Государственное бюджетное профессиональное образовательное учреждение

"Волжский политехнический техникум"

ДНЕВНИК

ПРОХОЖДЕНИЯ ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вид практики

Специальность \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Код и наименование специальности

Студента(ки)\_\_\_\_\_\_\_\_ курса, группы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

форма обучения\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(очная, заочная)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Фамилия, имя, отчество)

Место практики

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Название организации)

Срок практики с «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_202\_\_ г. по «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_202\_\_ г.

Волжский 2022

Форма дневника

|  |  |  |  |
| --- | --- | --- | --- |
| Дата | Описание выполненной работы | Оценка | Подпись руководителя практики |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Содержание объемов выполненных работ подтверждаю

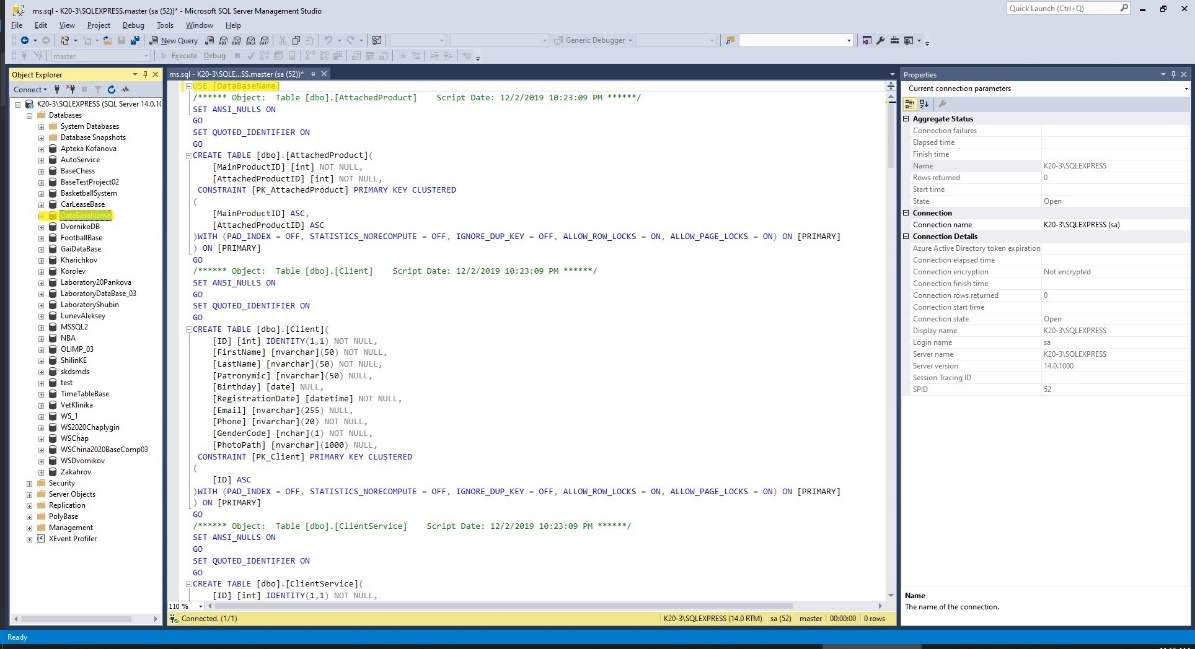
Руководитель практики от предприятия \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

подпись Ф.И.О.

МП

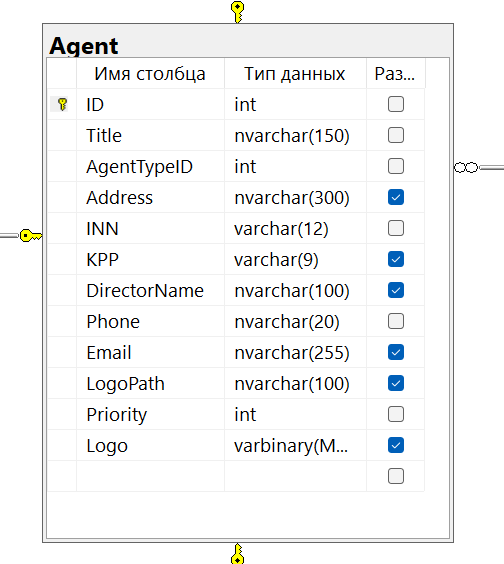
**Импорт и создание базы данных**

1. Для восстановления базы данных воспользуйтесь предоставленным скриптом (ms.sql). В процессе разработки приложения Вы можете изменять базу данных на свое усмотрение. Откройте MS SQL Server. Перетащите скрипт ms.sql в среду. Обратите внимание, что скрипт использует уже созданную базу данных, которую вам необходимо создать предварительно с тем же именем, что и указано в скрипте (EyesBase). Запустите скрипт в среде.

