**МИНИСТЕРСТВО ОБРАЗОВАНИЯ МОСКОВСКОЙ ОБЛАСТИ**

**Государственное бюджетное профессиональное образовательное учреждение Московской области «Люберецкий техникум имени Героя Советского Союза, летчика-космонавта Ю.А.Гагарина»**

**ОТЧЕТ**

**УЧЕБНОЙ ПРАКТИКИ**

\_\_\_\_\_\_\_ТЕПЛУХИН МАКАР АЛЕКСЕЕВИЧ\_\_\_\_\_\_

(Фамилия, имя, отчество студента)

по профессиональному модулю

\_\_\_\_\_\_\_\_*ПМ.01 «Разработка модулей программного обеспечения*\_\_\_\_\_\_\_\_

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ для компьютерных систем»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

Специальность \_*09.02.07 "Информационные системы и программирование"*\_

Код, название

Курс \_*4*\_ Группа № \_*185*\_

Период практики с «\_*25*\_» \_\_*ноября*\_\_ 20*21* г. по «\_*30*\_» \_*ноября*\_ 20*21*г.

Руководитель практики

от техникума Жирнова Ю.В. /\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

Люберцы 2021

**Оглавление**

**1. Индивидуальное задание на практику………………...……….3**

**2.Дневник прохождения практики………………………..……….5**

**3. Введение…………………………………………………………….[7](#_Введение)**

**4. Изучение предметной области…………………………………...[9](#_Изучение_предметной_области)**

**5. Изучение программных средств………………………………...**[**12**](#_Изучение_программных_средств)

**6. Практическая часть………………………………………………**[**12**](#_Изучение_программных_средств)

**7. Выводы……………………………………………………………..**[**29**](#_Выводы)

**8. Список литературы……………………………………………….**[**30**](#_Литературные_источники)

# 

# Введение

Практика – это одна из важнейших составляющих профессиональной

подготовки специалистов. Учебная практика направлена на то, чтобы перенести полученные знания и умения из области теории в область повседневной профессиональной деятельности, на развитие экономического сознания студентов, развитие аналитических способностей, а также на самостоятельное решение выполнения функций управления.

Целью учебной практики является разработка системы для производственной компании, которая ежедневно выпускает продукцию в большом объеме, а также реализует ее агентам, которые в дальнейшем продают эти товары конечным потребителям.

Для достижения данной цели требуется решить следующие задачи:

1. Описание предметной области
2. Изучение программных средств
3. Восстановление базы данных из скрипта
4. Импорт данных
5. Разработка desktop-приложения
6. Чтение, добавление, редактирование, удаление данных
7. Сортировка, фильтрация, поиск
8. Разработка API
9. Кроссплатформенная мобильная разработка XAMARIN.FORMS
10. Загрузка файлов и описания проекта на репозиторий системы контроля версий git.

В ходе учебной практики планируется освоение следующего вида деятельности: «Разработка модулей программного обеспечения для компьютерных систем » (ПК):

ПК 1.1 Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.

ПК 1.2 Разрабатывать программные модули в соответствии с техническим заданием.

ПК 1.3 Выполнять отладку программных модулей с использованием специализированных программных средств

ПК 1.4 Выполнять тестирование программных модулей

ПК 1.5 Осуществлять рефакторинг и оптимизацию программного кода

ПК 1.6 Разрабатывать модули программного обеспечения для мобильных платформ.

# Изучение предметной области

Компания “Глазки-save” занимается производством защитных очков для различных сфер использования. Она реализует продукцию не напрямую покупателям, а работает с агентами, которые уже впоследствии продают продукцию конечному потребителю. Для удобного взаимодействия с агентами и контроля их работы требуется разработать систему, позволяющую аккумулировать всю информацию по агентам.

Система должна включать в себя следующий функционал:

• просмотр списка агентов,

• добавление/удаление/редактирование данных об агенте,

• управление историей реализации продукции агентом.

Microsoft SQL Server — система управления реляционными базами данных (РСУБД), разработанная корпорацией Microsoft. Основной используемый язык запросов — Transact-SQL, создан совместно Microsoft и Sybase. Transact-SQL является реализацией стандарта ANSI/ISO по структурированному языку запросов (SQL) с расширениями. Используется для работы с базами данных размером от персональных до крупных баз данных масштаба предприятия; конкурирует с другими СУБД в этом сегменте рынка.

Развитие клиент-серверных технологий во второй половине 80-х было обусловлено развитием двух ключевых направлений, активно разрабатываемых с конца 70-х годов: персональных компьютеров с одной стороны, и компьютерных сетей — с другой. Долгое время СУБД были доступны лишь для мэйнфреймов, и лишь благодаря росту производительности процессоров для домашних компьютеров и мини-ЭВМ разработчики СУБД (как, например, Oracle) начали создавать соответствующие версии своих продуктов. Одной из первых СУРБД для ПК стала Oracle v3, выпущенная в 1983 году. На тот момент немногочисленные владельцы ПК использовали их в основном для разработки приложений и тестирования[1]. Одним из ключевых этапов в развитии СУБД стал 1986 год. К этому времени появилось ещё несколько компаний-разработчиков СУБД, одной из самых заметных из них стала компания Sybase, основанная двумя годами ранее. К 1986 году Sybase начала комплектовать интеллектуальные рабочие станции (как правило, разработки Sun Microsystems или Apollo Computer) с серверами базы данных (разработанных, например, Oracle). При этом сама клиент-серверная технология сделала возможным отделение модулей обработки информации (т. н. back end) от модулей интерфейса (т. н. front end). Учтя постоянный рост проникновения компьютерных сетей, поставщики решений перешли к задачам распределения остальных задач (например, форматирование отчётов, проверка данных и т. д.) среди рабочих станций сети, оставив серверу выполнять лишь задачи, требующие централизованного решения (хранение и защита данных, оптимизация потока выполнения запросов и т. д.)[1].

C# (произносится си шарп) — объектно-ориентированный язык программирования. Разработан в 1998—2001 годах группой инженеров компании Microsoft под руководством Андерса Хейлсберга и Скотта Вильтаумота[7] как язык разработки приложений для платформы Microsoft .NET Framework. Впоследствии был стандартизирован как ECMA-334 и ISO/IEC 23270.

C# относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java. Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов (в том числе операторов явного и неявного приведения типа), делегаты, атрибуты, события, переменные, свойства, обобщённые типы и методы, итераторы, анонимные функции с поддержкой замыканий, LINQ, исключения, комментарии в формате XML

Переняв многое от своих предшественников — языков C++, Delphi, Модула, Smalltalk и, в особенности, Java — С#, опираясь на практику их использования, исключает некоторые модели, зарекомендовавшие себя как проблематичные при разработке программных систем, например, C# в отличие от C++ не поддерживает множественное наследование классов (между тем допускается множественная реализация интерфейсов).

C# разрабатывался как язык программирования прикладного уровня для CLR и, как таковой, зависит, прежде всего, от возможностей самой CLR. Это касается, прежде всего, системы типов C#, которая отражает BCL. Присутствие или отсутствие тех или иных выразительных особенностей языка диктуется тем, может ли конкретная языковая особенность быть транслирована в соответствующие конструкции CLR. Так, с развитием CLR от версии 1.1 к 2.0 значительно обогатился и сам C#; подобного взаимодействия следует ожидать и в дальнейшем (однако, эта закономерность была нарушена с выходом C# 3.0, представляющего собой расширения языка, не опирающиеся на расширения платформы .NET). CLR предоставляет C#, как и всем другим .NET-ориентированным языкам, многие возможности, которых лишены «классические» языки программирования. Например, сборка мусора не реализована в самом C#, а производится CLR для программ, написанных на C# точно так же, как это делается для программ на VB.NET, J# и др. Из-за технических ограничений на отображение (стандартные шрифты, браузеры и т. д.), а также из-за того, что знак диеза ♯ не представлен на стандартной клавиатуре компьютера, при записи имени языка программирования используют знак решётки (#). Это соглашение отражено в Спецификации языка C# ECMA-334. Тем не менее, на практике «Майкрософт» использует знак диеза.

# Изучение программных средств

* + - 1. Восстановление базы данных из скрипта

Заходим в папку, выбираем скрипт и заходим в него (Рисунок 1)

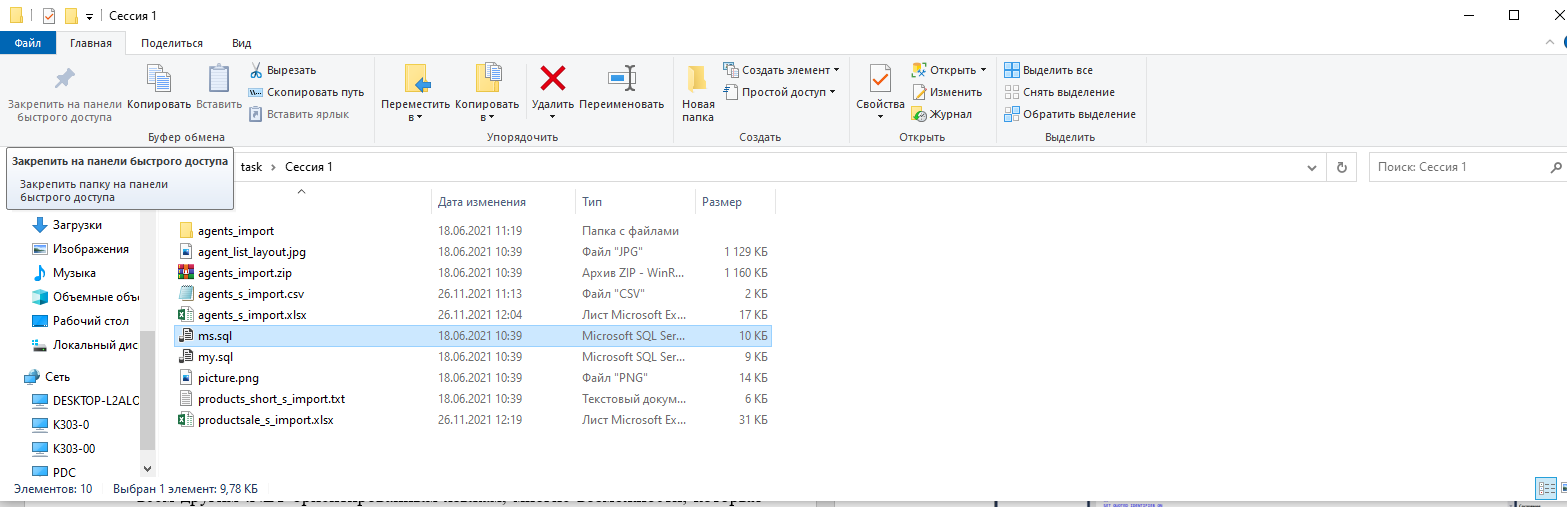


Рисунок 1. Выбор скрипта MYSQL

Вводим логин и пароль базы данных и выполняем скрипт(Рисунок 1.1)

