращает список объектов Dish, также обрабатывая цены.

Класс использует библиотеку NPOI для работы с Excel и включает обработку ошибок для корректного чтения данных.

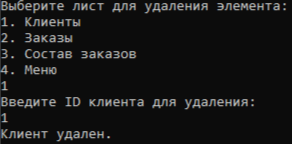
# Класс для управления данными:

**DatabaseService**: Класс, предоставляющий методы для управления данными. Он включает функции для просмотра, удаления, редактирования и добавления клиентов, заказов, составов заказов и блюд. Также реализует методы для выполнения специфических запросов

1. **Методы для просмотра данных:**
   * ViewClients(): Выводит список клиентов.
   * ViewOrders(): Выводит список заказов.
   * ViewOrderItems(): Выводит состав заказов.
   * ViewDishes(): Выводит меню.



1. **Методы для удаления данных:**
   * + DeleteClient(int clientId, string filePath): Удаляет клиента и связанные с ним заказы и элементы заказов, затем сохраняет изменения в Excel.
     + DeleteOrder(int orderId, string filePath): Удаляет заказ и связанные с ним элементы, затем сохраняет изменения.
     + DeleteOrderItem(int orderItemId, string filePath): Удаляет элемент заказа и сохраняет изменения.
     + DeleteDish(int dishId, string filePath): Удаляет блюдо и связанные с ним элементы заказа, затем сохраняет изменения.



1. **Методы для редактирования данных:**

* EditClient(int clientId, string filePath): Редактирует данные клиента и сохраняет изменения.
* EditOrder(int orderId, string filePath): Редактирует данные заказа и сохраняет изменения.
* EditOrderItem(int orderItemId, string filePath): Редактирует данные элемента заказа и сохраняет изменения.
* EditDish(int dishId, string filePath): Редактирует данные блюда и сохраняет изменения.



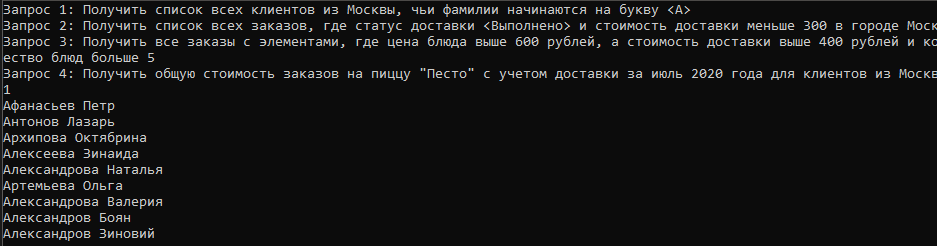
1. **Методы для добавления данных:**

* AddClientFromConsole(string filePath): Добавляет нового клиента, запрашивая данные у пользователя, и сохраняет изменения.
* AddOrderFromConsole(string filePath): Добавляет новый заказ, запрашивая данные у пользователя, и сохраняет изменения.
* AddOrderItemFromConsole(string filePath): Добавляет новый элемент заказа, запрашивая данные у пользователя, и сохраняет изменения.
* AddDishFromConsole(string filePath): Добавляет новое блюдо, запрашивая данные у пользователя, и сохраняет изменения.



1. **Запросы для выборки данных:**

* first\_1table(): Получает список клиентов из Москвы с фамилиями, начинающимися на "А".
* second\_2table(): Получает список заказов с выполненным статусом доставки и стоимостью доставки меньше 300 рублей для клиентов из Москвы.
* firth\_3table(): Получает все заказы с элементами, где цена блюда выше 600 рублей, стоимость доставки выше 400 рублей и количество блюд больше 5.
* fourth\_3table(): Получает общую стоимость заказов на пиццу "Пицца Песто" с учетом доставки за июнь 2020 года для клиентов из Москвы.



# Класс для записи данных:

**DatabaseWriter**: Класс, отвечающий за запись данных обратно в Excel-файлы. Содержит методы для сохранения клиентов, заказов, составов заказов и блюд. Каждый метод открывает файл, очищает существующие данные и записывает новые.

Каждый метод открывает файл Excel, обновляет соответствующий лист, а затем записывает изменения в файл, обеспечивая актуальность данных.

# Класс для логирования:

**Logger**: Класс, который записывает события в лог-файл с меткой времени. Содержит метод Log(), который принимает сообщение и добавляет его в файл. Также имеется вариант создать новый файл для записи или уже существующий.