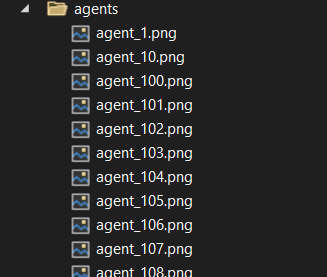


2. Заказчик системы предоставил файлы с данными (с пометкой import в ресурсах) для переноса в новую систему. Подготовьте данные файлов для импорта и загрузите в разработанную базу данных.

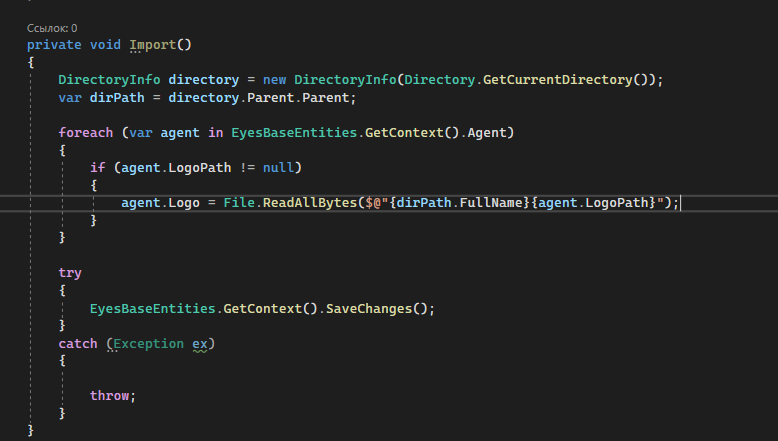
3. Чтобы добавить картинку в базу данных,ПОСЛЕ заполнения всех таблиц, создайте новое поле в таблице Agent–Logo(старое поле Logo переименуйте в LogoPath). Укажите тип данных для поля Logovarbinary(MAX), укажите что эти два поля могут иметь значение NULL.



После того, как вы подключите базу данных и реализуете Singleton подключение, создайте папку agentsв проекте, закиньте туда фотографии данные вам для импорта.

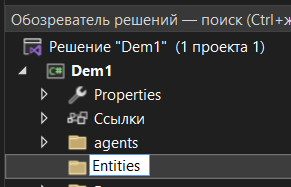


Затем в коде проекта создайте метод Import(), который будет заполнять поле Logo, ссылаясь на путь к картинке из LogoPath. Получите путь к корневой папке проекта с помощью классов DirectoryInfo и Directory.Затем поочерёдно проверяйте каждого агента на наличие пути к картинке, если оно не равно NULL, с помощью метода File.ReadAllBytes() записывайте картинку в свойство Logo. Сохраните именения. После использования этого метода, его следует удалить, больше он не понадобиться. Также советую удалить из таблицы Agent поле LogoPath.