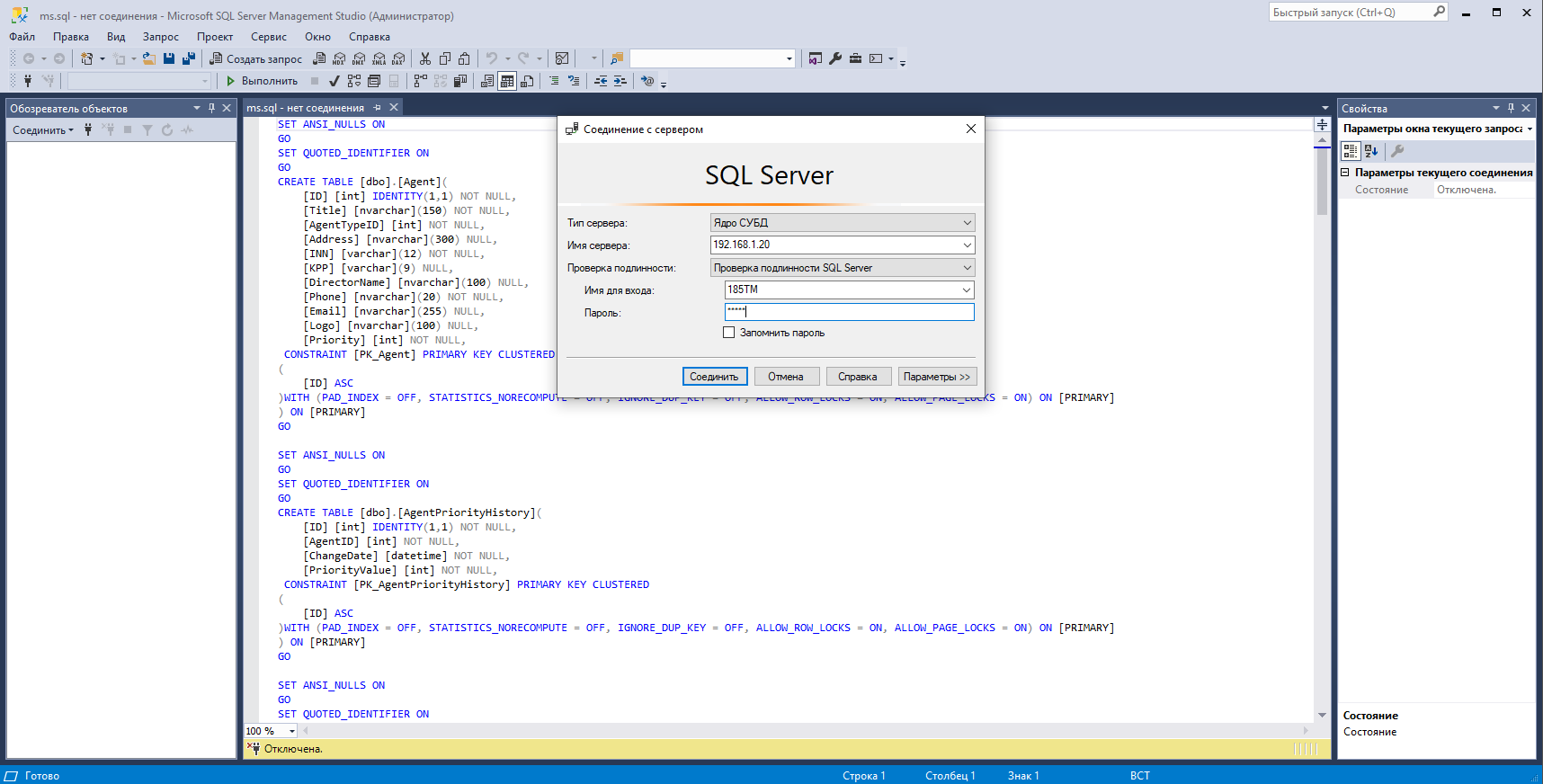


Рисунок 1.1.(Вход в SQL Server)

База данных создана, создаем диаграмму отношений(Рисунок 1.2)

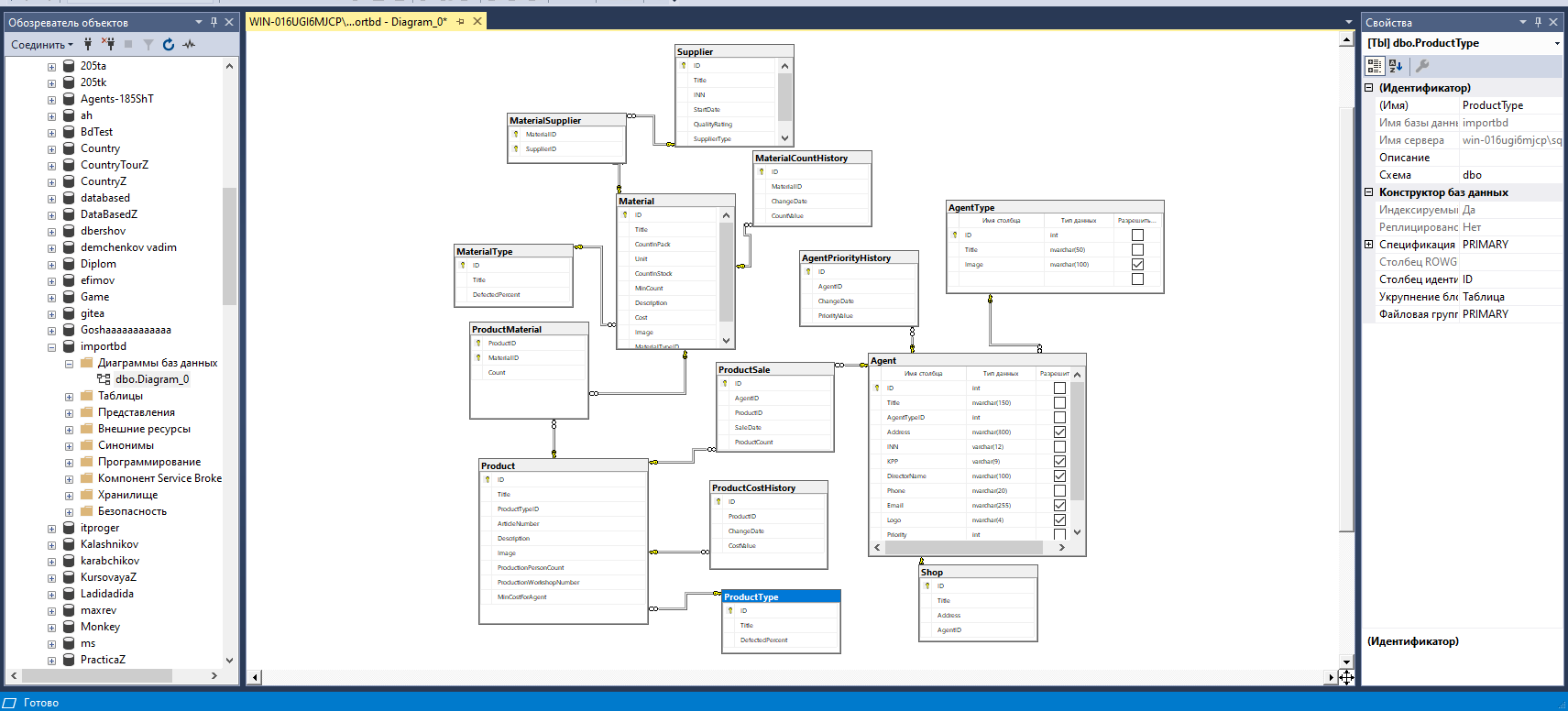


Рисунок 1.2. Диаграмма отношений

* + - 1. Импорт данных

Имеющиеся файлы Excel приводим к нормальному виду для импорта в БД (Рисунок 2)



Рисунок 2. Нормальный вид данных для импорта в БД

Данные из одной таблицы переставляем в соответствующие ячейки второй с помощью функции ВПР(Рисунок 2.1)

В нашем случае выбираем столбец со всеми типами продуктов(E1), затем из этих типов убираем повторяющиеся (A), добавляем ему порядковый номер и выделяем(B), выбираем номер столбца(2) и интервальный просмотр-ложь(0)

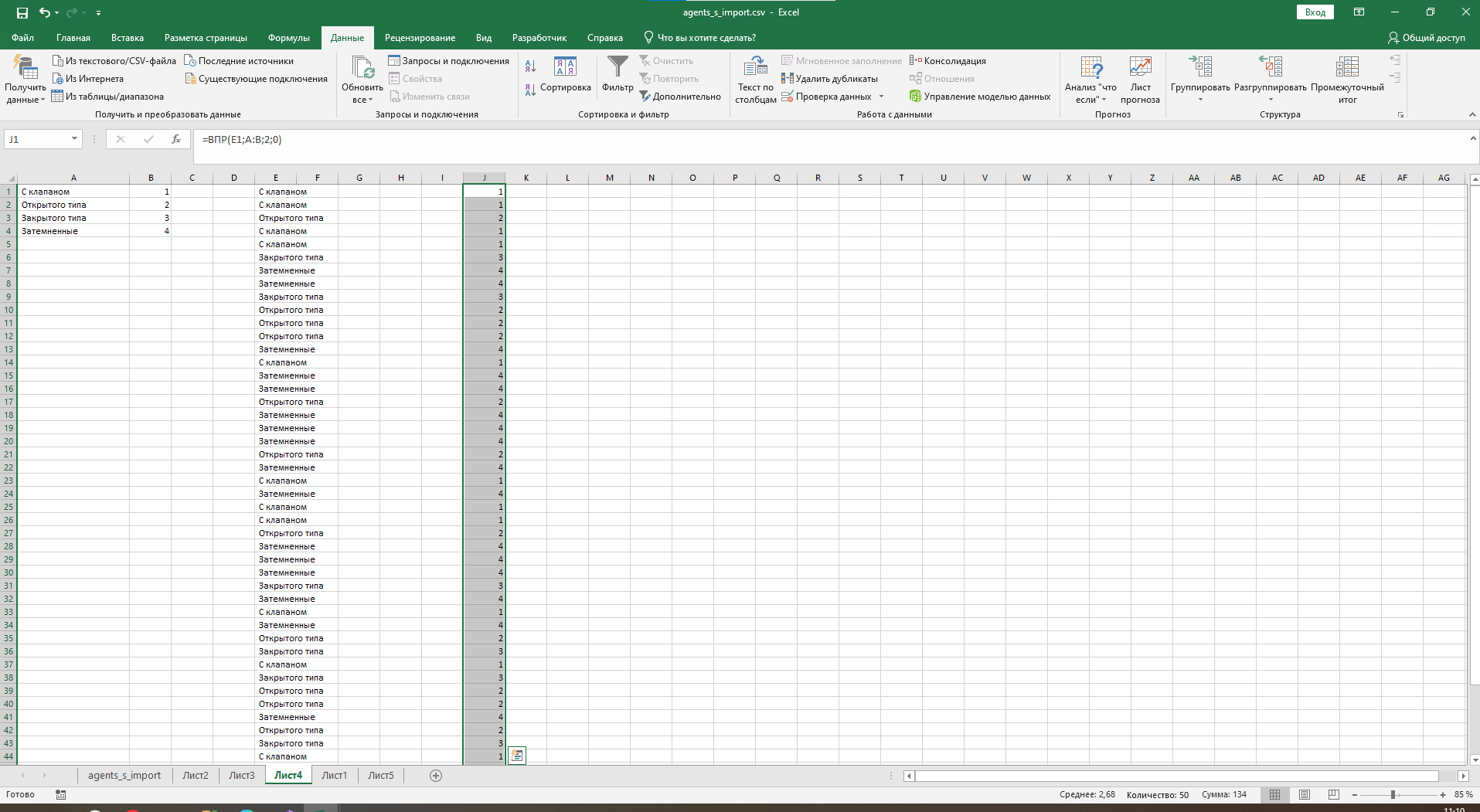


Рисунок 2.1. Функция ВПР

Продолжаем заполнять данные для импорта(Рисунок 2.2, Рисунок 2.3)

