**Практическая работа 1. Создание приложения для производственной компании «Мастер пол».**

**Цель работы:** получить практические навыки создания информационной системы с применением импорта данных.

**Перечень оборудования и программного обеспечения:**

- Персональный компьютер;

- Microsoft Office (Word, Visio);

- Microsoft Visual Studio.

**Порядок выполнение работы:**

**База данных.**

Для начала разработки программы производится анализ предметной области, после чего создается ER - диаграмма базы данных, показанная на рисунке 1.

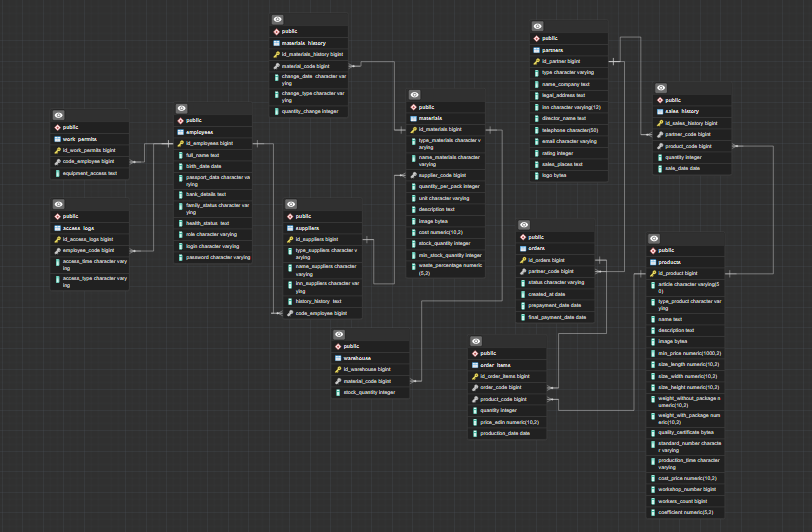


Рисунок 1 – ER - диаграмма базы данных

Были созданы такие таблицы базы данных как:

Таблица «Партнеры» хранит сведения о деловых партнерах компании. Ключевые атрибуты: уникальный идентификатор, тип партнера, наименование компании, юридический адрес, ИНН, ФИО директора, контактная информация (телефон, email), логотип, рейтинг партнера и информация о местах продаж.

Таблица «История реализации продукции» содержит записи о каждой продаже продукции партнерам. Отражает идентификатор партнера, идентификатор продукта, количество проданного товара и дату продажи, позволяя отслеживать объемы продаж и динамику по каждому партнеру и продукту.

Таблица «Продукция» содержит детальную информацию о производимой продукции. Включает: уникальный артикул, тип, наименование, описание, изображение, минимальную цену для партнера, габариты упаковки (длина, ширина, высота), вес (без и с упаковкой), сертификат качества, номер стандарта, время изготовления, себестоимость, номер цеха и количество сотрудников на производстве.

Таблица «Заявки» хранит информацию о заказах, поступающих от партнеров. Записывается идентификатор партнера, статус заявки (новая, ожидание оплаты, в производстве, готово, отменена), дата создания, дата предоплаты и дата окончательной оплаты.

Таблица «Позиции в заявке» детализирует состав каждой заявки, указывая, какие продукты и в каком количестве были заказаны. Включает идентификатор заявки, идентификатор продукта, количество продукции, цену за единицу и дату производства.

Таблица «Сотрудники» содержит персональные данные о сотрудниках компании. Фиксируются: ФИО, дата рождения, паспортные данные, банковские реквизиты, семейное положение, состояние здоровья, логин, пароль и роль сотрудника.

Таблица «Кадры» определяет права доступа сотрудников к различного рода оборудованию. Здесь хранится информация об идентификаторе сотрудника и списке оборудования, к которому у него есть доступ.

Таблица «Доступ» представляет собой журнал регистрации входов и выходов сотрудников. Записываются: идентификатор сотрудника, время доступа и тип доступа (Вход или Выход).

Таблица «Материалы» содержит сведения о сырье и материалах, используемых в производстве. Включает: тип, наименование, идентификатор поставщика, количество в упаковке, единицу измерения, описание, изображение, стоимость за единицу, количество на складе, минимальное количество на складе и процент брака.