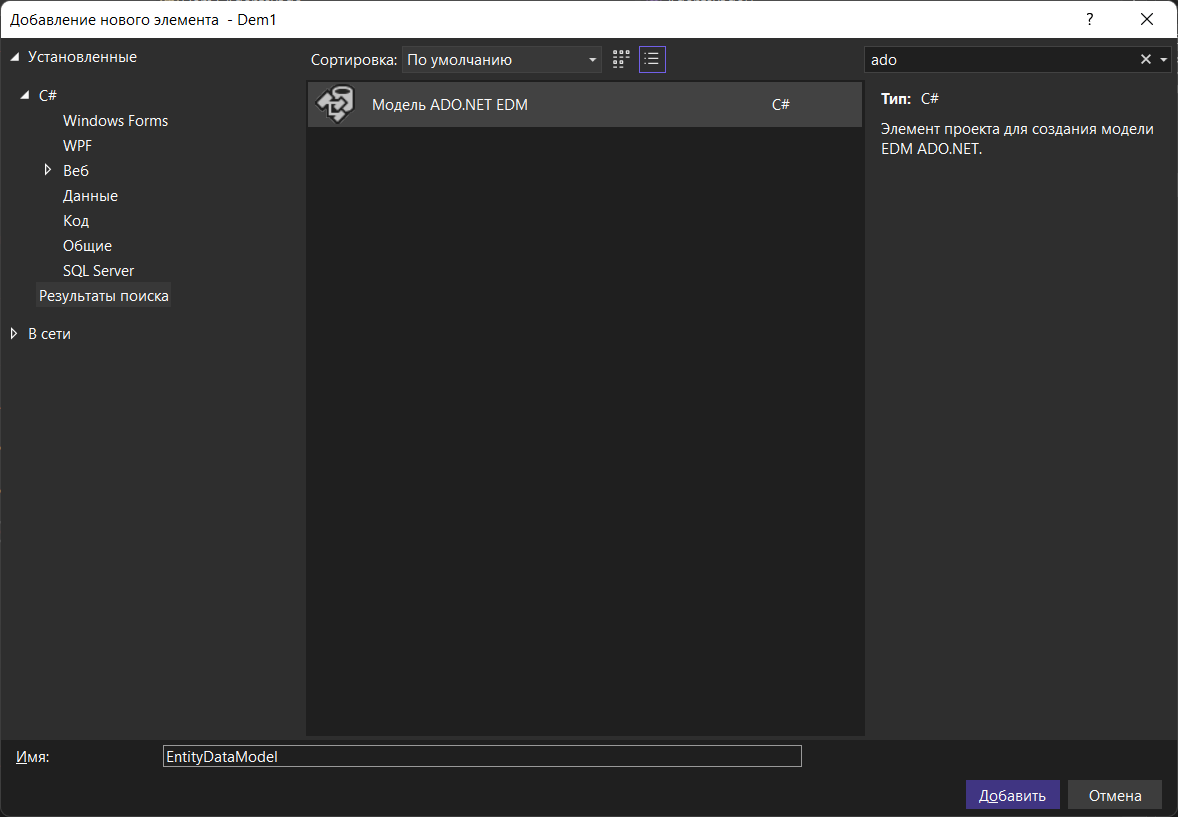


**Подключение к базе данных**

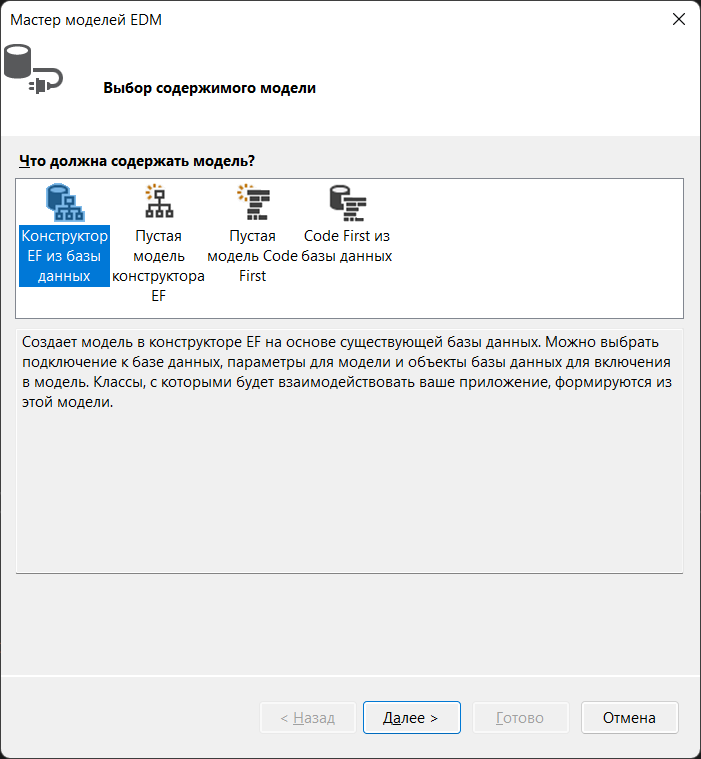
1. Создайте новую папку в проекте с названием Entities.