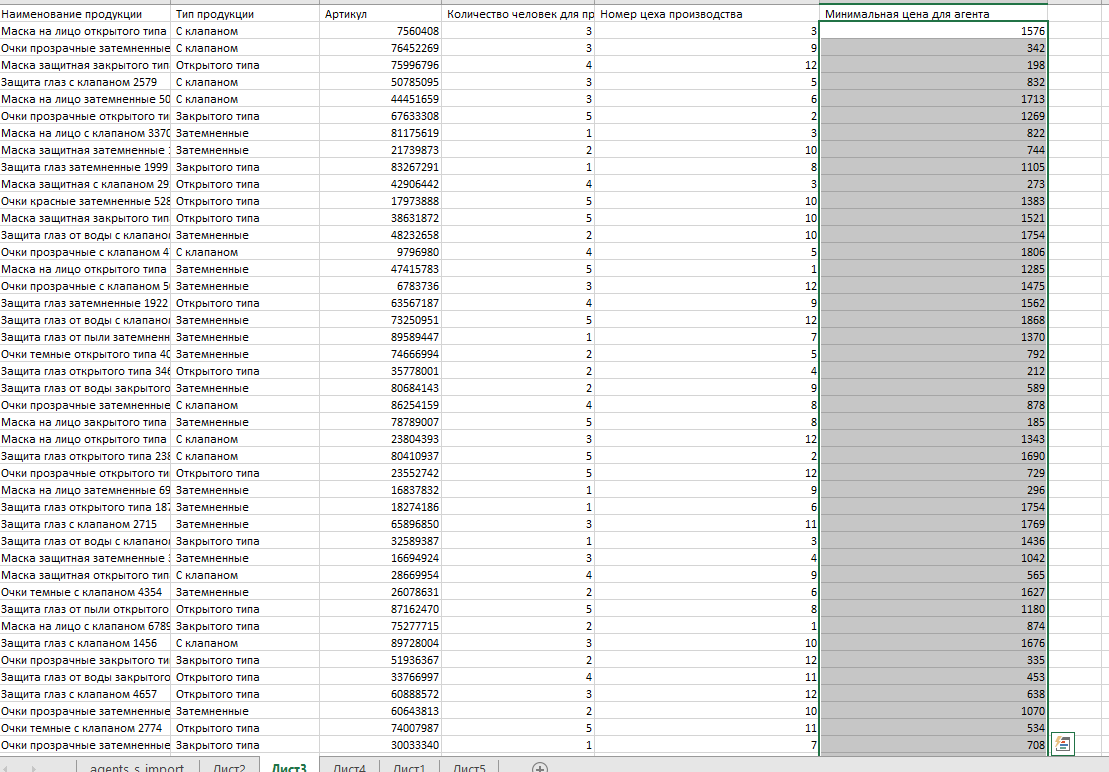


Рисунок 2.2. Продукция

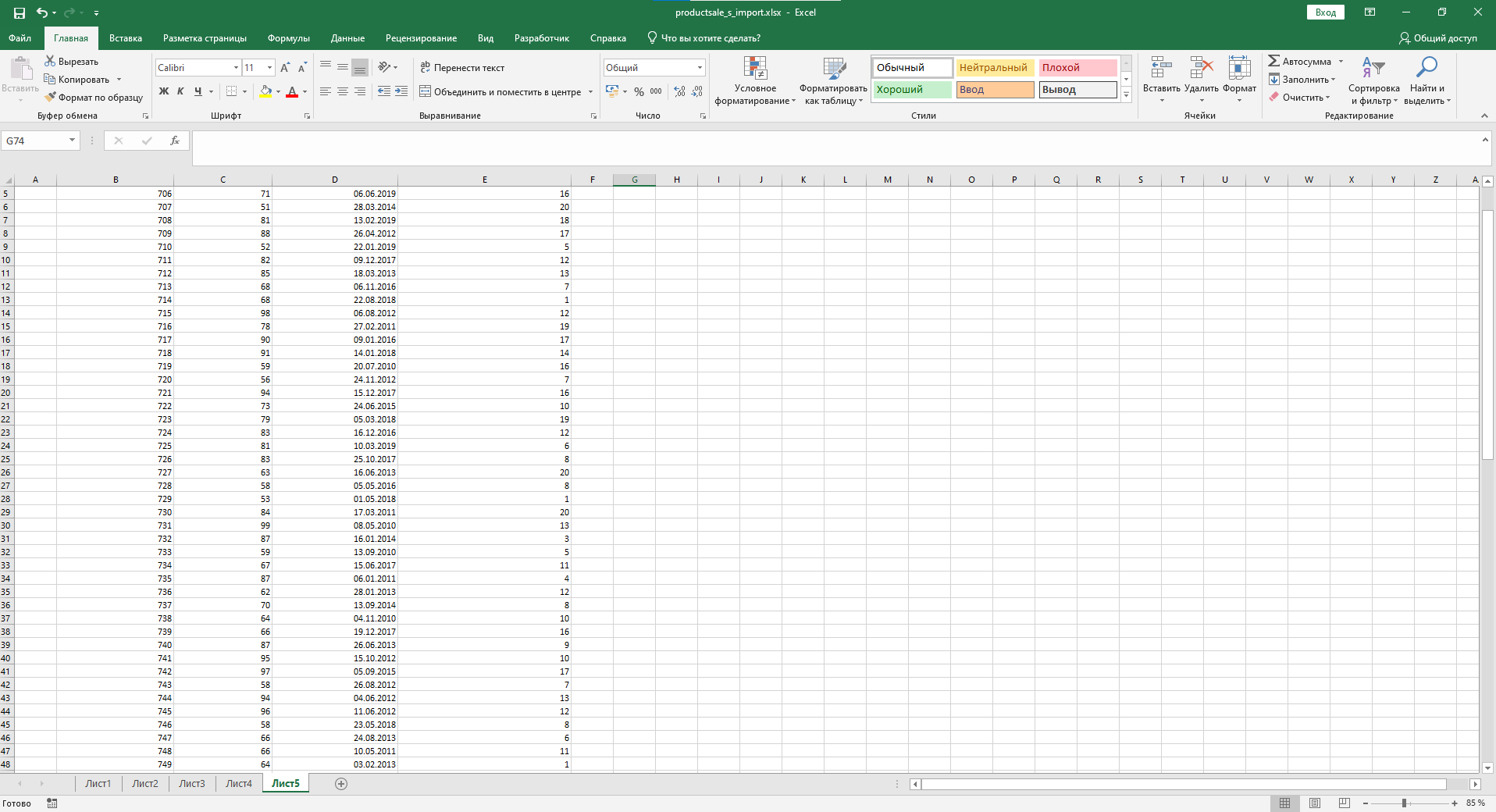


Рисунок 2.3. Нормальный вид данных для импорта

**Чтение, добавление, редактирование, удаление данных**

1. **Добавление.**

На странице AddAgent добавим новое поле, которое будет хранить в себе экземпляр добавляемого отеля. (Рисунок 3)

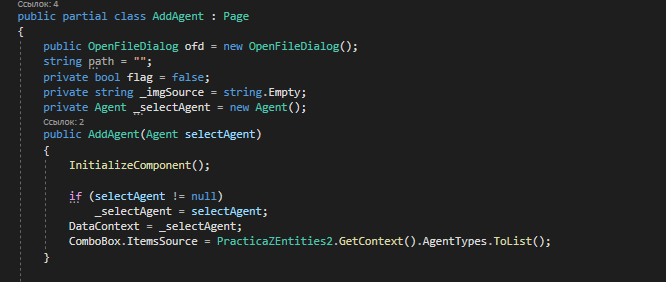
****

Рисунок 3. Поле AddAgent

Затем, используя привязку данных, укажем, какому свойству обращаться к каждому элементу при загрузке данных. Например, свойство Text у первого TextBox'а будет обращаться к названию отеля. (Рисунок 3.1)

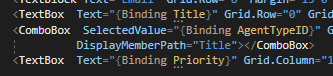
****

Рисунок 3.1. Привязка данных

Далее обработаем нажатие на кнопку Сохранение и в коде пропишем логику обращения к модели данных и добавления нового экземпляра отеля. (Рисунок 3.4)

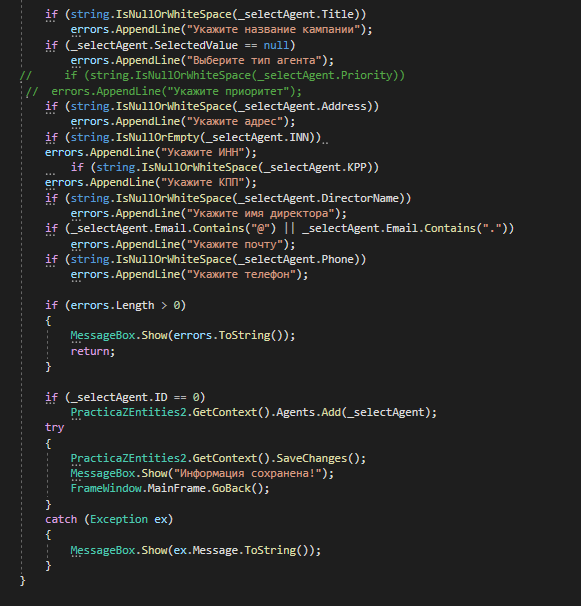


Рисунок 3.4. Добавление нового экхемпляра

1. **Редактирование.**

В первую очередь добавим параметр нашей странице AddAgent. В нее мы будем передавать экземпляр выбранного отеля и, если он не пустой, присваиваем нашему полю selectAgent.

Вызов страницы AddAgent требует какого-то аргумента. При этом для редактирования BtnEdit мы уже будем передавать экземпляр, прописав для этого код. (Рисунок 4)

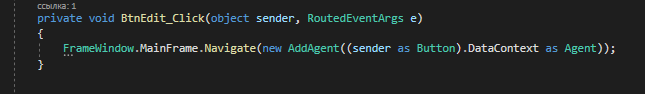
****

Рисунок 4. Вызов страницы AddAgent

1. **Удаление.**

Выбираем все элементы, которые мы выделили, преобразуем их в список отелей. И затем, в сообщении, будем спрашивать пользователя: «Вы точно хотите удалить следующие AgentsRemoving.Count() элементов?». Укажем здесь заголовок сообщения — «Внимание», затем укажем, какие кнопки доступны при диалоге с пользователем: «Да» или «Нет», и выберем изображение — «Question». (Рисунок 5)

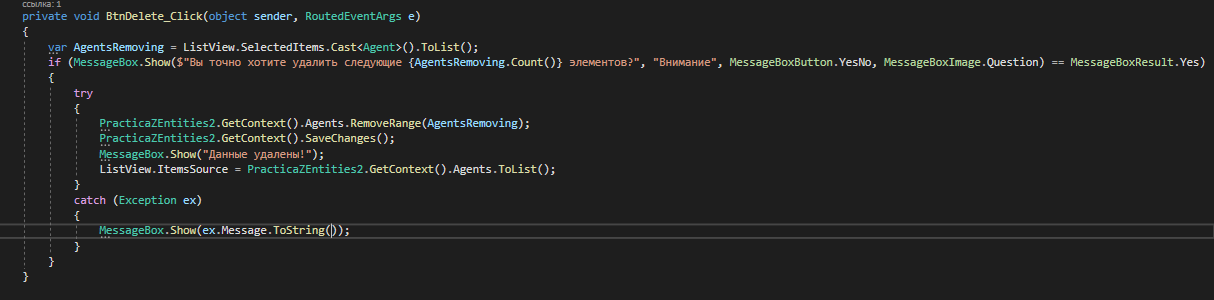
****

Рисунок 5. Удаление

1. **Сортировка, фильтрация, поиск**

Добавляем Update и в скобках пишем string sort(сортировка), filt(фильтрация), search(поиск), чтоб они могли отсылаться к обновлению страницы.

Добавляем в первый цикл search и прописываем, что он должен выводить таблицу Agent из столбцов Email и Phone в программу, и с помощью этого находить нужный нам телефон или почту через TextBox.

Добавляем второй цикл filt, прописываем, что мы берем из таблицы AgentType столбец Title и выводим данные в ComboBox, а «Все типы» - это фильтрация с данными таблицы.