Таблица «История материалов» отслеживает изменения количества материалов. Фиксируются: идентификатор материала, дата изменения, тип изменения (Приход или Расход) и величина изменения.

Таблица «Склад» содержит информацию о текущем количестве материалов, хранящихся на складе. Обязательно содержит идентификатор материала и его количество.

Таблица «Поставщики» предоставляет информацию о поставщиках материалов, включая: тип поставщика, наименование компании, ИНН и историю поставок.

**Импорт данных.**

Перед импортом данных в базу данных необходимо подготовить данные в Excel. В данном случае обработаем информацию о продукции, включая их идентификатор продукции, наименование, артикул, минимальная цена и коэффициент. Создадим таблицу с данными о продукции (Рисунок 1).

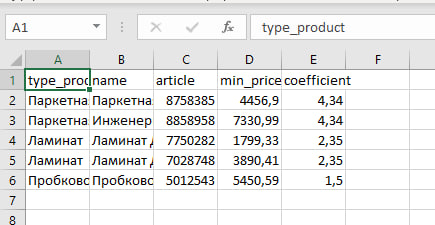


Рисунок 1 – Данные продукции в Excel для импорта

После подготовки данных импортируем их в базу данных.

Перед загрузкой CSV-файла в базу данных PostgreSQL создадим временную таблицу для проверки корректности данных (Рисунок 2).

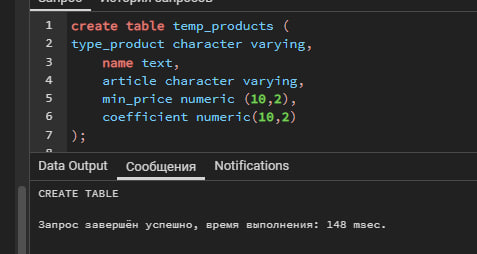


Рисунок 2 - Создание временной таблицы

Используем теперь команду для загрузки данных из CSV-файла в временную таблицу (Рисунок 3).



Рисунок 3 – Команда для загрузки данных

Теперь перенесем данные из временной таблицы в основную таблицу products (Рисунок 4).

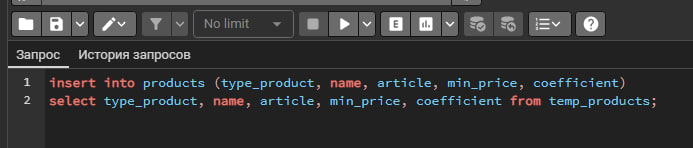


Рисунок 4 - Перенос данных из временной таблицы в основную таблицу

Проверяем данные (Рисунок 5).

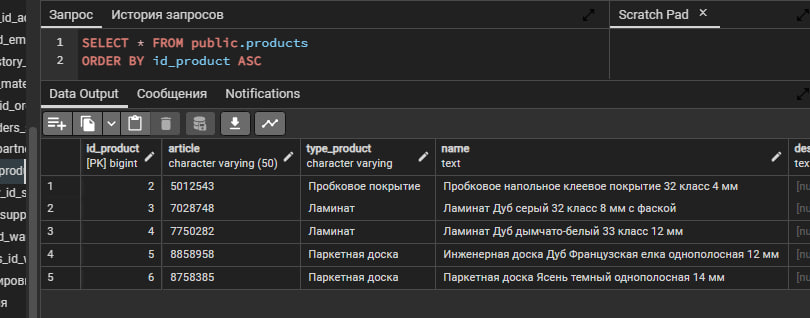


Рисунок 5 – Проверка данных

Теперь импортируем данные для других таблиц. Импортируем данные партнера. Перед импортом данных в базу данных необходимо подготовить данные в Excel (Рисунок 6).

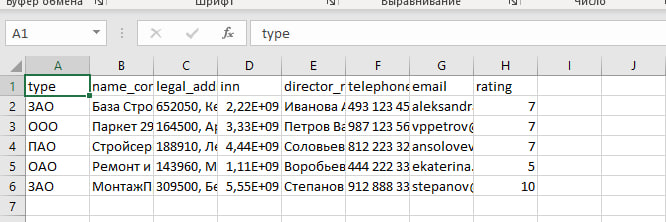


Рисунок 6 – Данные партнера в Excel для импорта

После подготовки данных импортируем их в базу данных.