2. Добавьте в эту папку модель ADO.NET, которая преобразует вашу базу данных из таблиц в модель с набором связанных классов. (ПКМ по папке – Add – NewItem)

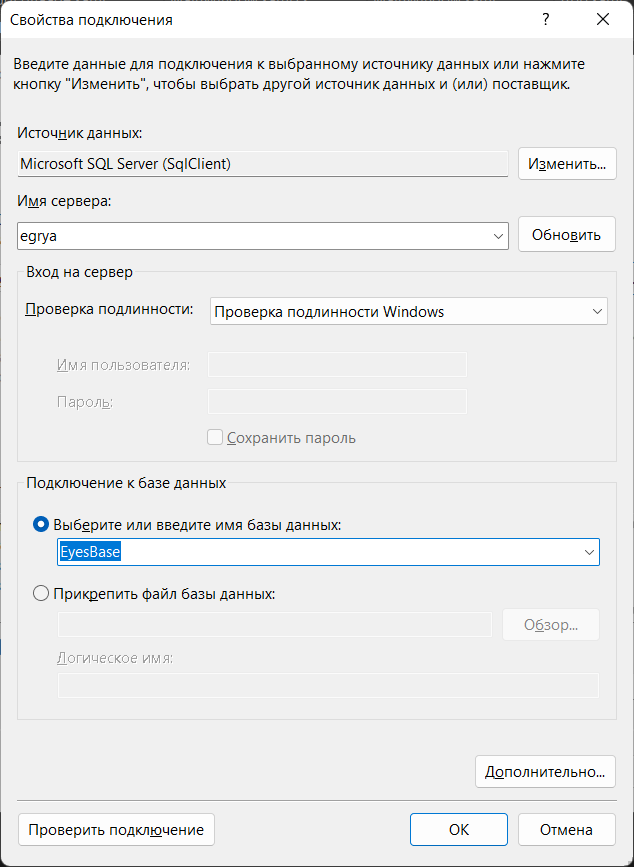


3. Выберите способ генерации модели – EFDesignerfromdatabase.

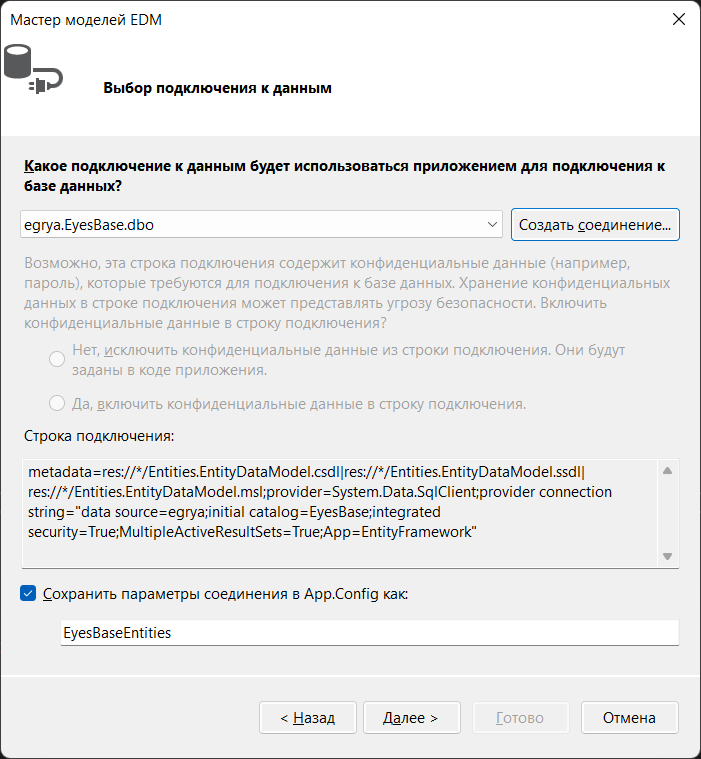


4. В появившемся окне выберите NewConnection для создания нового подключения к базе данных.