Добавляем третий цикл sort, прописываем также несколько циклов и создаем для них текст, который должен быть одинаков как в ComboBoxItem, если надо вывести по убыванию пишем-OrderBy, если по возрастанию – OrderByDescending, также указываем путь к столбцу.( Рисунок 6, рисунок 6.1)

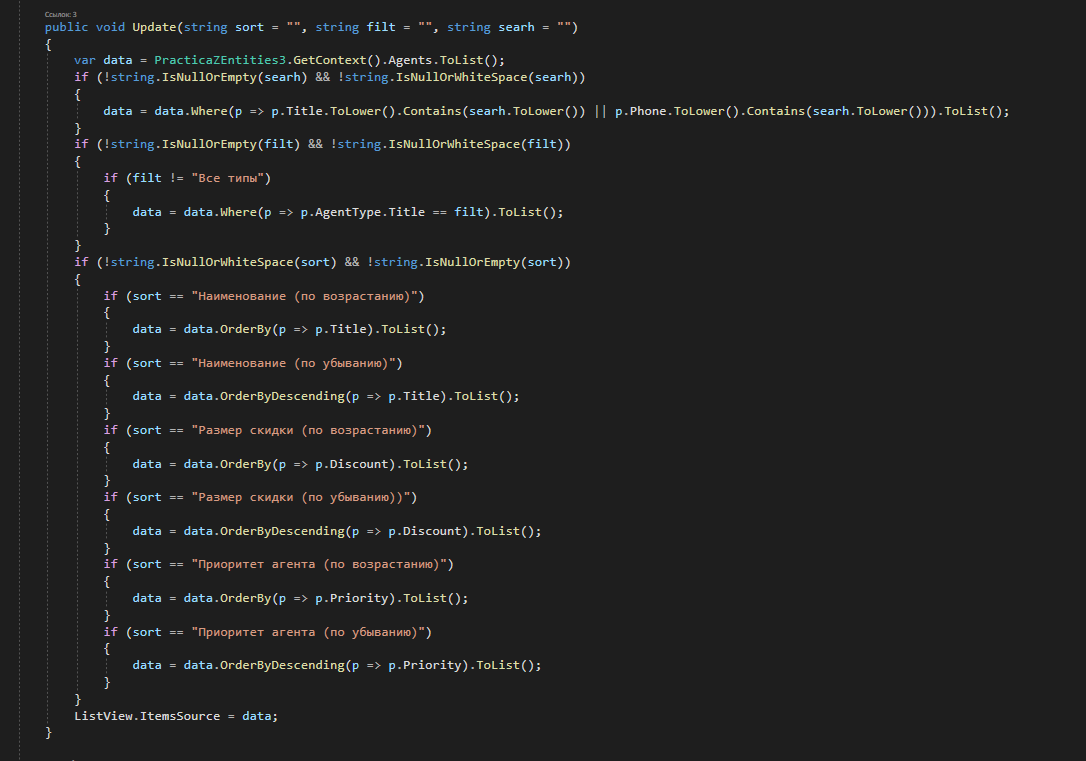
****

Рисунок 6. Сортировка, фильтрация, поиск

Создаем для TypeBox(поисковик) и двум ComboBox – SelectionChanged, чтоб мы могли прописать Update и обновлять страницу без потерь данных возвращаясь к содержимому.

В Test используется Keyboard для TextBox(поисковик), чтоб он не выводил каждый раз «Ничего не найдено» при нажатий каждой клавиши, то используется if(e.key == e.Enter) и тогда можно будет

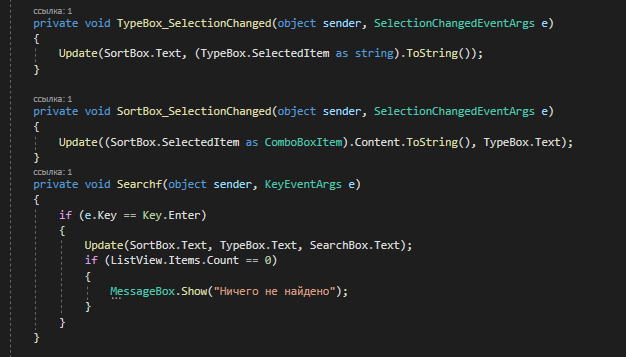
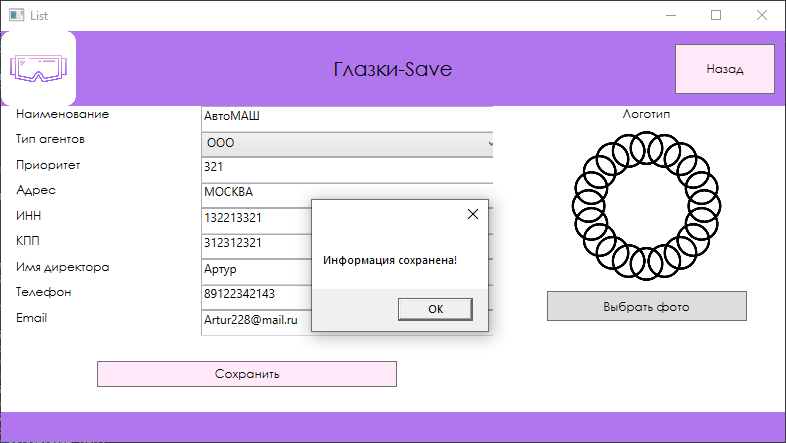
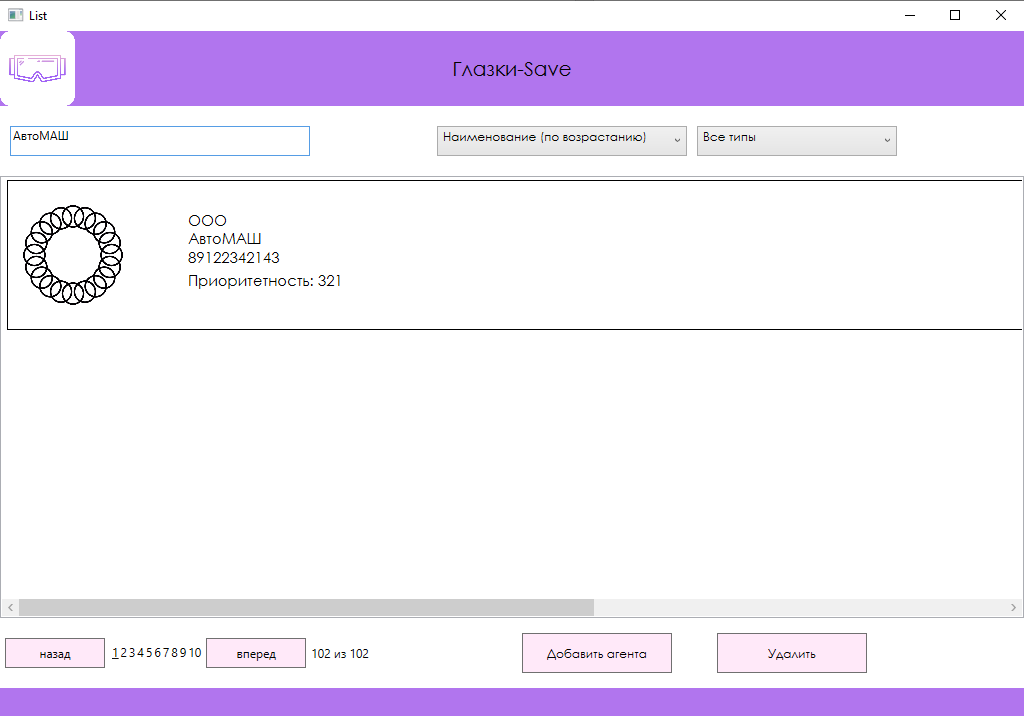
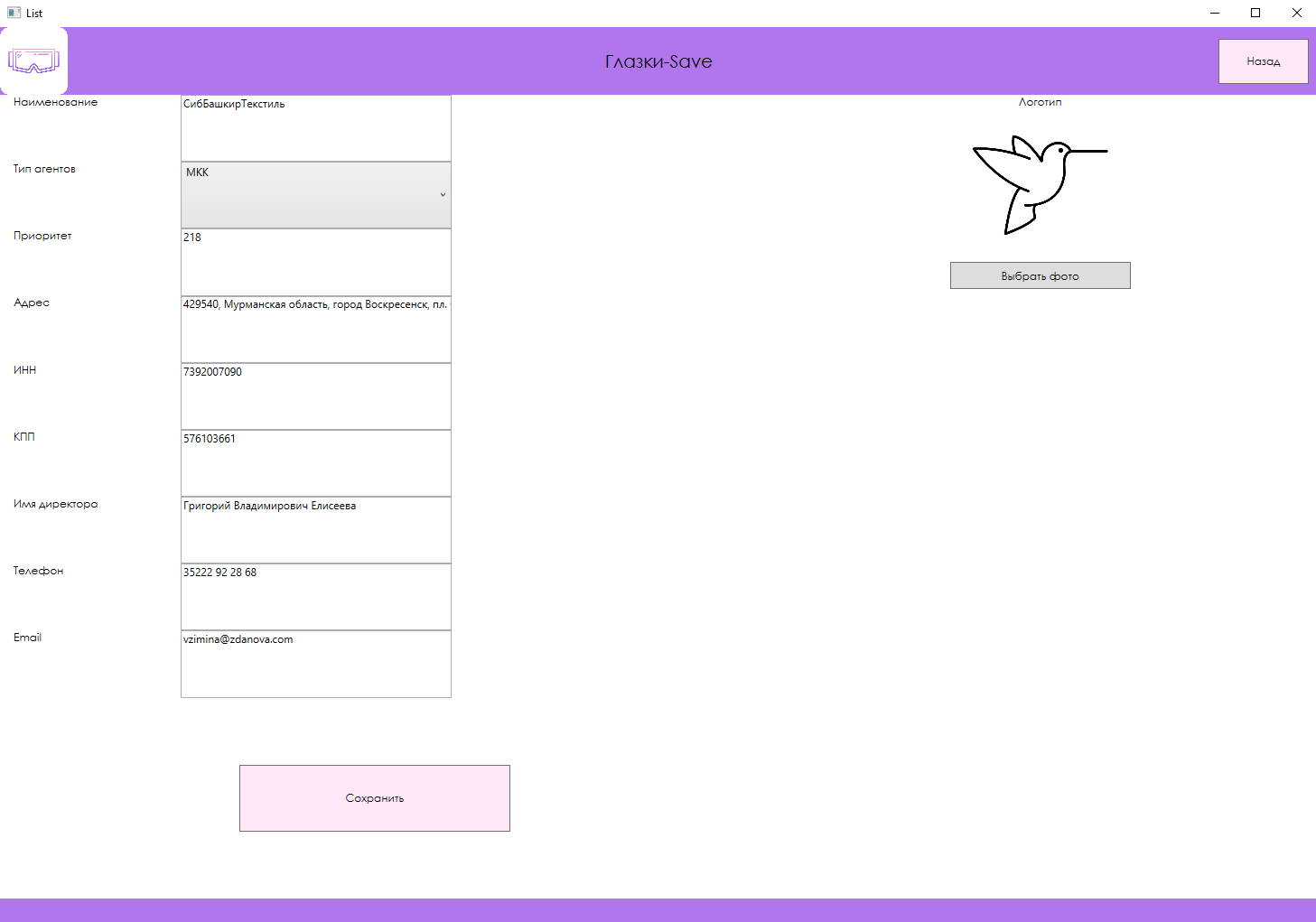
****

Рисунок 6.1. (Сортировка, фильтрация, поиск)