Используем команду для загрузки данных из CSV-файла в таблицу (Рисунок 7).

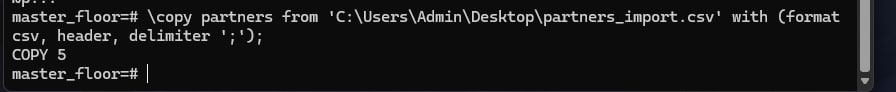


Рисунок 7 – Команда для загрузки данных

Проверяем данные (Рисунок 8).

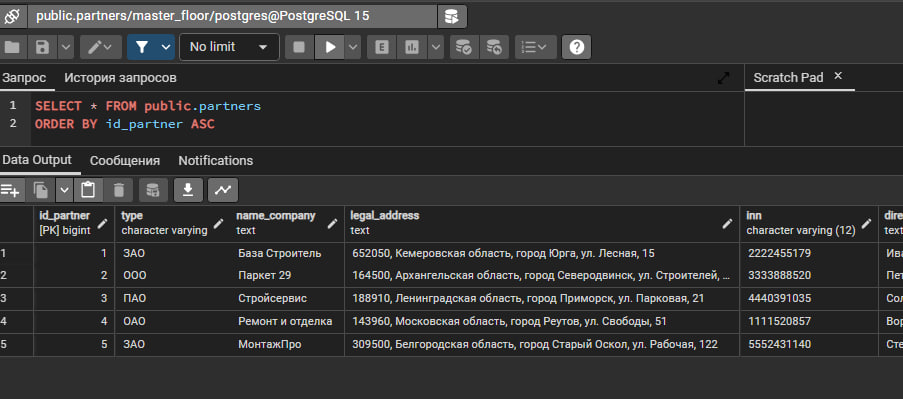


Рисунок 8 – Проверка данных

Импортируем данные истории продаж. Перед импортом данных в базу данных необходимо подготовить данные в Excel (Рисунок 9).

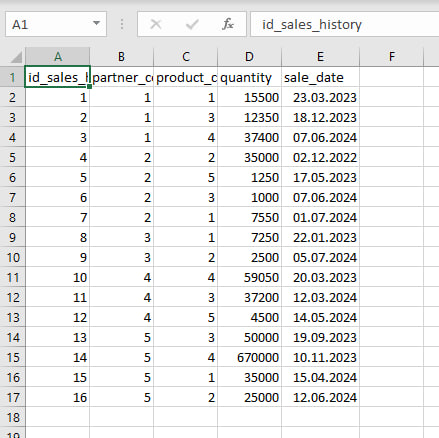


Рисунок 9 – Данные истории продаж в Excel для импорта

После подготовки данных импортируем их в базу данных.

Используем команду для загрузки данных из CSV-файла в таблицу (Рисунок 10).

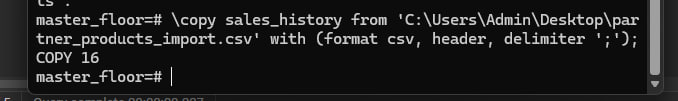


Рисунок 10 – Команда для загрузки данных

Проверяем данные (Рисунок 11).

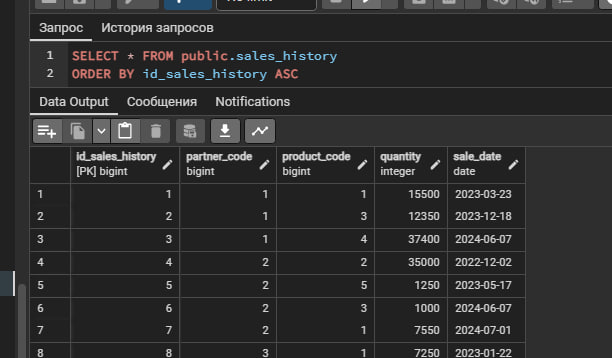


Рисунок 11 – Проверка данных

Теперь импортируем данные материала. Перед импортом данных в базу данных необходимо подготовить данные в Excel (Рисунок 12).

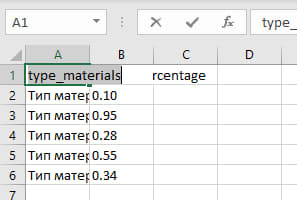


Рисунок 12 – Данные материала в Excel для импорта

После подготовки данных импортируем их в базу данных.

Перед загрузкой CSV-файла в базу данных PostgreSQL создадим временную таблицу для проверки корректности данных (Рисунок 13).

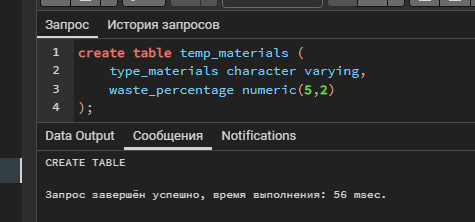


Рисунок 13 - Создание временной таблицы

Используем теперь команду для загрузки данных из CSV-файла в временную таблицу (Рисунок 14).

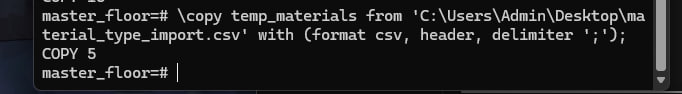


Рисунок 14 – Команда для загрузки данных

Теперь перенесем данные из временной таблицы в основную таблицу materials (Рисунок 15).

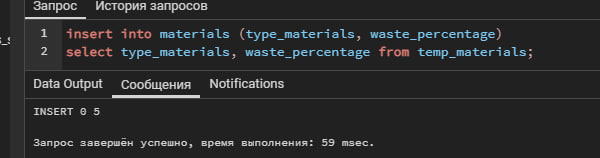


Рисунок 15 - Перенос данных из временной таблицы в основную таблицу

Проверяем данные (Рисунок 16).

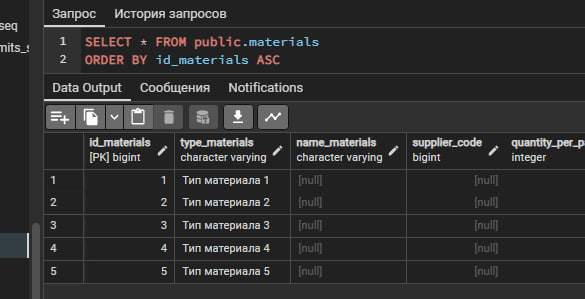


Рисунок 16 – Проверка данных

**Окно «Авторизация».**

В самом начале работы с программой пользователю будет предложена страница авторизации, на которой расположены два текстовых поля для ввода логина и пароля, заголовок, логотип, а также кнопка для входа в систему.

Внешний вид окна авторизации и его элементов предоставлен на   
рисунке 17.

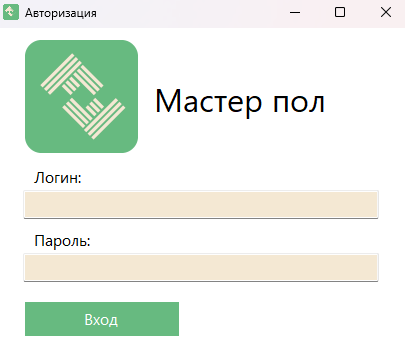


Рисунок 17 – Окно авторизации

**Окно «Главная»**

Данное программное обеспечение позволяет пользователю получать актуальную информацию из базы данных. Окно управления данными партнеров служит для хранения и редактирования информации о партнерах. Оно содержит кнопки для добавления и поиск партнера по наименованию, кнопка для возврата к экрану авторизации, а также панель для отображения информации и текстовое поля для поиска. Внешний вид окна представлен на рисунке 18.

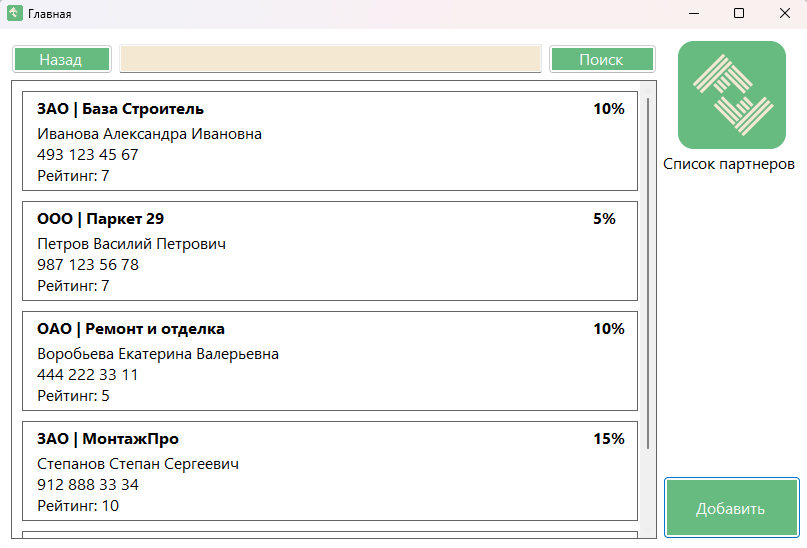


Рисунок 18 – Окно «Главная»

**Окно для добавление нового партнера.**

Чтобы добавить нового партнера, необходимо нажать на кнопку «Добавить». После этого появится окно с текстовыми полями и возможностью вставки рисунка при нажатии на него. Заполнив все необходимые поля, нажимаем на кнопку «Добавить», и новый партнер отобразится в списке.

Внешний вид окна для добавление нового партнера представлен на рисунке 19.

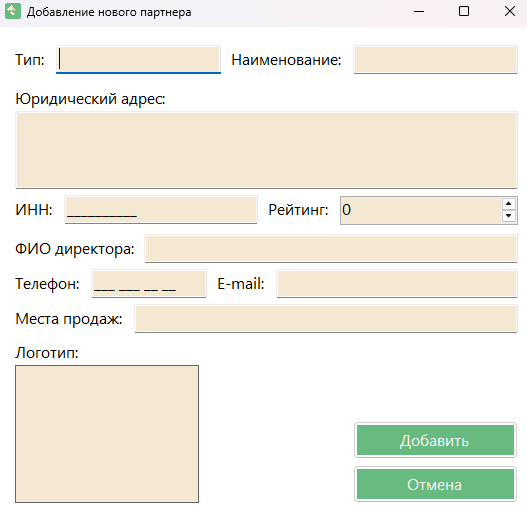


Рисунок 19 – Окно для добавления нового партнера

Чтобы изменить данные партнера, необходимо выбрать его в списке и дважды щелкнуть по нему. После этого откроется окно с текстовыми полями и возможностью вставки изображения, а также будет отображена таблица с историей партнера, в которой будут указаны наименование продукции, количество товара и дата продажи. Также предусмотрена кнопка для удаления партнера. Заполнив все необходимые поля, нажмем на кнопку «Редактировать», и измененный партнер отобразится в списке.

Внешний вид окна для редактирования данных партнера представлен на рисунке 20.

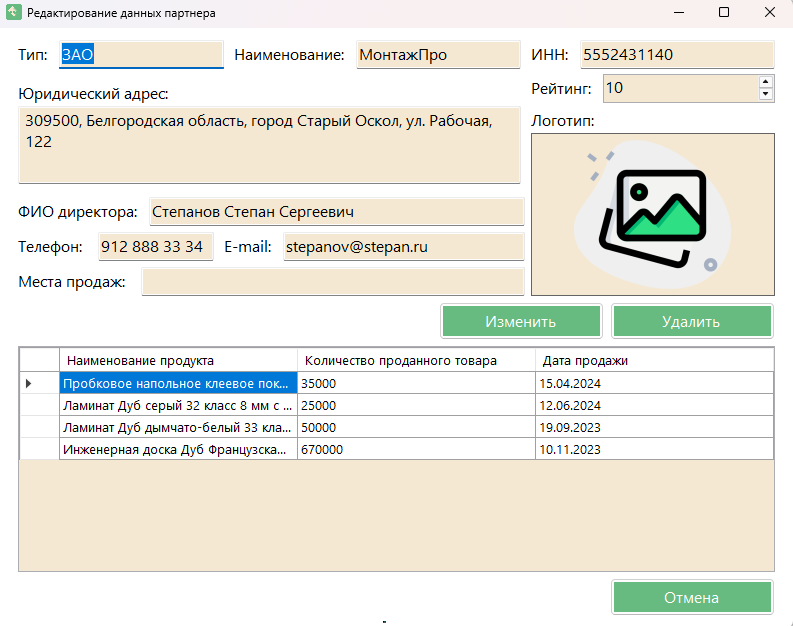


Рисунок 20 - Окно для редактирования данных партнера

**Разработка функциональных модулей.**

На рисунке 21 представлена функция подключения к базе данных.



Рисунок 21 - Реализация подключения к базе данных

Модуль для добавления данных включает в себя функцию, предназначенную для сохранения информации в таблице, что иллюстрируется на рисунке 22.

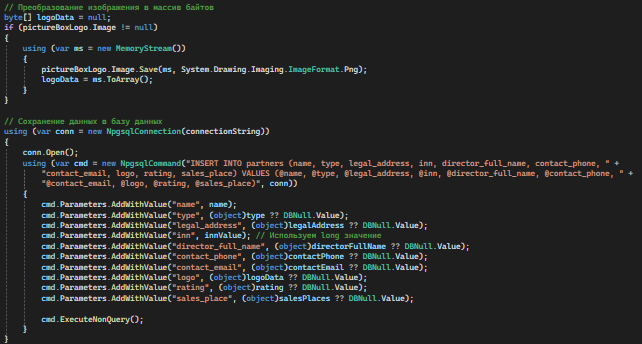


Рисунок 22 - Фрагмент кода для добавления информации в базу данных

Для изменения добавленных данных разработан метод, код который представлен на рисунке 23.

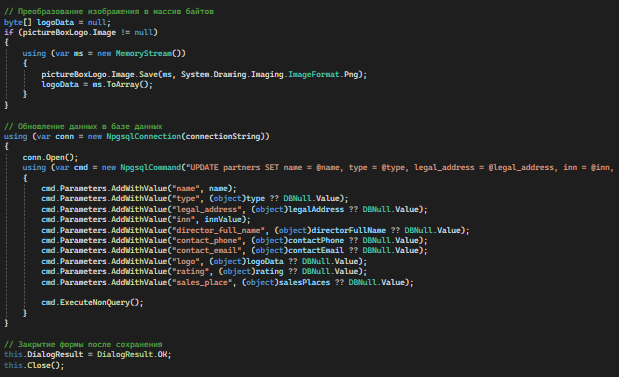


Рисунок 23 - Фрагмент кода для редактирования информации из базы данных

Для просмотра истории реализации продукции партнером существует метод, фрагмент кода которого представлен на рисунке 24.

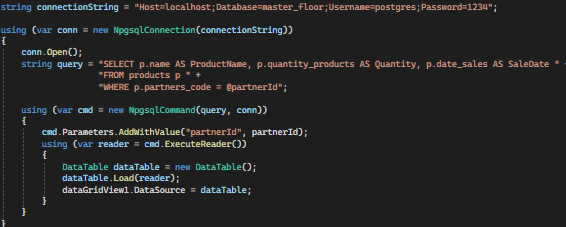


Рисунок 24 – Фрагмент кода для просмотра истории реализации продукции партнером

Для авторизации пользователя в системе разработан метод, код которого представлен на рисунке 25.

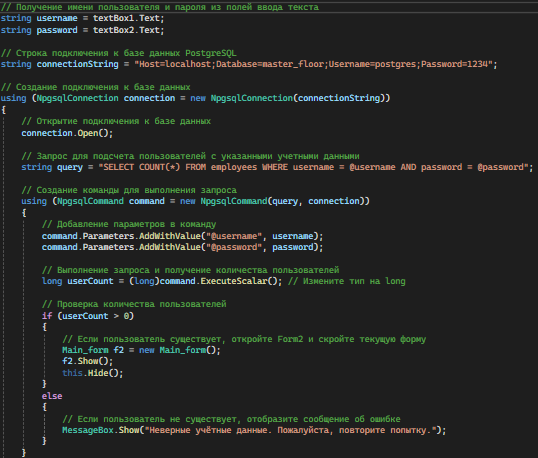


Рисунок 25 - Фрагмент кода для авторизации

Для удаления данных разработан метод, код которого представлен на рисунке 26.

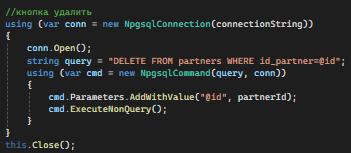


Рисунок 26 - Фрагмент кода для удаления партнера из базы данных

**Отладка приложения.**

Для отладки проекта в Visual Studio выполним следующие шаги:

Для начала запустим Visual Studio и откроем проект, который необходимо отладить.

Установка точек останова: найдем проблемные участки кода и установим точки останова в тех местах, где хотим остановить исполнение программы для анализа.

Для этого наведем курсор на левый край строки возле номера строки кода и щелкнем левой кнопкой мыши (Рисунок 27).

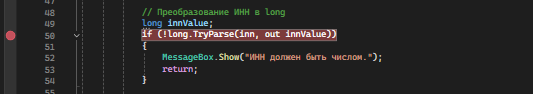


Рисунок 27 - Установка точек останова

Теперь запустим отладку, для этого нажмем клавишу F5 или выберем «Debug» → «Start Debugging» для запуска программы в режиме отладки.

Пошаговое выполнение: используем кнопки управления отладкой (Step Over и Step Into) для пошагового выполнения кода и внимательного наблюдения за значениями переменных.

Исправление ошибок: найдем причины сбоев, критических ошибок или логических ошибок и внести исправления в код (Рисунок 28).

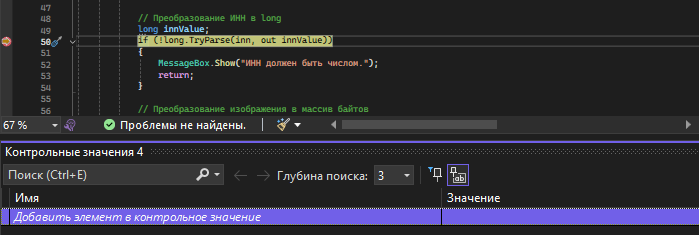


Рисунок 28 – Запуск откладки

После анализа кода следует перейти к рефакторингу:

Выделение функций: разобьём длинные функции на более мелкие, каждая из которых будет выполнять одну конкретную задачу. Это улучшит читаемость и тестируемость кода.

Инструменты рефакторинга Visual Studio: используйте встроенные инструменты рефакторинга. Для этого щелкнем правой кнопкой мыши на функции или переменной и выберите соответствующий пункт меню, например, Refactor, чтобы переименовать, переместить или изменить структуру кода (Рисунок 29).

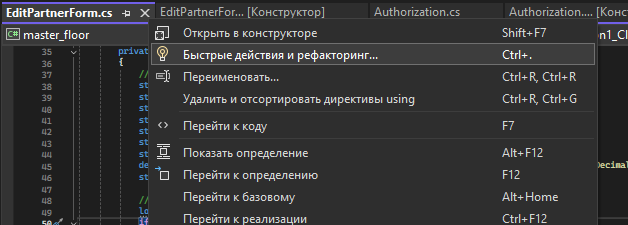


Рисунок 29 – Использование рефакторинга для преобразования и очистки кода.

На этом этапе необходимо провести оптимизацию кода:

Удаление неиспользуемых переменных и функций: проверим код на наличие переменных и функций, которые не используются, и удалим их, чтобы уменьшить объем кода и повысить его чистоту.

Анализ алгоритмов: оценим эффективность используемых алгоритмов и, при необходимости, заменим их на более оптимальные решения, что может значительно улучшить производительность приложения.

**Вывод:** разработка подсистемы для работы с партнерами завершена успешно. Создана и настроена база данных, импортированы данные, разработан интуитивно понятный интерфейс и основные функциональные модули (просмотр, добавление, редактирование, история, поиск, удаление). Проведена отладка, повышена надежность. Подсистема готова к использованию и дальнейшему развитию.