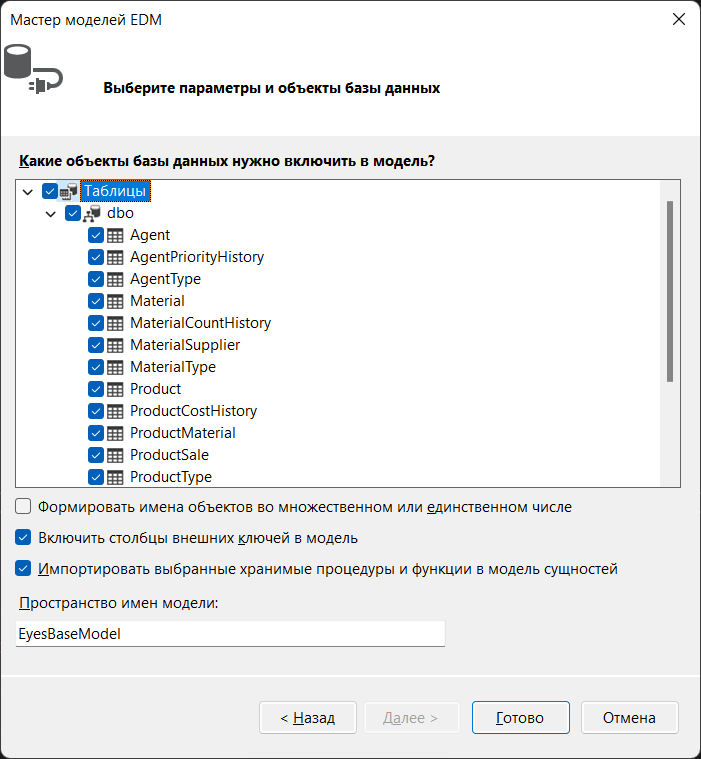
Способ авторизации аналогичен окну в MSSQLServer. После ввода данных выберите необходимую БД из выпадающего списка в поле SelectorEnteradatabasename.



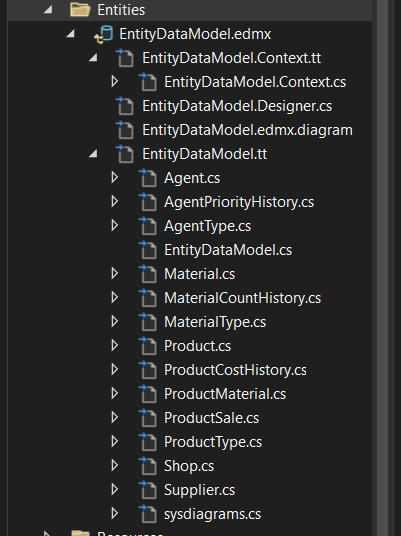
5. Нажмите Next.



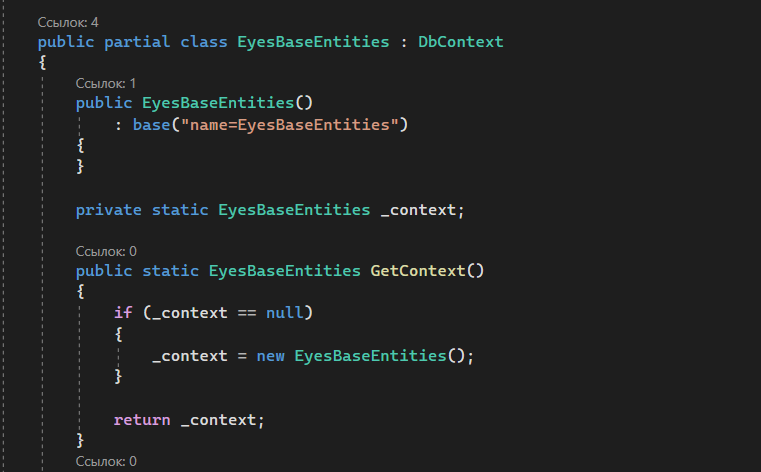
6. Выберите все таблицы из базы данных и убедитесь, что включены указанные ниже маркеры.



7. В папке Entitiesпоявится новая модель базы данных.

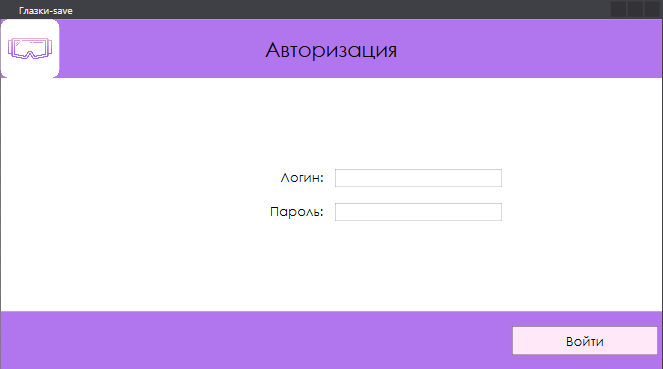


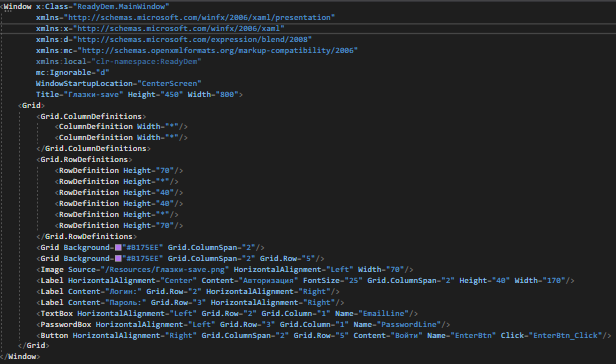
8. Добавьте в класс EntityDataModel.Context.csприватное статическоесвойство \_сontext и публичный статический метод GetContext(), возвращающий экземпляр модели EntityFramework(реализуйте паттерн Singleton).