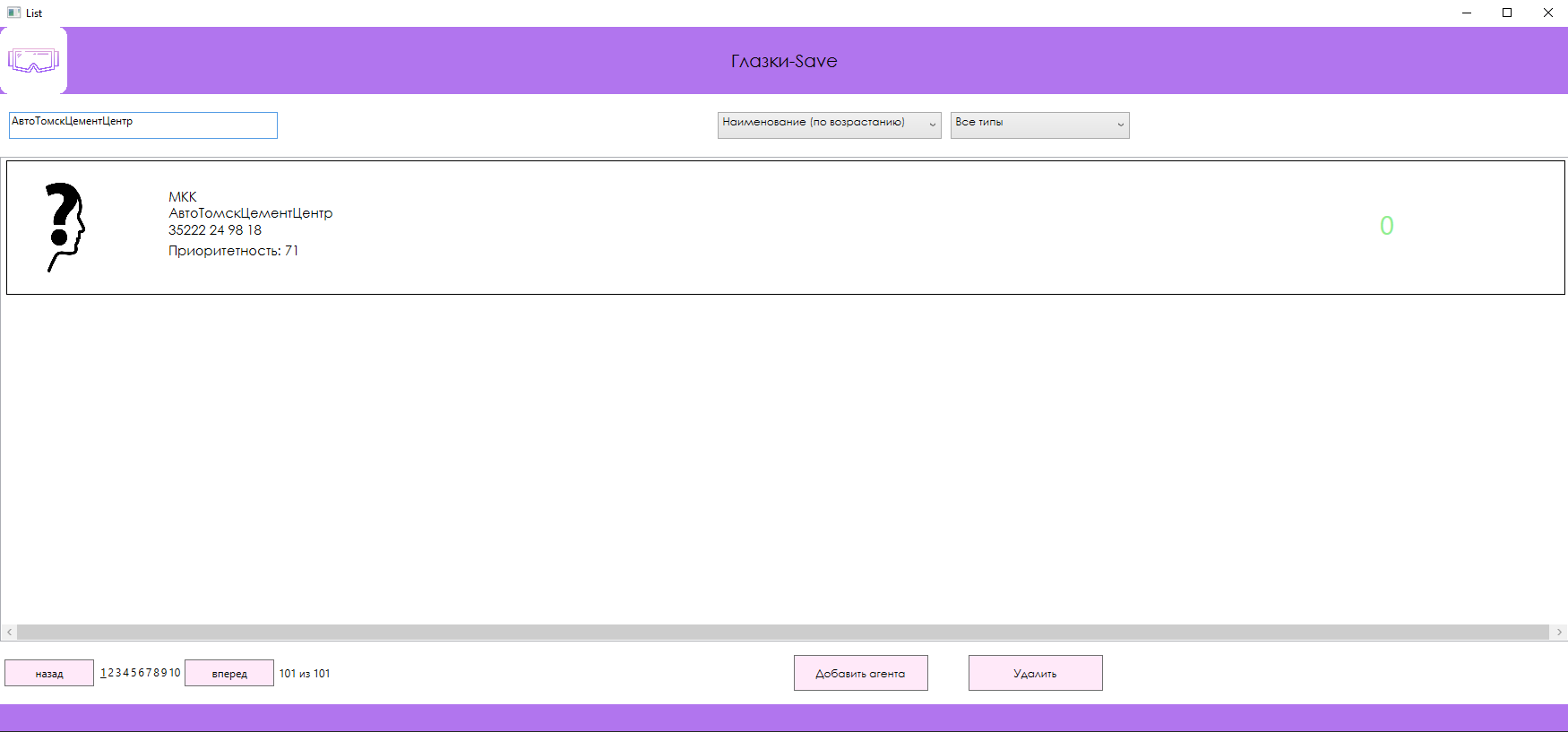




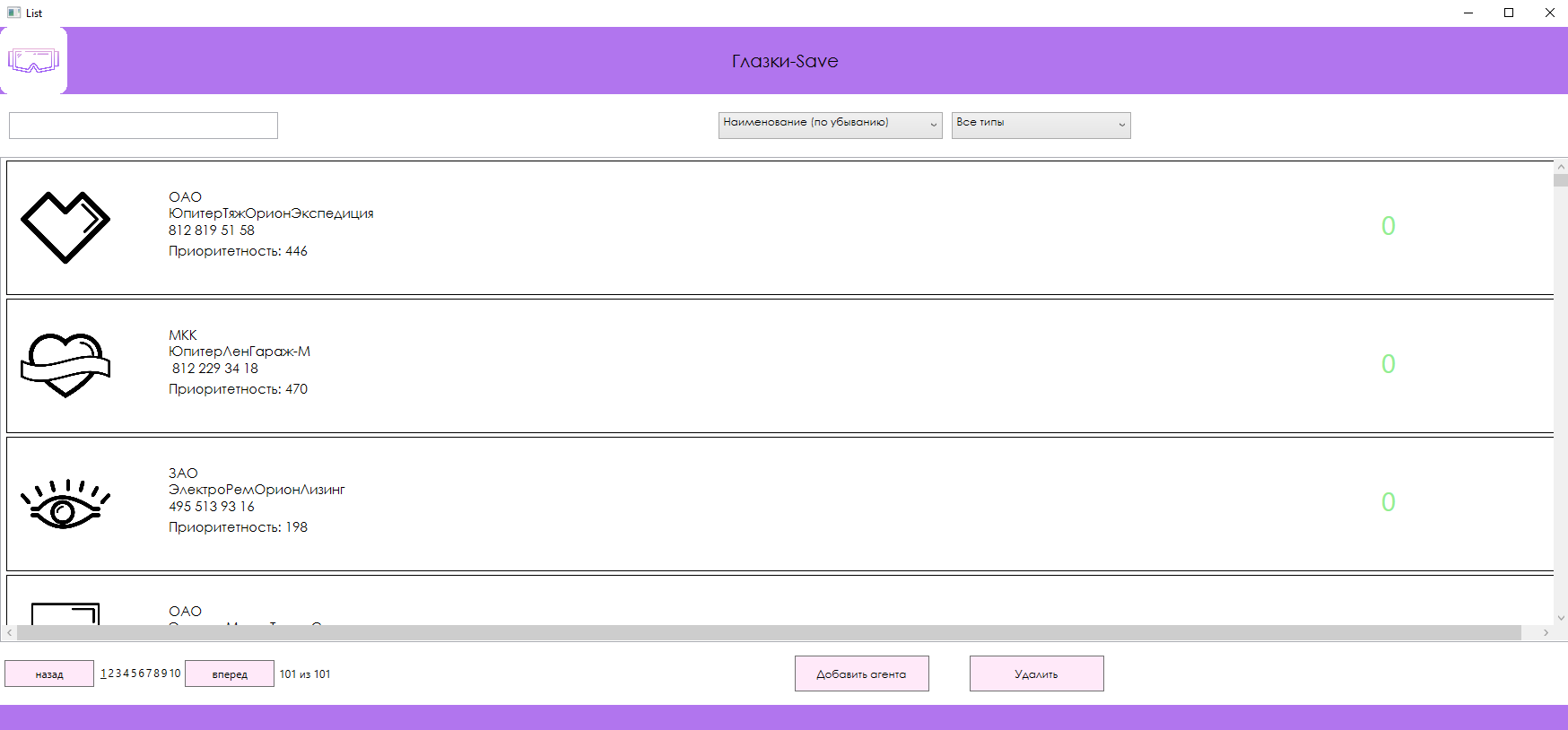
Добавление

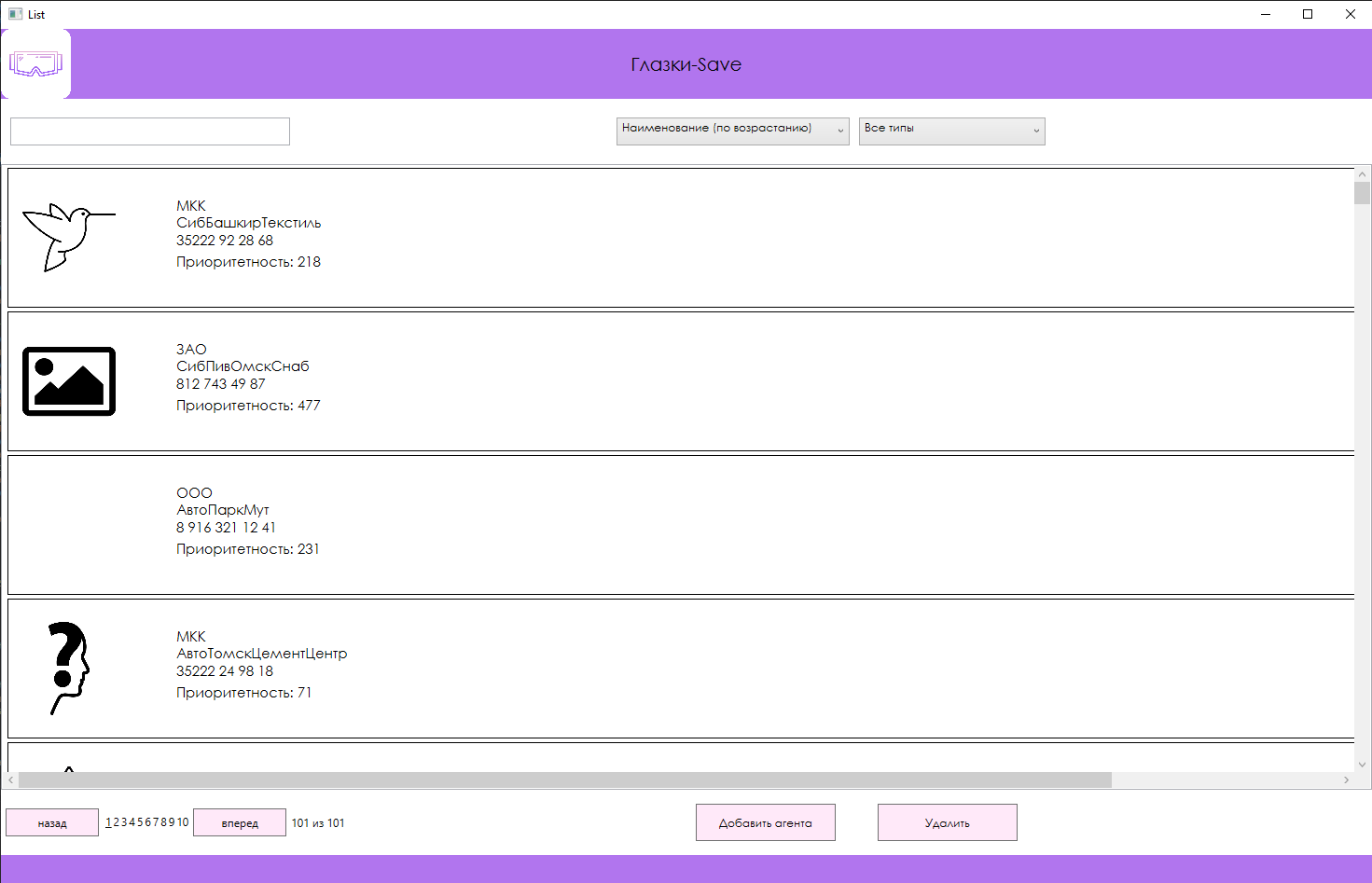


Редактирование

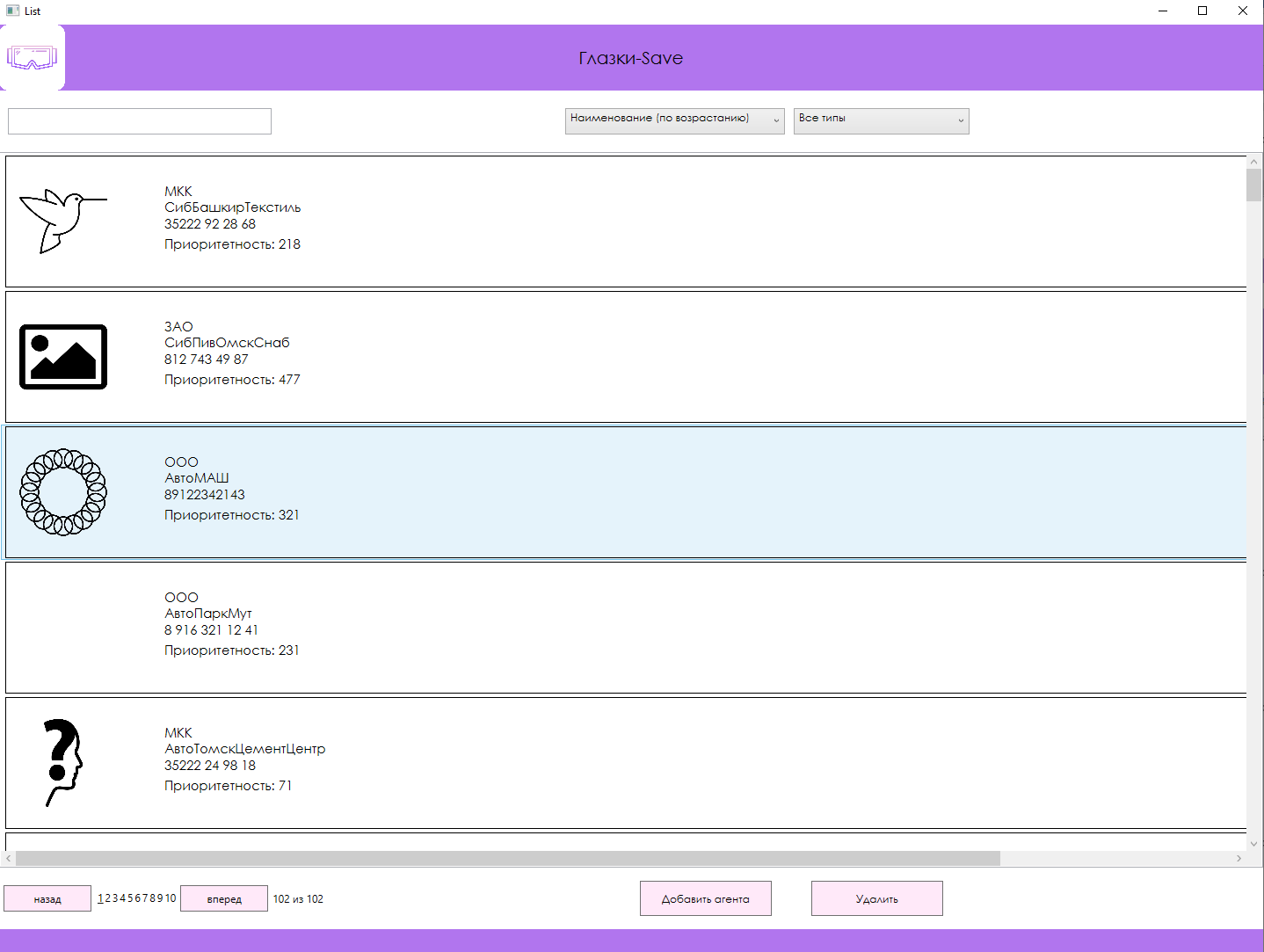
****

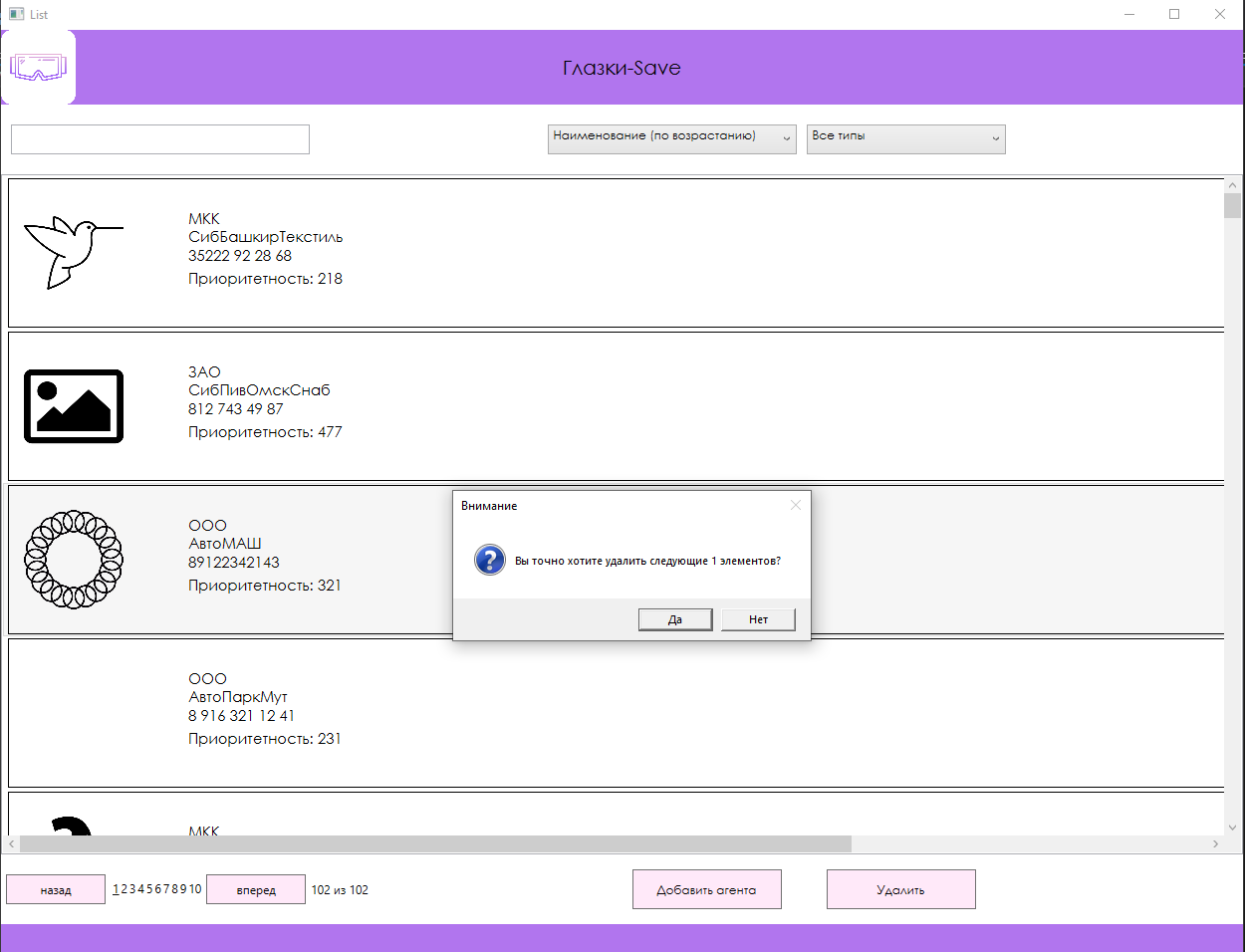
Работающий поиск

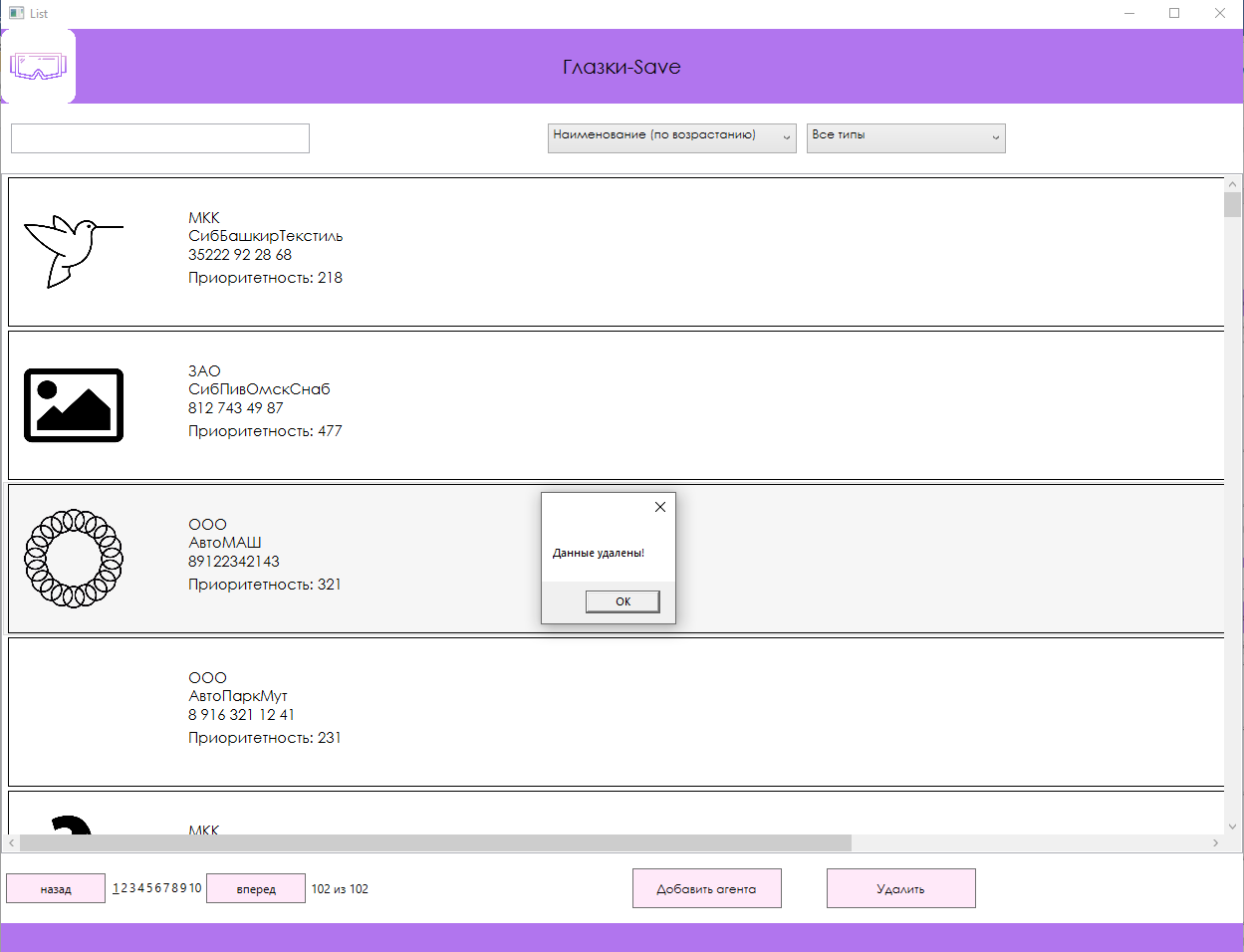




Сортировка и Фильтрация







Удаление

Разработка API

* + - 1. Для начала настраиваем возвращаемый формат json в файле WebAPIConfig (Рисунок 7)

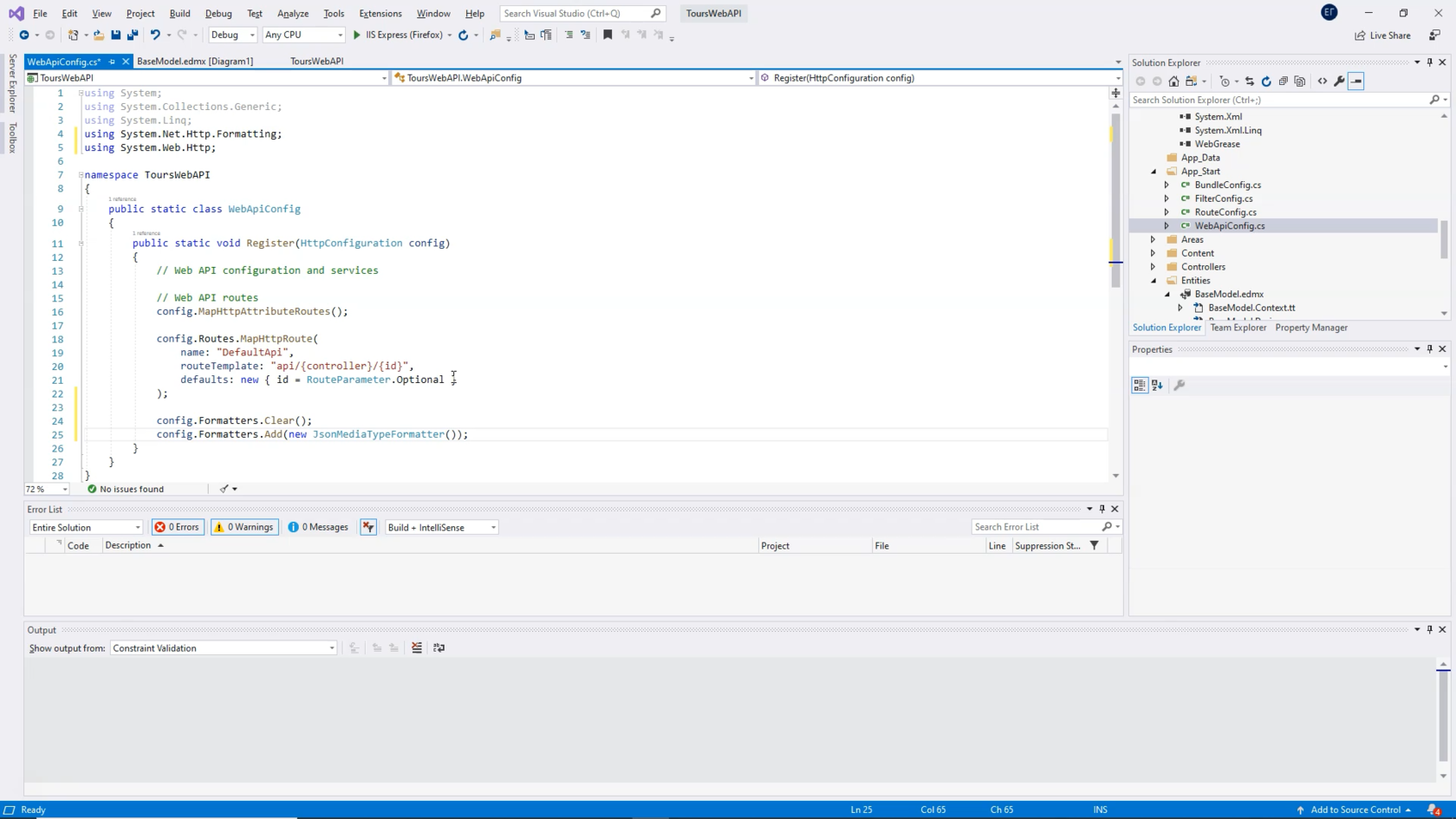


Рисунок 7. WebAPIConfig

* + - 1. Создаем контроллеры для класса Agent (Рисунок 7.1)

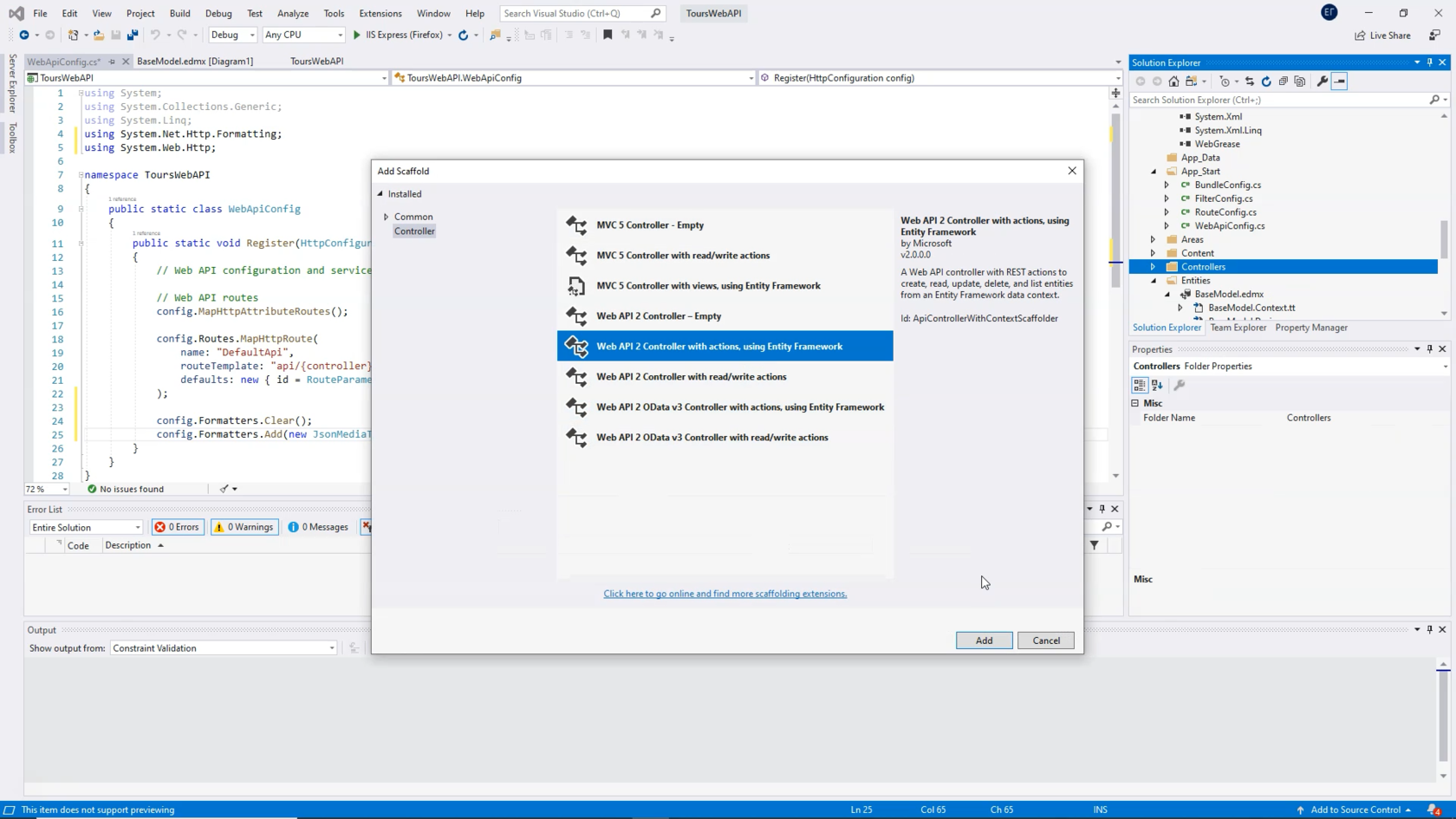


Рисунок 7.1. Создание контроллеров

* + - 1. Запускаем и тестируем это в браузере. У нас есть раздел API, который описывает документацию наших запросов (Рисунок 7.2)

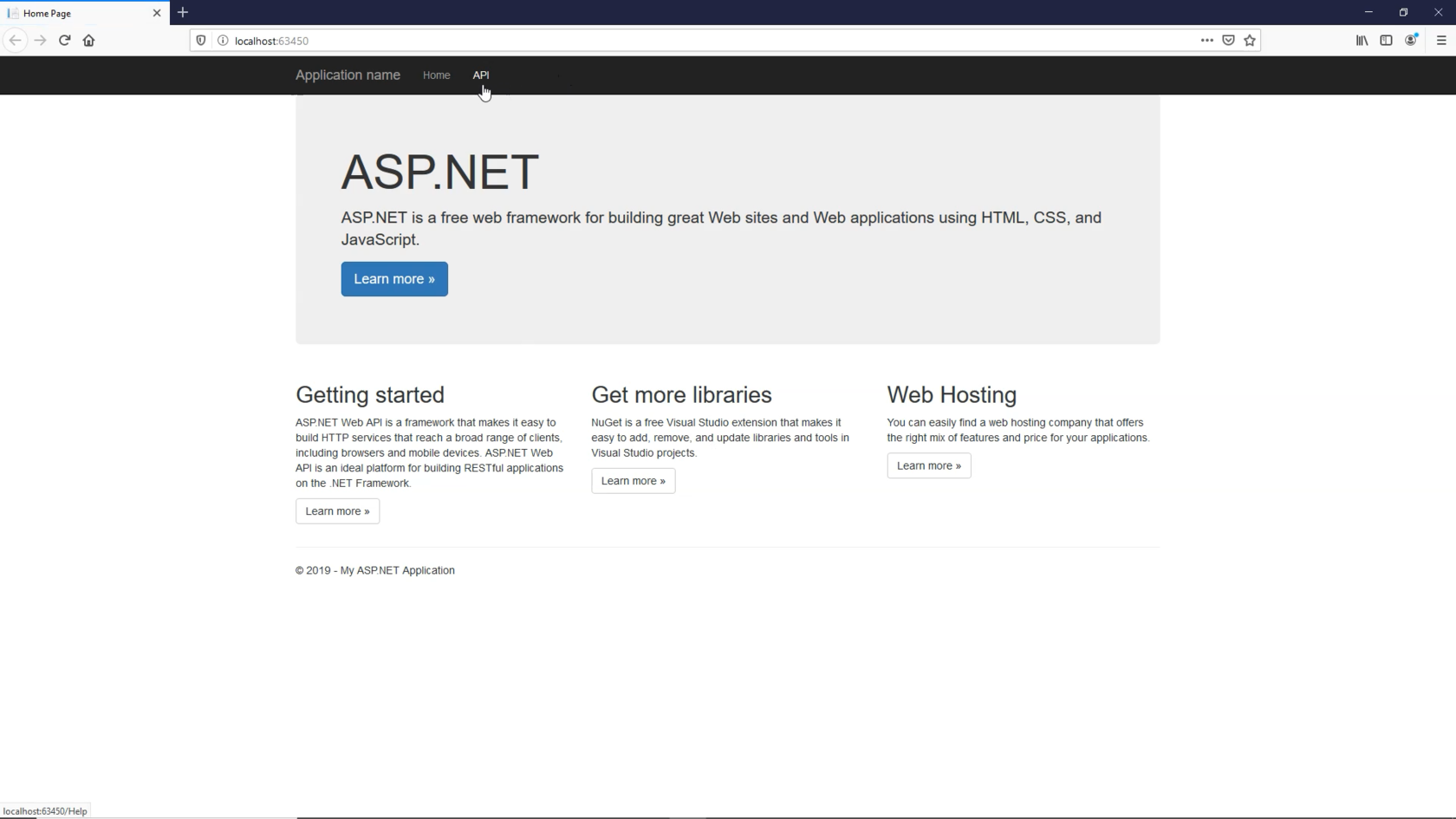


Рисунок 7.2. Раздел API

* + - 1. Здесь у нас есть созданные контроллеры, в том числе Agents. На методе Agents типа get у нас есть Response Information, который мы, как раз, указали в качестве атрибута, и список его свойств (Рисунок 7.3)

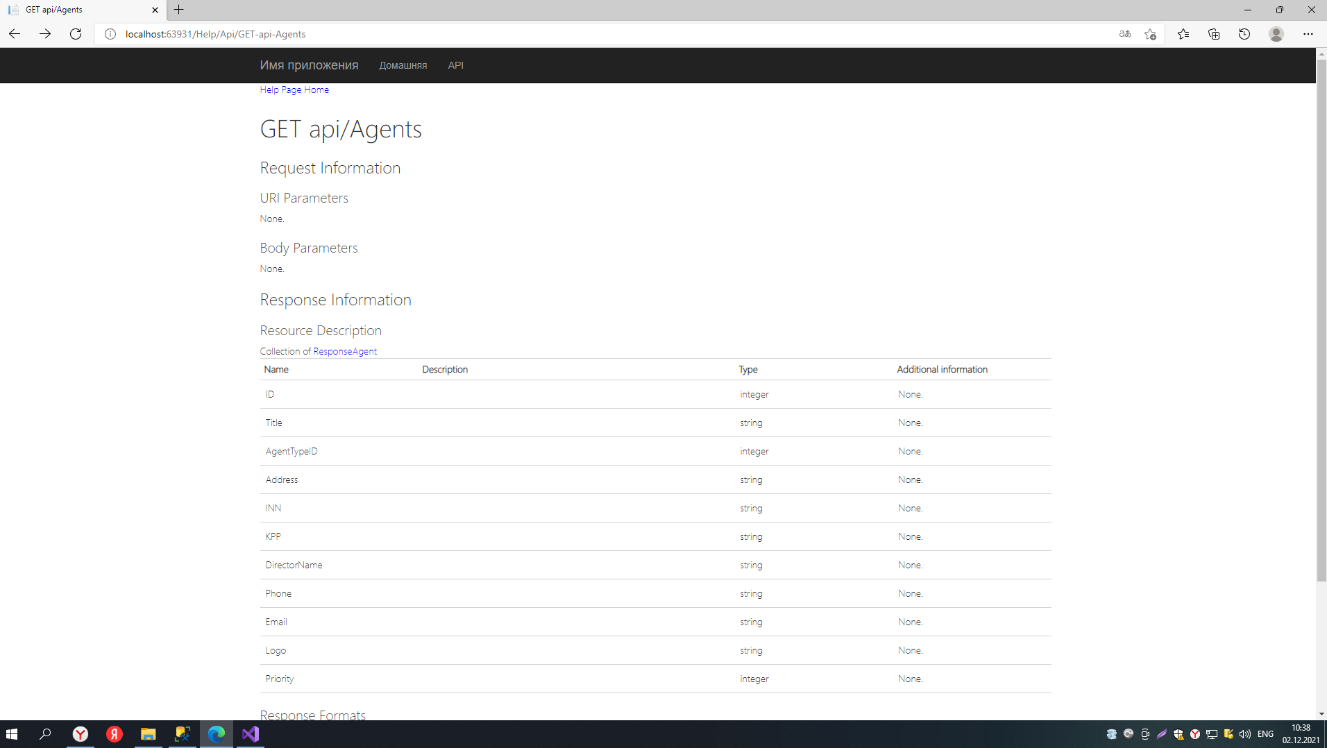


Рисунок 7.3. Метод Agent типа get

1. Попробуем вызвать запрос на получение списка отелей. (Рисунок 7.4)

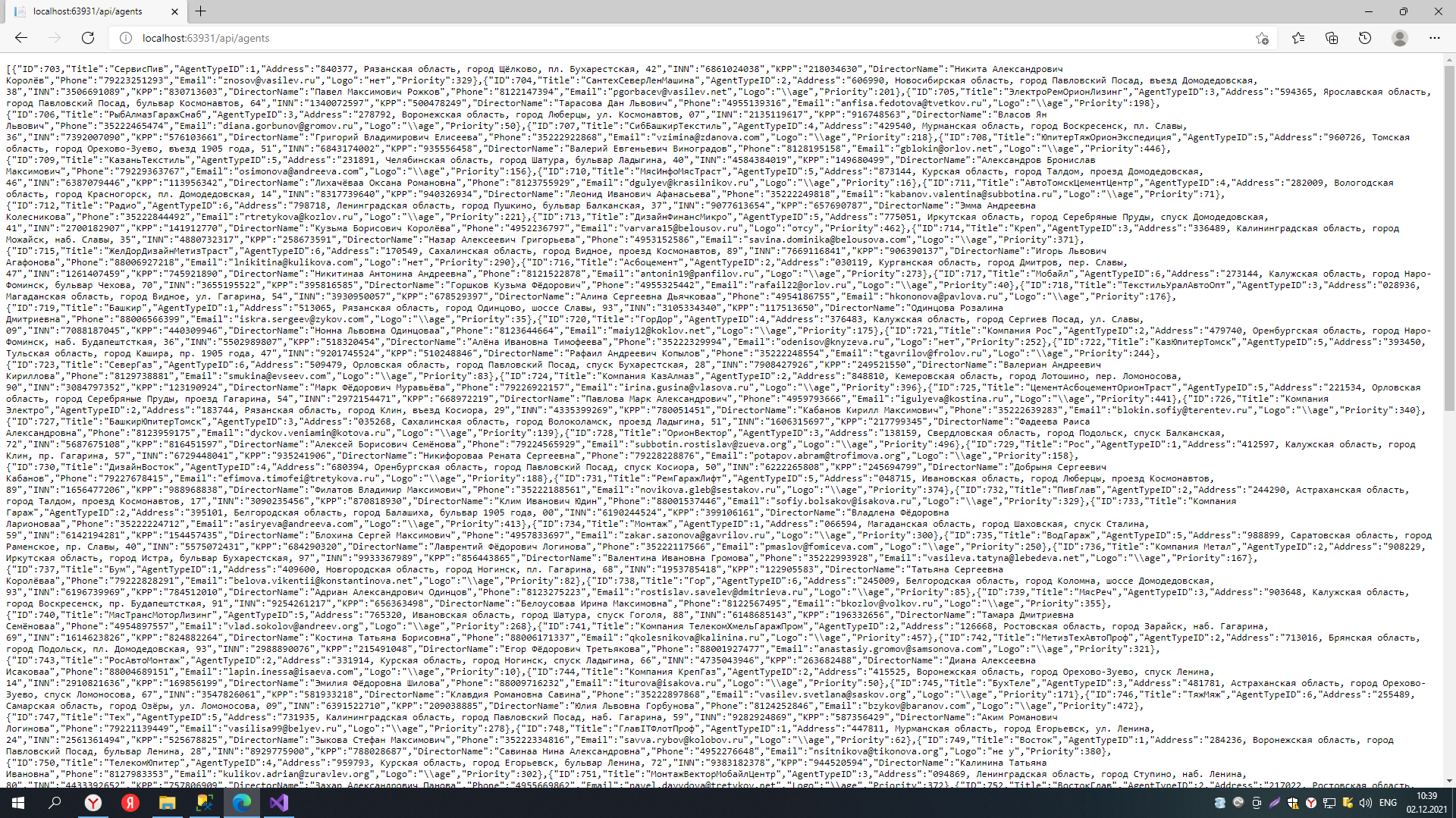


Рисунок 7.4. Запрос на получение

1. Открываем Postman, вводим в адресную строку путь, вводим значения, которые хотим добавить в нашу БД (Рисунок 7.5)

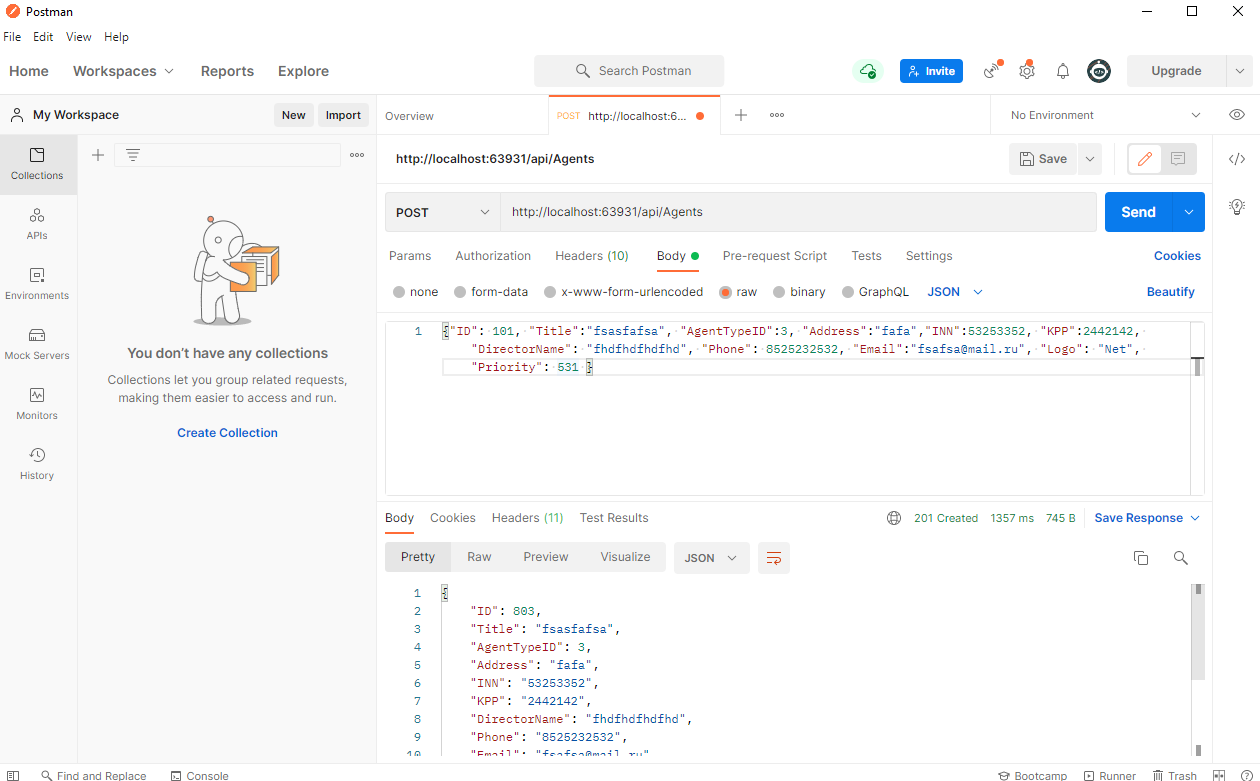


Рисунок 7.5. Postman

1. Можно увидеть новую запись в таблице Agents (Рисунок 7.6.)

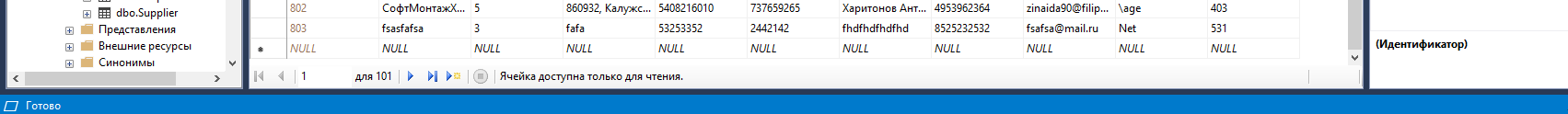


Рисунок 7.6. Добавление в базу данных новой информации

# Выводы

По окончании учебной практики были решены следующие задачи:

1. Описана предметная область
2. Изучены программные средства
3. Восстановлена базы данных из скрипта
4. Импортированы данные
5. Разработка desktop-приложения
6. Реализовано чтение, добавление, редактирование, удаление данных
7. Реализована сортировка, фильтрация и поиск данных
8. Разработан API
9. Разработано мобильное приложение
10. Загружен файл и описание проекта на репозиторий системы контроля версий git.

В ходе этой практики были сформированы следующие профессиональные компетенции:

ПК 1.1 Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.

ПК 1.2 Разрабатывать программные модули в соответствии с техническим заданием.

ПК 1.3 Выполнять отладку программных модулей с использованием специализированных программных средств

ПК 1.4 Выполнять тестирование программных модулей

ПК 1.5 Осуществлять рефакторинг и оптимизацию программного кода

ПК 1.6 Разрабатывать модули программного обеспечения для мобильных платформ.

# Список литературы

1. [Орлов С.А.](http://library.sgu.ru/cgi-bin/irbis64r_15/cgiirbis_64.exe?LNG&amp;Z21ID&amp;I21DBN=NIKA&amp;P21DBN=NIKA&amp;S21STN=1&amp;S21REF=1&amp;S21FMT=fullwebr&amp;C21COM=S&amp;S21CNR=20&amp;S21P01=0&amp;S21P02=1&amp;S21P03=A%3D&amp;S21STR=%D0%9E%D1%80%D0%BB%D0%BE%D0%B2%2C%20%D0%A1%D0%B5%D1%80%D0%B3%D0%B5%D0%B9%20%D0%90%D0%BB%D0%B5%D0%BA%D1%81%D0%B0%D0%BD%D0%B4%D1%80%D0%BE%D0%B2%D0%B8%D1%87) Программная инженерия. Технологии разработки программного обеспечения: учебник / С. А. Орлов. – 5-е изд., обновл. и доп. – Москва; Санкт- Петербург: Питер, 2016. – 640 с.
2. Скит Джон. C# для профессионалов. Тонкости программирования / Скит Джон – Москва; Вильямс, 2019. – 608.
3. Павловская Т.А. С/С++. Программирование на языке высокого уровня. СПб.: Питер, 2017. – 461 с.
4. Семакин И. Г. Основы алгоритмизации и программирования: учебник для студ. учреждений сред. проф. образования / И. Г. Семакин, А. П. Шестаков. — М.: Издательский центр «Академия», 2017. — 304 с
5. Хабибулин И.Ш. Программирование на языке высокого уровня. C/C++. – СПб.: БХВ– Петербург, 2017. – 512 с.