**Авторизация**

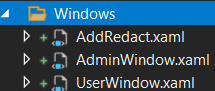
1. Создайте проект, пропишите стили, сверстайте первоначальный макет окна MainWindow.



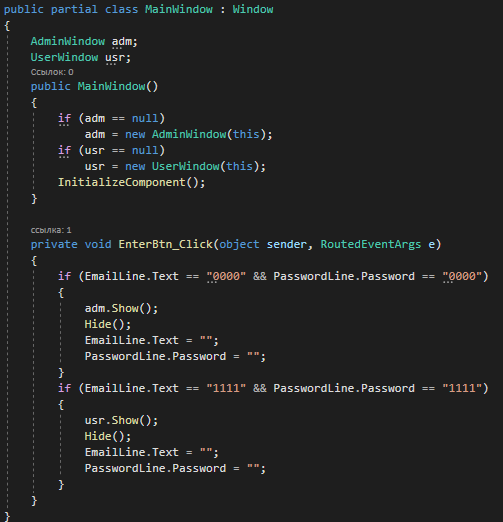


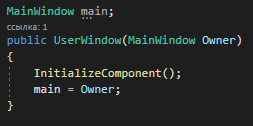
1. Создайте еще несколько окон.

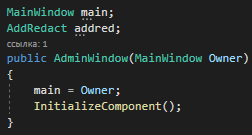
Добавьте еще 3 окна: 1)Окно пользователя; 2)Окно Админа; 3)Окно для добавления и редактирования.



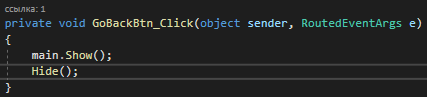
1. Написать код, для входа в окно пользователя и админа.





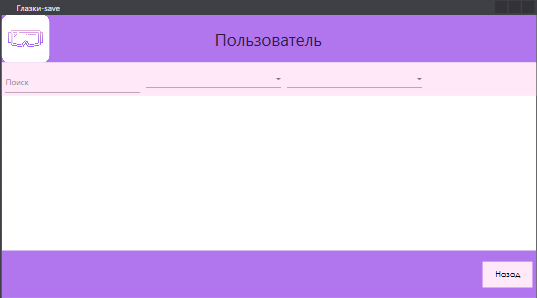


1. Обработайте нажатие кнопки назад:



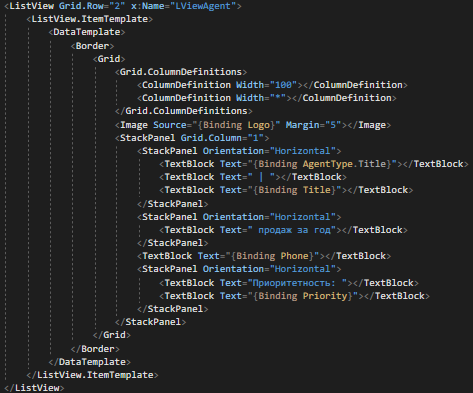
**Список агентов**

1. Создать макет для окна пользователя, в котором будут: фильтры, поиск, результирующую информацию, сам список агентов).

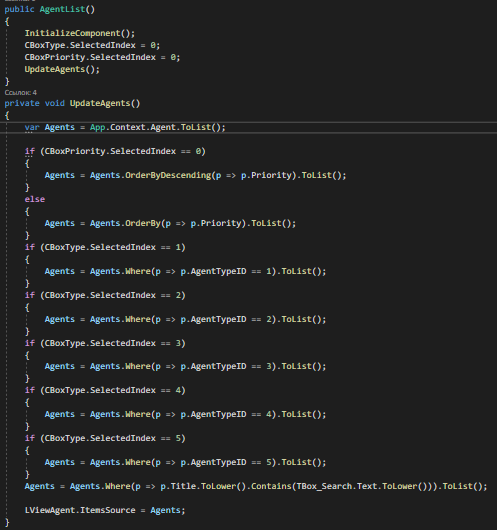


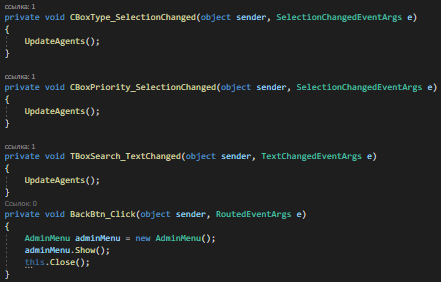


2. Настройте отображение элементов внутри ListView

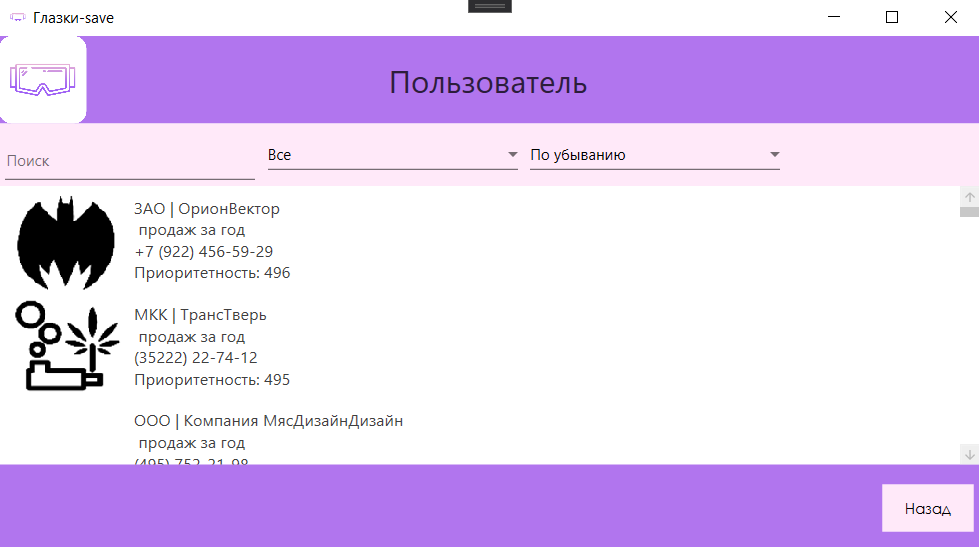


3. Создайте единый метод для загрузки списка агентов с учтением параметров сортировки, поиска и фильтрации(заранее создайте для поиска событие TextChanged, а для фильтрации и сортировки SelectionChanged, в них будите вызывать метод UpdateAgents())



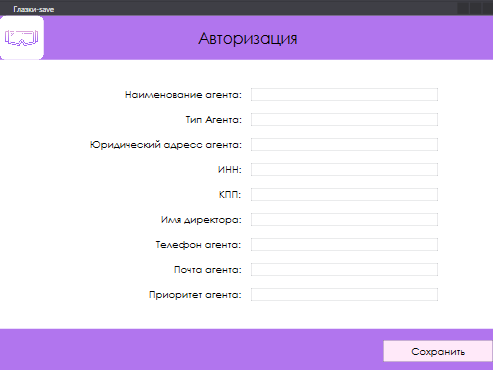


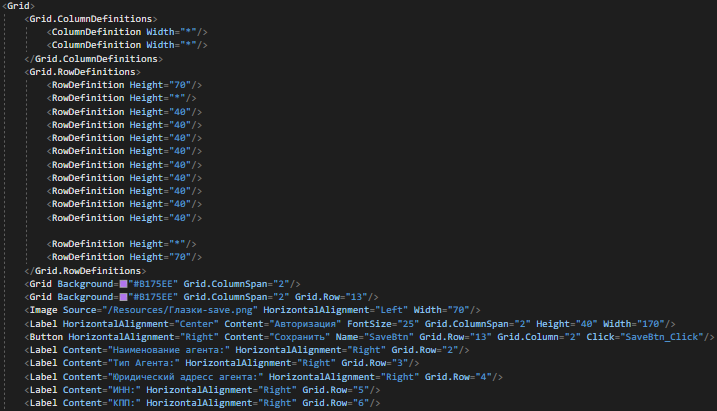
4. Готово

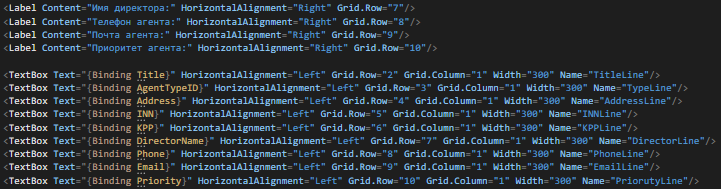


**Добавление, удаление, редактирование**

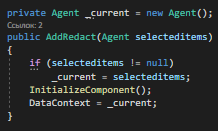
1. Сверстайте макет на странице AddRedact – совершите переход на эту страницу по клику на кнопку Добавить из окна AdminWindow. Обработайте новыми событиями клики по кнопкам.



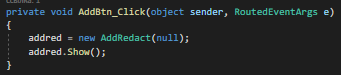




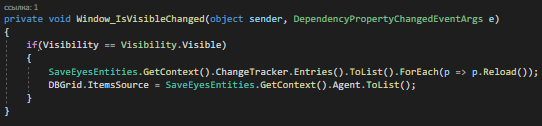
2. В коде страницы добавьте в принимаемые значения конструктора сущность агента, если принимаемое значение не будет равно null присвойте его значение к свойству \_current и приведите его к DataContext страницы. Также заполните комбо-бокс с типами агента данными.



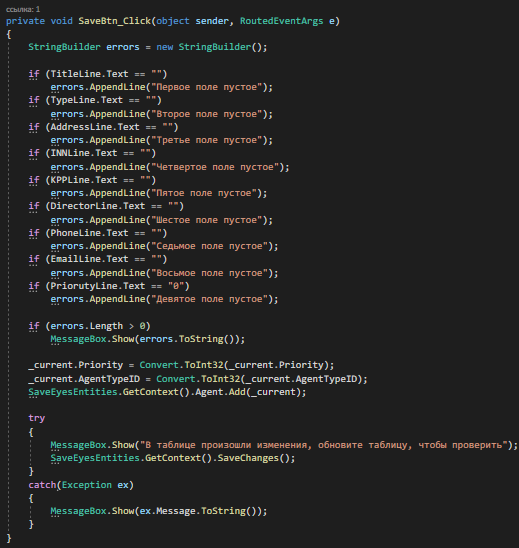
3. По клику на кнопку добавления добавьте следующий код:



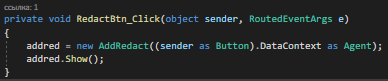
4. В окне AdminWindow обработайте событие IsVisibleChanged, которое будет срабатывать каждый раз, когда страница будет загружена. Это нужно для того, чтобы услуги динамически обновлялись про добавлении/редактировании с других страниц.



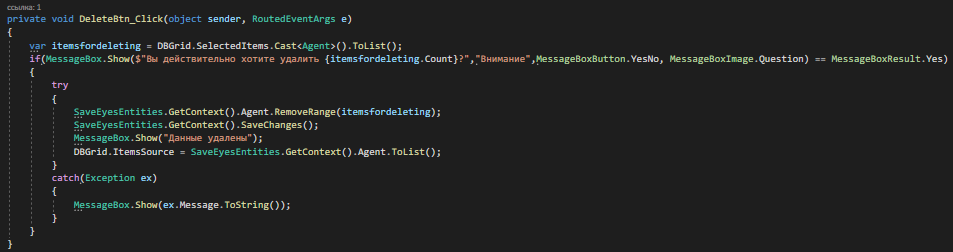
5. Добавьте код для кнопки сохранения.



6. В окне AdminWindow пропишите логику клика на кнопку Редактировать.

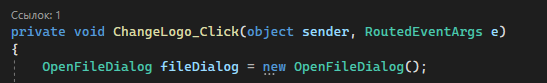


7. Для удаления записи пропишите логику при клике на кнопку Удалить в AdminWindow. Мы получаем список выделенных элементов, затем запрашиваем подтверждения пользователя. Если он нажмет кнопку «Да», то услуга удаляется, список обновляется.



**Добавление изображения**

1. Обработайте клик на кнопку выбора фотографии. В логике нужно создать переменную типа OpenFileDialog (тип будет подчеркиваться красным, просто наведите на него, нажмите на лампочку, выберите using Microsoft.Win32, это подключит необходимое пространство имён для работы с классом).



2. Установите фильтры для поиска и допишите код до такого состояния:

