|  |
| --- |
| **МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ** |
| **федеральное государственное бюджетное образовательное учреждение высшего образования «Российский экономический университет имени Г.В. Плеханова»**  **Московский приборостроительный техникум** |

**ОТЧЕТ**

по учебной практике

УП 01.01 «Прикладное программирование»

*индекс по УП и наименование практики*

Профессионального модуля ПМ.01 «Разработка модулей программного обеспечения для компьютерных систем»

*индекс по УП и наименование профессионального модуля*

Специальность 09.02.07 «Информационные системы и программирование»

*код и наименование специальности*

Студент       Стукалов Илья Олегович

*(фамилия, имя, отчество)*

Группа П50-6-21

Руководитель по практической подготовке от техникума

София Алексеевна Скорогудаева

*(фамилия, имя, отчество)*

«29» марта 2023 года

СОДЕРЖАНИЕ

[Лабораторная работа №1 «Подключение базы данных. Чтение данных из БД» 2](#_30j0zll)

[Лабораторная работа №2](#_1fob9te) «Добавление данных в БД» 10

Лабораторная работа №3 «Удаление данных в БД» 16

Лабораторная работа №4 «Изменение данных в БД» 21

# Лабораторная работа №1 «Подключение базы данных. Чтение данных из БД»

Цель работы: научиться подключать базу данных к WPF проекту, а также читать данные из базы данных.

1. Оформление интерфейса

Для начала оформим дизайн нашего приложения. Сделаем адаптивную верстку, а также добавим две кнопки для перемещения между таблицами и сам вид таблицы.

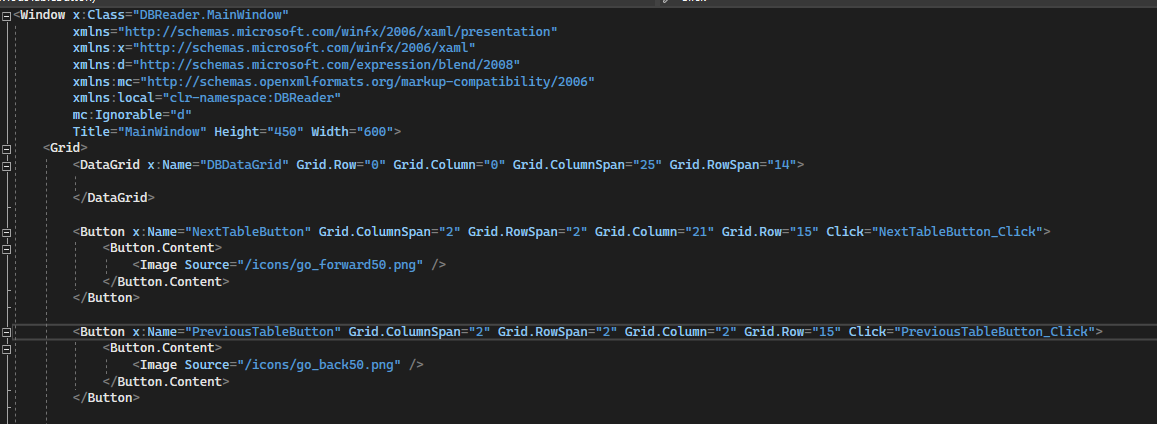


Рис. 1. Добавление кнопок и вида таблиц

Затем сделаем грид, необходимый для адаптивной верстки.

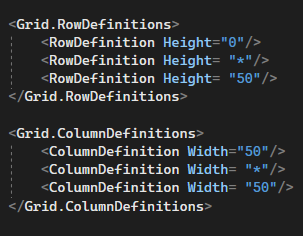


Рис. 2. Грид для адаптивной верстки

1. Создание базы данных, таблиц и SQL запросов

Сначала создадим таблицу, описывающую должности сотрудников. SQL запрос выглядит так:

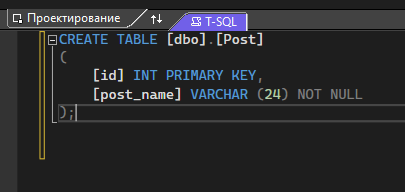


Рис. 3. SQL запрос для создания таблицы с должностями

Здесь id является уникальным ключом. Далее создадим SQL запрос для создания таблицы сотрудников. Запрос будет выглядеть так:

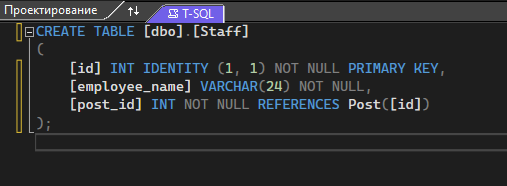


Рис. 4. Создание таблицы с сотрудниками

Здесь id является уникальным ключом для каждого сотрудника, а post\_id является ссылкой на запись в таблице с должностями.

В итоге получились две таблицы:

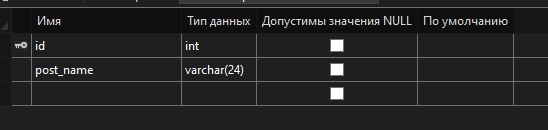


Рис. 5. Таблица должностей

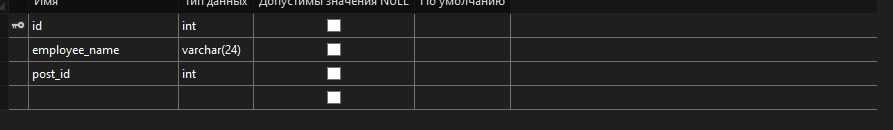


Рис. 6. Таблица сотрудников

1. Написание логики приложения

Далее напишем логику приложения. Для начала объявим и инициализируем необходимые переменные.

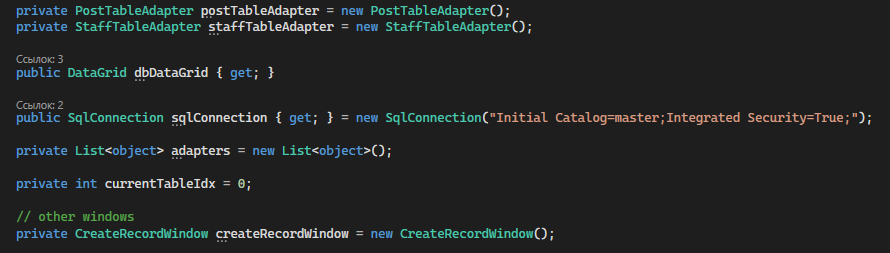


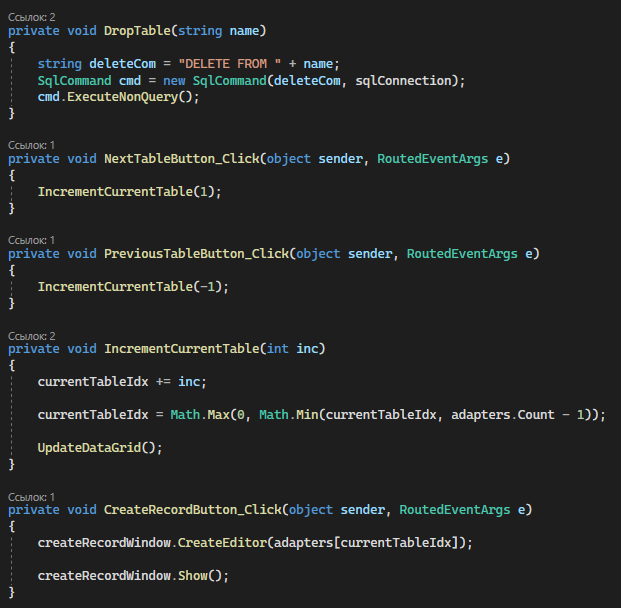
Рис. 7. Объявление и инициализация переменных

Далее в главном методе, который вызывается при создании окна напишем инициализацию некоторых переменных, а также удалим записи из всех таблиц в базе данных и добавим новые записи



Рис. 8. Инициализация и работа с БД

Далее напишем метод для удаления записей в таблице по имени таблицы. Также напишем обработку событий для левой кнопки на интерфейсе и правой.

Рис. 9. Удаление записи таблицы, обработка нажатий кнопок

1. Результаты:

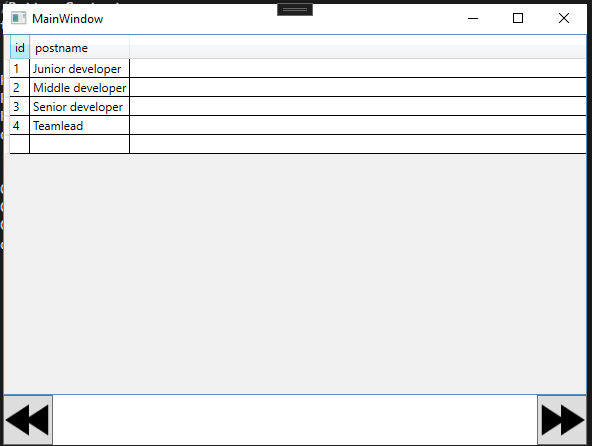


Рис. 10. Результат разработки

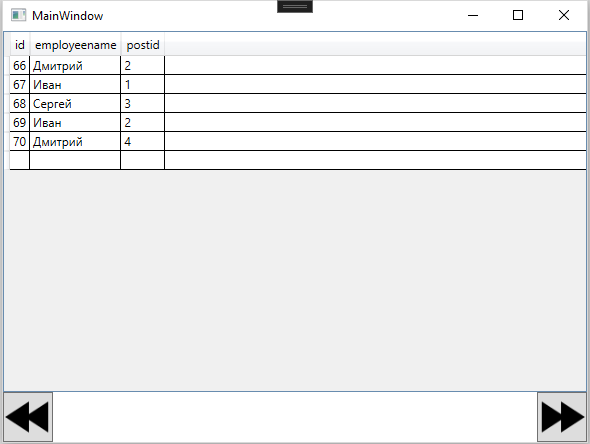


Рис. 11. Результат разработки

Вывод: научились подключать БД к WPF проекту. Улучшили знания SQL

Лабораторная работа №2 «Добавление данных в БД»

Цель работы: научиться добавлять данные в БД. Создать приложение для добавления записей в БД.

* + - 1. Оформление интерфейса

Для начала добавим все кнопки, а также DataGrid для вывода БД на экран.

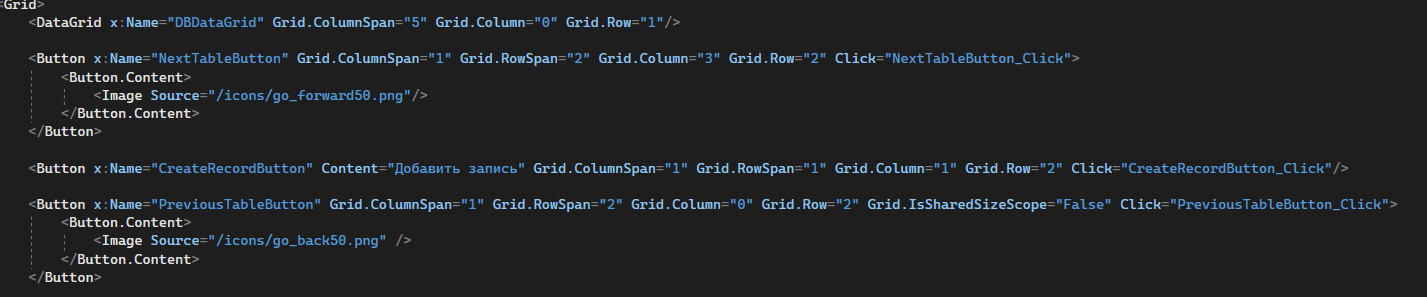


Рис. 11. Оформление интерфейса

Далее оформим грид, необходимый для адаптивной верстки.

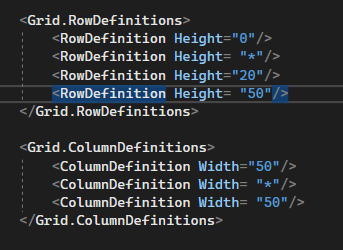


Рис. 12. Оформление грида

Также создадим второе окно для добавления новых записей и оформим его интерфейс. Создадим две кнопки «Закрыть» и «Добавить». Также добавим DataGrid для вывода информации о новой записи. Создадим TextBlock для вывода информации об ошибках. Оформим грид для адаптивной верстки.

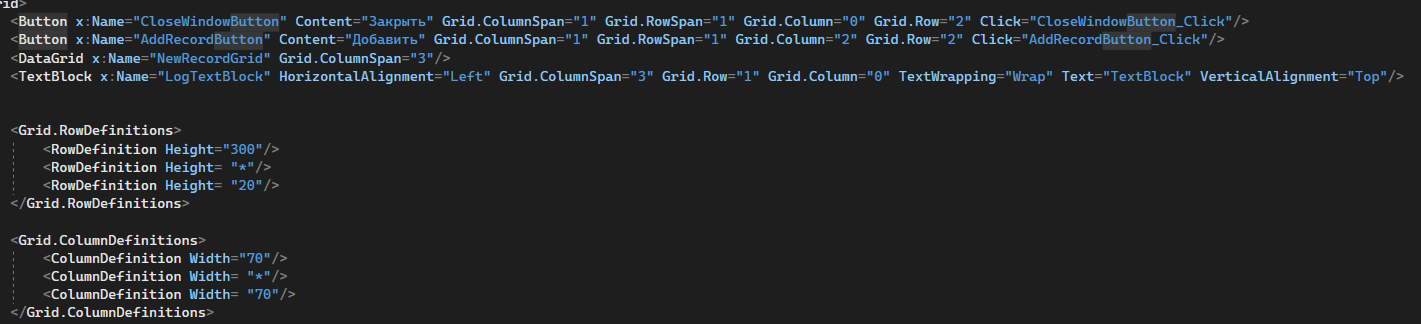


Рис. 13. Оформление второго окна

* + - 1. Написание кода основного окна

Код для основного окна почти не меняется.

Изменяются поля, добавляются новые. Добавим поле окна для создания записи и инициализируем его.

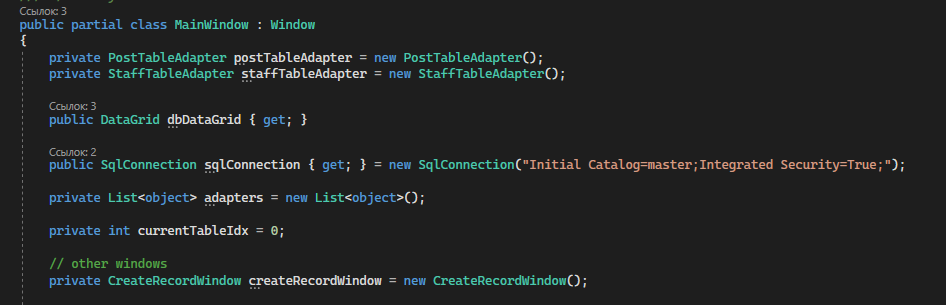


Рис. 14. Поля класса MainWindow

В конструкторе присвоим полю mainWindow у инстанса класса окна для создания записей данный инстанс.



Рис. 15. Присвоение полю класса CreateRecordWindow

Добавим метод для получения DataTable от адаптера таблицы.



Рис. 16. Метод для получения DataTable

Также добавим метод для обновления источника элементов у DataGrid, необходимого для отображения текущей таблицы в БД.

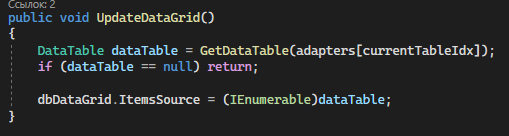


Рис. 17. Метод для обновления DataGrid

* + - 1. Написание кода окна, для создания записей

Объявим поля данного класса – DataGrid нужен для отображения информации о новой записи, TextBlock нужен для отображения ошибок выполнения запросов, DataTable является необходимым для редактирования новой записи, mainWindow ссылается на экземпляр главного окна. Object currentTableAdapter нужен для операций с текущей таблицей

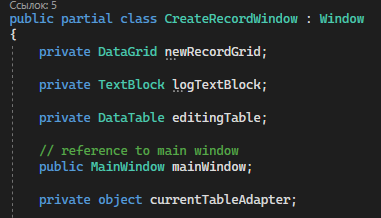


Рис. 18. Поля класса CreateRecordWindow

Далее напишем конструктор, в котором найдем необходимые элементы.

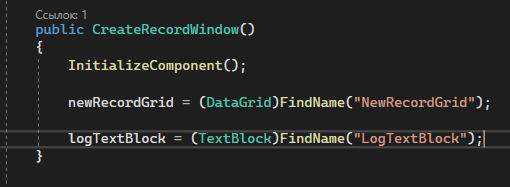


Рис. 19. Конструктор класса CreateRecordWindow

Далее напишем метод, срабатывающий при нажатии кнопки «Добавить». Он добавляет новую запись в текущую таблицу.

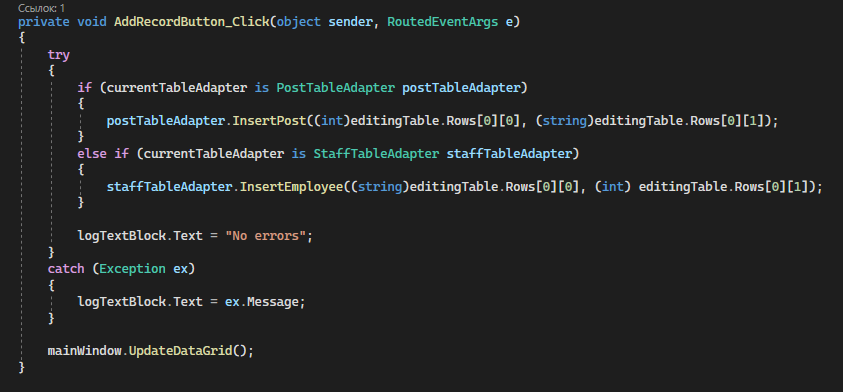


Рис. 20. Методы для формирования и выполнения SQL запроса

Напишем метод для создания редактора новой записи.



Рис. 21. Метод для создания редактора

В конце напишем два метода для скрытия окна, если был нажат крестик или кнопка «Закрыть».

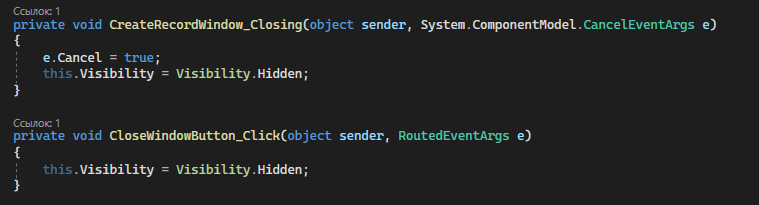


Рис. 22. Метод для скрытия окна

4. Результаты:

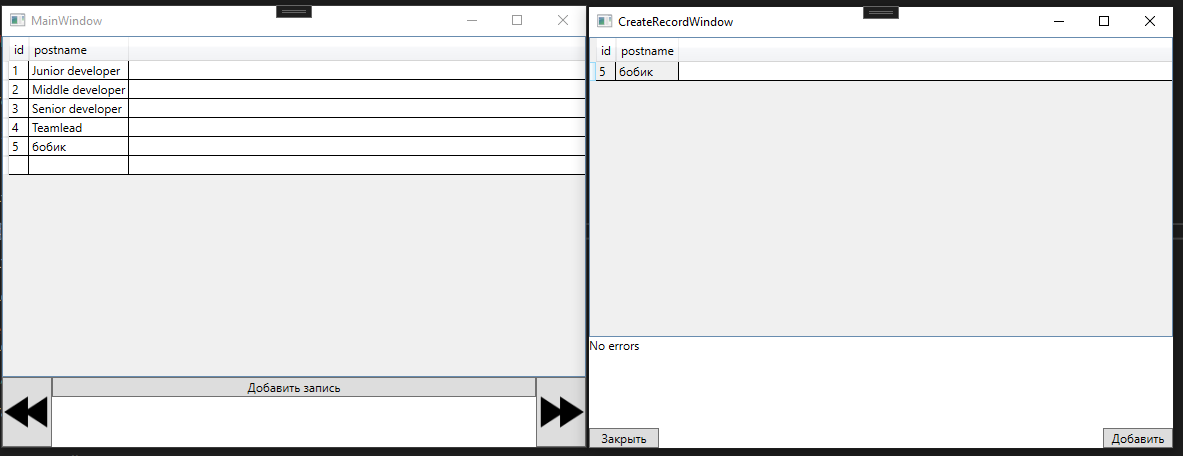


Рис. 23. Результаты

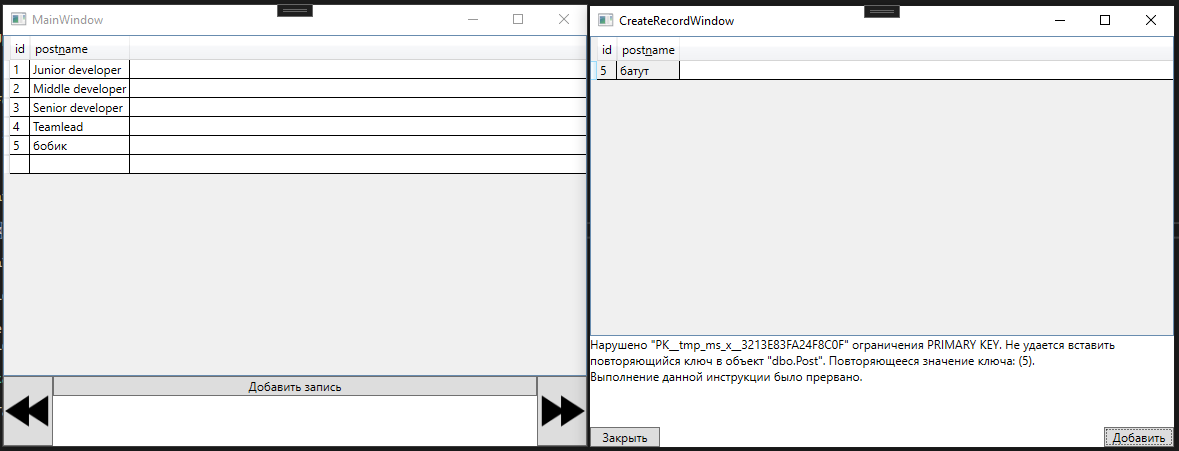
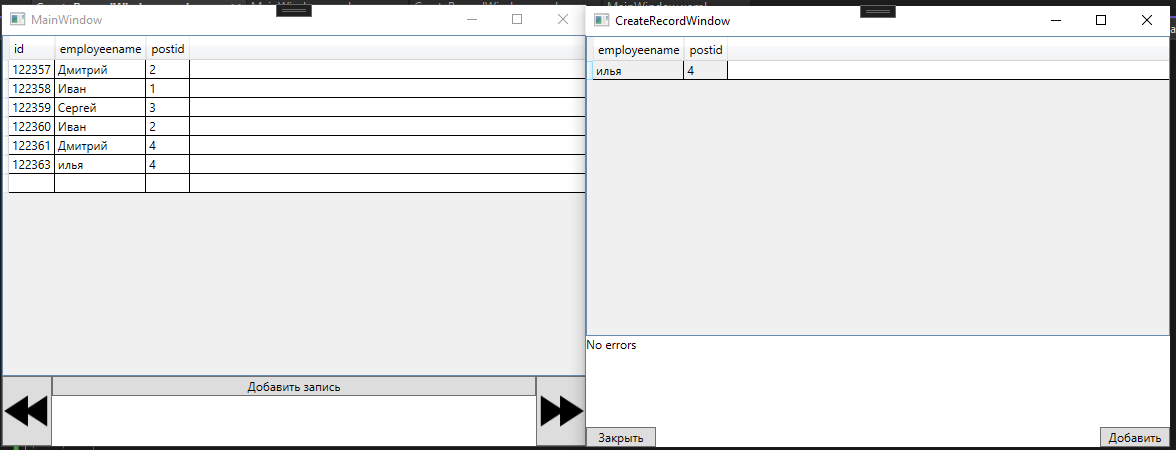


Рис. 24. Результаты

Рис. 25. Результаты

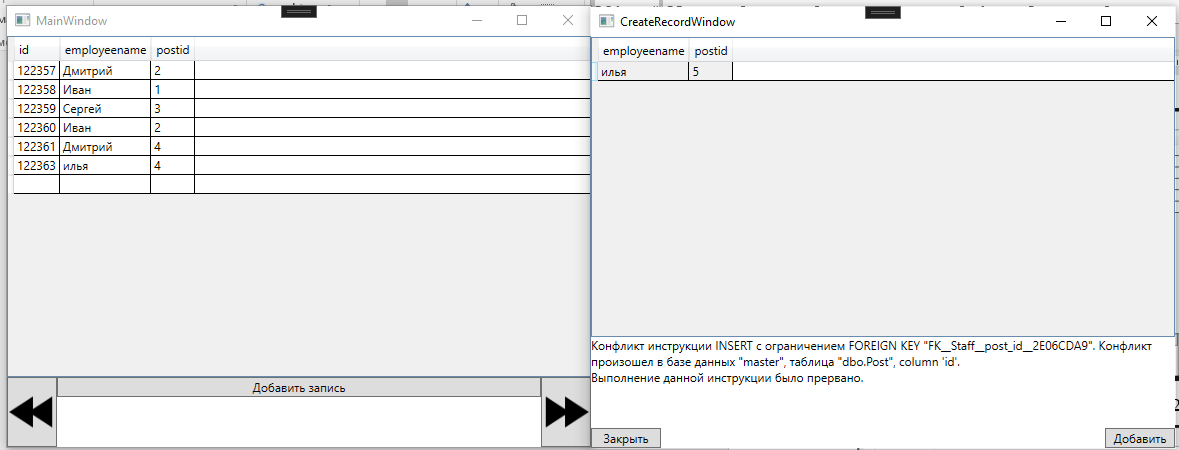


Рис. 26. Результаты

Вывод: научились добавлять данные в БД. Создали приложение для добавления записей в БД.

# Лабораторная работа №3 «Удаление данных в БД»

Цель работы: научиться удалять данные из БД с помощью Delete в классах адаптеров таблиц

Добавление новых запросов в таблицы

Сначала добавим новый запрос на удаление данных из таблицы с постами. Вместе с записью из таблицы с постами будут удаляться все записи из таблицы с сотрудниками, которые ссылаются на запись из таблицы.

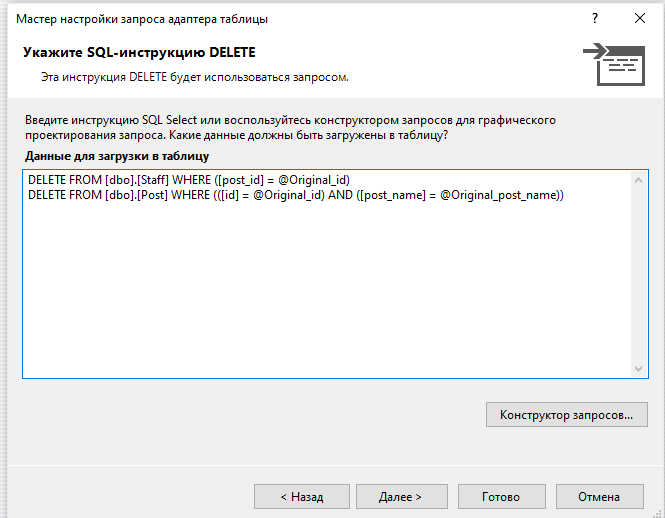


Рис. 27. Удаление записи из таблицы

Далее добавим новый запрос на удаление сотрудника из таблицы сотрудников.

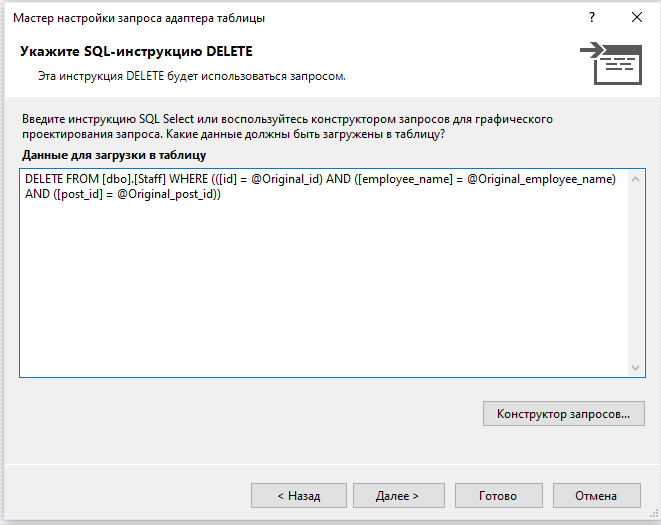


Рис. 28. Запрос на удаление записи

Добавление кнопки и обработчика

Для начала добавим новую кнопку для удаления записи, а также изменим положение кнопок для добавления записи и перелистывания таблиц.



Рис. 29. Настройка кнопок

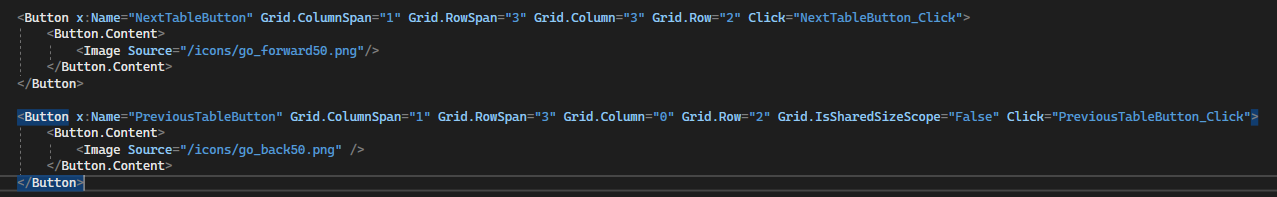


Рис. 30. Настройка кнопок

Изменим грид.

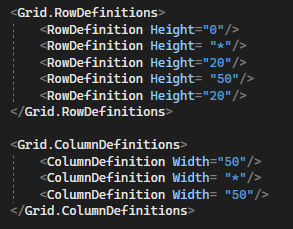


Рис. 31. Изменение грида

Добавим на кнопку обработчик и напишем логику удаления записи из таблицы. Вместе с записью будут удаляться все ссылки на неё в других таблицах.

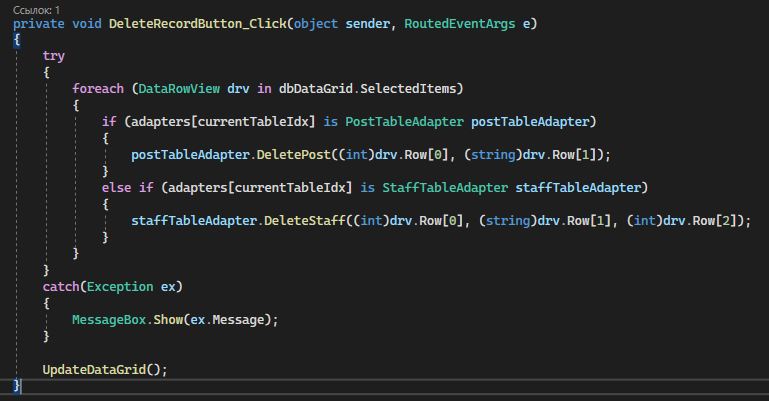


Рис. 32. Логика удаления записи в таблице

Результаты

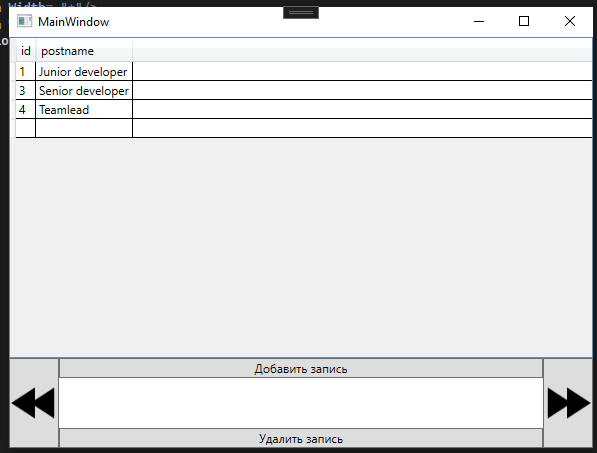


Рис. 33. Результаты

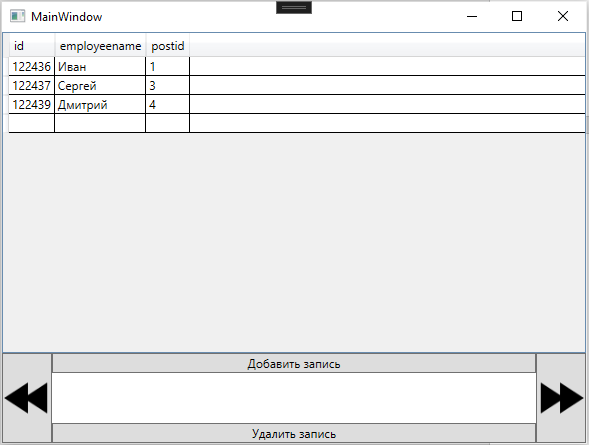


Рис. 34. Результаты

Вывод: научились удалять данные из БД с помощью Delete в классах адаптеров таблиц.

Лабораторная работа №4 «Изменение данных в БД»

Цель работы: научиться обновлять данные с помощью SQL запроса. Сделать обновление данных в приложении.

1. Добавление новых запросов в таблицы.

Для начала добавим новые запросы на обновление данных в таблицы.

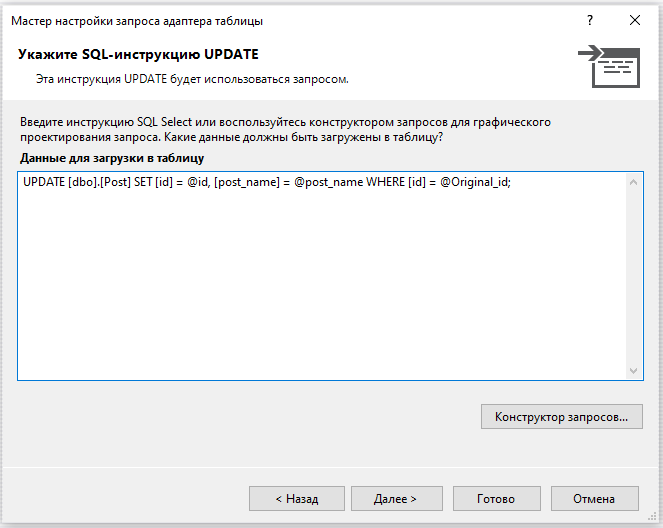


Рис. 35. SQL запрос на обновление должности по ID

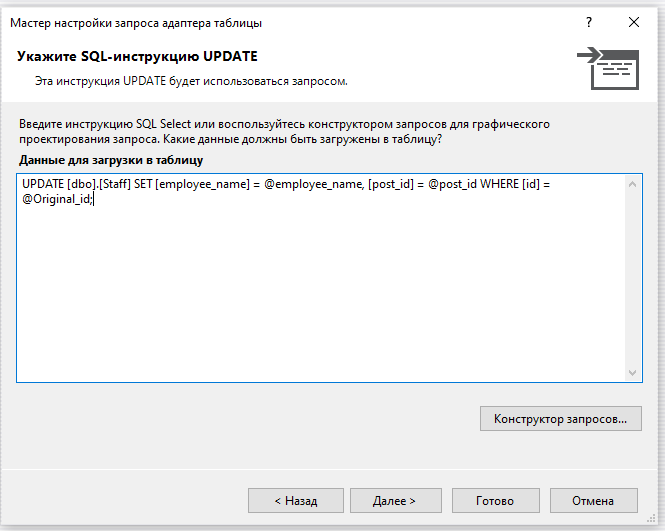


Рис. 36. SQL запрос на обновление сотрудника

1. Добавление обработчика

Добавим обработчик для DataGrid, который будет срабатывать по завершении редактирования строки. В данном методе обновим строку в таблице с помощью методов в адаптерах, которые мы сделали ранее.

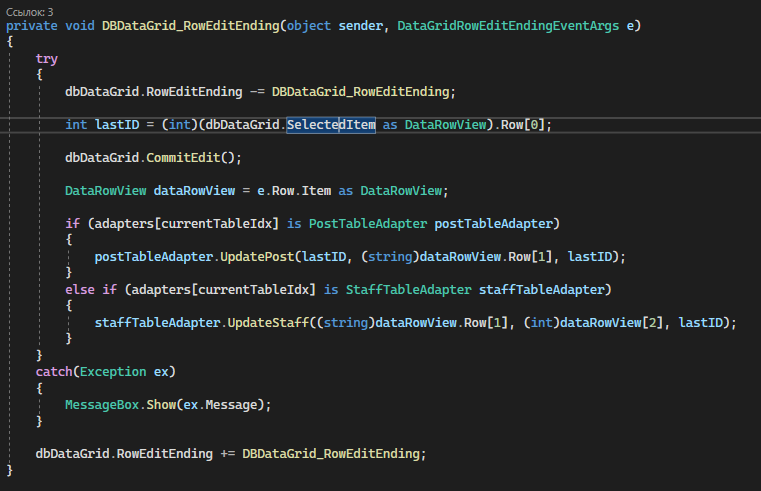


Рис. 37. Обработчик

1. Результаты:

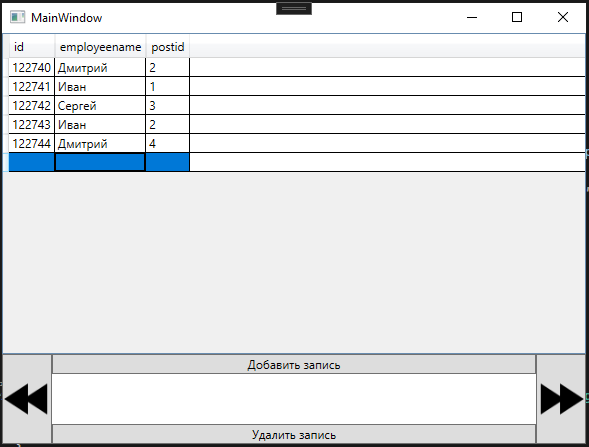


Рис. 38. До изменения строчки с ID 122744

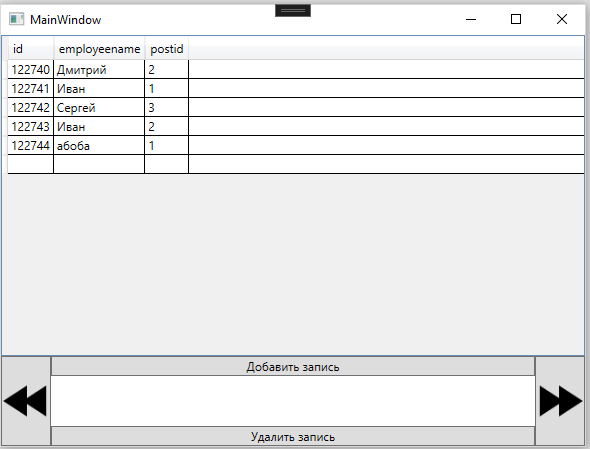


Рис. 39. После изменения строчки с ID 122744



Рис. 40. Обработанная ошибка

Вывод: научились обновлять данные с помощью SQL запроса. Сделали обновление данных в приложении.

Лабораторная работа №5 «Разработка информационной системы “Компании по производству мебели”»

Цель работы: разработать информационную систему для компании по производству мебели, а также базу данных для неё.

1. Создание необходимых таблиц

Для начала создадим нужные нам таблицы через SQL запросы в Visual Studio. Всего их будет 10.

Первая таблица FurnitureCarriers (Перевозчики мебели) имеет такой SQL запрос:

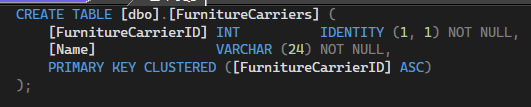


Рис. 41. SQL запрос для создания таблицы FurnitureCarriers

Она содержит столбцы с уникальным ключом записи в данной таблице, а также название перевозчика.

Вторая таблица MaterialsSuppliers (Поставщики материалов) имеет такой SQL запрос:

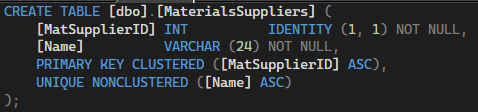


Рис. 42. SQL запрос для создания таблицы MaterialsSuppliers

Она содержит столбцы с уникальным ключом записи в данной таблице, а также уникальное название поставщика.

Третья таблица Factories (Заводы) имеет такой SQL запрос:

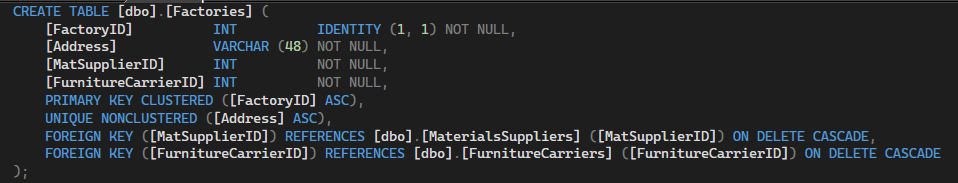


Рис. 43. SQL запрос для создания таблицы Factories

Она содержит столбцы с уникальным ключом записи в данной таблице, уникальный адрес завода, внешний ключ на поставщика материалов, а также внешний ключ на перевозчика мебели.

Четвертая таблица FurnitureTypes (Типы мебели) имеет такой SQL запрос:

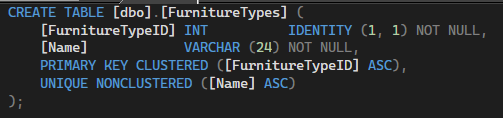


Рис. 44. SQL запрос для создания таблицы FurnitureTypes

Она содержит столбцы с уникальным ключом записи в данной таблице, а также уникальное название типа мебели.

Пятая таблица Furniture (Мебель) имеет такой SQL запрос:

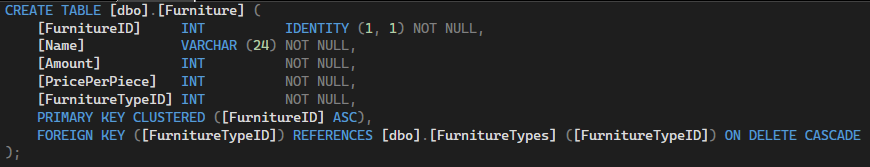


Рис. 45. SQL запрос для создания таблицы Furniture

Она содержит столбцы с уникальным ключом записи в данной таблице, название мебели, количество мебели, цена, а также внешний ключ на тип мебели.

Шестая таблица Materials (Материалы) имеет такой SQL запрос:

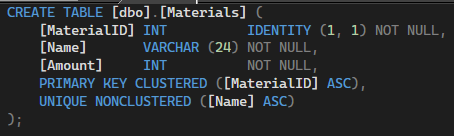


Рис. 46. SQL запрос для создания таблицы Materials

Она содержит столбцы с уникальным ключом записи в данной таблице, уникальное название материала, количество материалов.

Седьмая таблица Machines (Станки) имеет такой SQL запрос:

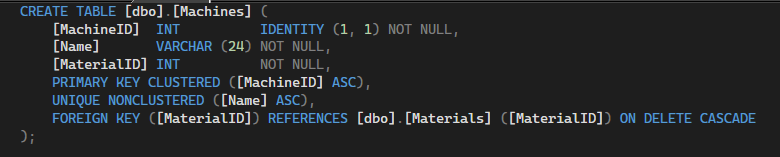


Рис. 47. SQL запрос для создания таблицы Machines

Она содержит столбцы с уникальным ключом записи в данной таблице, уникальное название станка, внешний ключ на материал, который используется для работы.

Восьмая таблица Post (Должности) имеет такой SQL запрос:

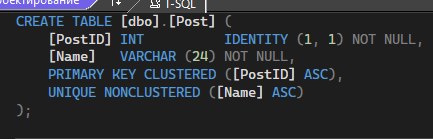


Рис. 48. SQL запрос для создания таблицы Post

Она содержит столбцы с уникальным ключом записи в данной таблице, уникальное название должности.

Девятая таблица Staff (Сотрудники) имеет такой SQL запрос:

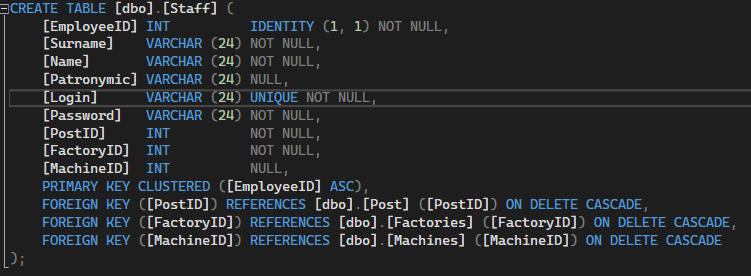


Рис. 49. SQL запрос для создания таблицы Staff

Она содержит столбцы с уникальным ключом записи в данной таблице, фамилию сотрудника, имя сотрудника, отчество сотрудника (если есть), уникальный логин, пароль, внешний ключ должности, внешний ключ завода, внешний ключ станка.

Десятая таблица Receipts (Чеки) имеет такой SQL запрос:

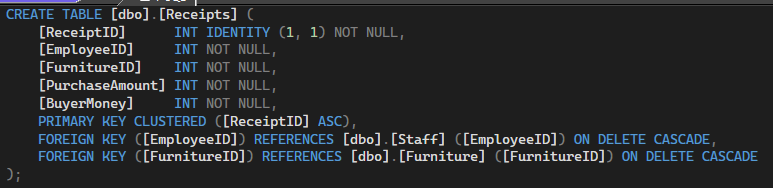


Рис. 50. SQL запрос для создания таблицы Receipts

Она содержит столбцы с уникальным ключом записи в данной таблице, внешний ключ на сотрудника, который выдал чек, внешний ключ мебели, которая была куплена, сумма покупки, а также сумма, которую дал покупатель.

1. Создание запросов для таблиц в DataSet

Запросы таблицы Staff.

На удаление сотрудника по ID:



Рис. 51. Запрос на удаление сотрудника

На получение сотрудника по ID:



Рис. 52. Запрос на получение сотрудника по ID

На получение сотрудника по логину:



Рис. 53. Запрос на получение сотрудника по логину

На добавление нового сотрудника:



Рис. 54. Запрос на добавление сотрудника

На обновление сотрудника по ID:



Рис. 55. Запрос на обновление сотрудника

Запросы таблицы Receipt:

На удаление чека по ID:



Рис. 56. Запрос на удаление чека по ID

На добавление чека:



Рис. 57. Запрос на добавление чека

На обновление чека по ID:



Рис. 58. Запрос на обновление чека

Запросы таблицы Furniture:

На удаление по мебели ID:



Рис. 59. Запрос на удаление мебели по ID

На получение мебели по ID:



Рис. 60. Запрос на получение мебели по ID

На получение мебели по названию:



Рис. 61. Запрос на получение мебели по названию

На добавление новой мебели:



Рис. 62. Запрос на добавление новой мебели

На обновление мебели по ID:



Рис. 63. Запрос на обновление мебели по ID

Запросы таблицы FurnitureTypes:

На удаление типа по ID:



Рис. 64. Запрос на удаление типа по ID

На получение типа по ID:



Рис. 65. Запрос на получение типа по ID

На получение типа по названию:



Рис. 66. Запрос на получение типа по названию

На добавление типа:



Рис. 67. Запрос на добавление типа

На обновление типа:



Рис. 68. Запрос на обновление типа

Запросы таблицы Factories:

На удаление завода по ID:



Рис. 69. Запрос на удаление завода по ID

На получение завода по ID:



Рис. 70. Запрос на получение завода по ID

На получение завода по адресу:



Рис. 71. Запрос на получение завода по адресу

На добавление нового завода:



Рис. 72. Запрос на добавление завода

На обновление завода по ID



Рис. 73. Запрос на обновление завода

Запрос таблицы FurnitureCarriers:

На удаление перевозчика по ID:



Рис. 74. Запрос на удаление перевозчика по ID

На получение перевозчика по ID:



Рис. 75. Запрос на получение перевозчика по ID

На получение перевозчика по названию:



Рис. 76. Запрос на получение перевозчика по названию

На добавление нового перевозчика:



Рис. 77. Запрос на добавление нового перевозчика

На обновление перевозчика по ID:



Рис. 78. Запрос на обновление перевозчика по ID

Запросы таблицы Machines:

На удаление станка по ID:



Рис. 79. Запрос на удаление станка по ID

На получение станка по ID:



Рис. 80. Запрос на получение станка по ID

На получение станка по имени:



Рис. 81. Запрос на получение станка по имени

На добавление станка:



Рис. 82. Запрос на добавление нового станка

На обновление станка по ID:



Рис. 83. Запрос на обновление станка по ID

Запросы таблицы Materials:

На удаление материала по ID:



Рис. 84. Запрос на удаление материала по ID

На получение материала по ID:



Рис. 85. Запрос на получение материала по ID

На получение материала по имени:



Рис. 86. Запрос на получение материала по имени

На добавление материала:



Рис. 87. Запрос на добавление нового материала

На обновление материала ID:



Рис. 88. Запрос на обновление материала по ID

Запросы таблицы MaterialsSuppliers:

На удаление поставщика материалов по ID:



Рис. 89. Запрос на удаление поставщика материалов по ID

На получение поставщика материалов по ID:



Рис. 90. Запрос на получение поставщика материалов по ID

На получение поставщика материалов по имени:



Рис. 91. Запрос на получение поставщика материалов по имени

На добавление поставщика материалов:



Рис. 92. Запрос на добавление поставщика материалов

На обновление поставщика материалов по ID:



Рис. 93. Запрос на обновление поставщика материалов по ID

Запросы таблицы Post:

На удаление должности по ID:



Рис. 94. Запрос на удаление должности по ID

На получение должности по ID:



Рис. 95. Запрос на получение должности по ID

На получение должности по имени:



Рис. 96. Запрос на получение должности по имени

На добавление должности:



Рис. 97. Запрос на добавление должности

На обновление должности по ID:



Рис. 98. Запрос на обновление должности по ID

1. Написание кода приложения

Для начала напишем окно авторизации. Сделаем дизайн окна.

XAML файл:

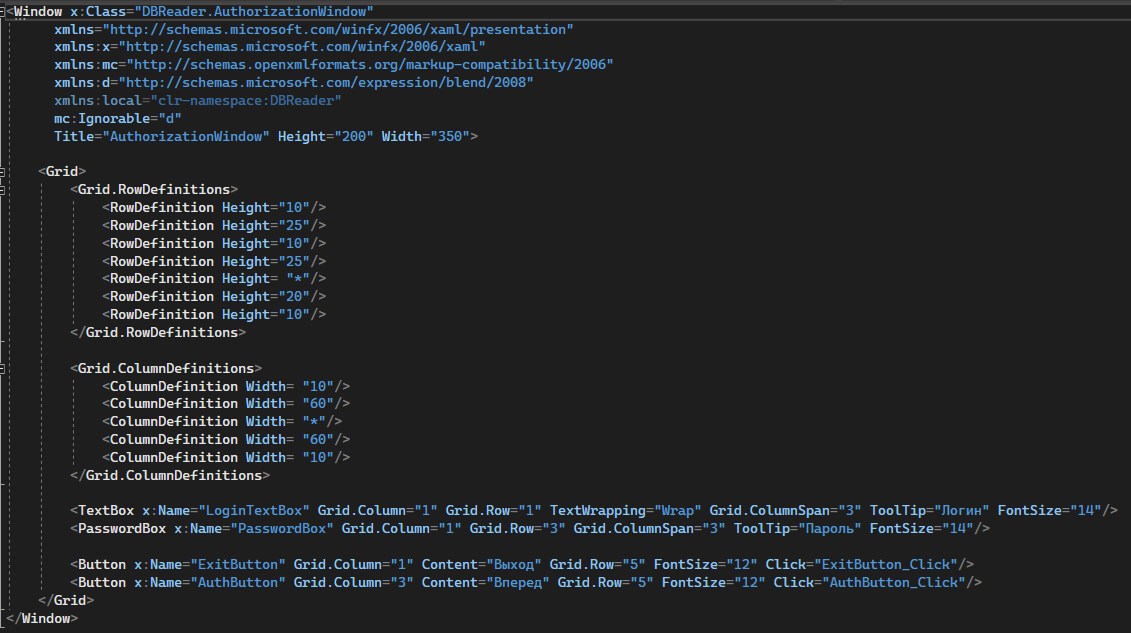


Рис. 99. XAML файл окна авторизации

Получившийся вид окна:

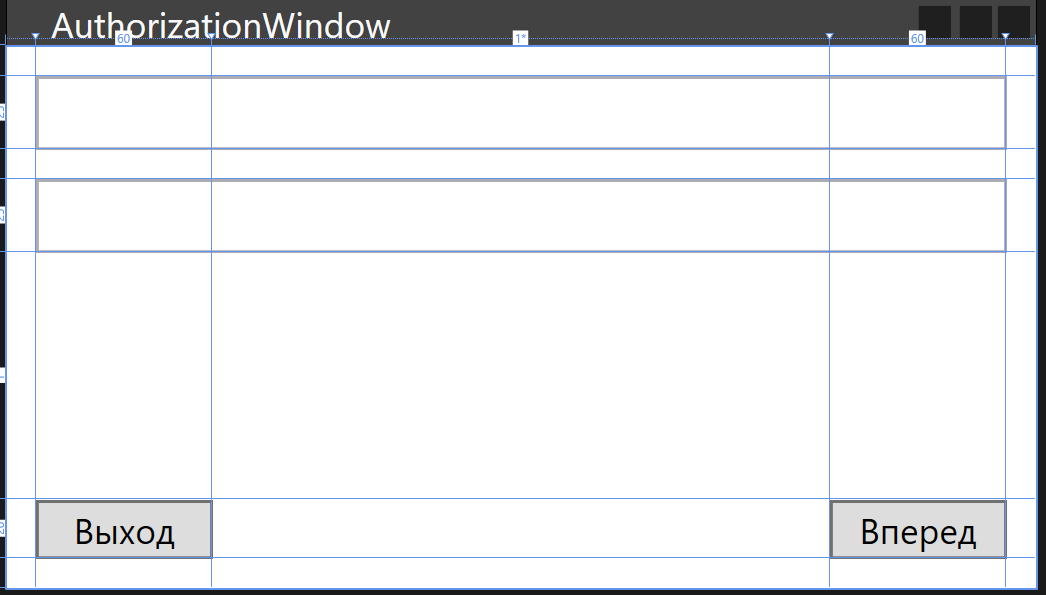


Рис. 100. Вид окна авторизации

Далее напишем саму логику класса окна для авторизации.

Сначала напишем поля класса, а также свойства для них:



Рис. 101. Поля класса авторизации

Далее напишем основную логику окна. При нажатии кнопки “Вперед” у нас будет вызываться метод AuthButton\_Click, при нажатии на кнопку “Назад” будет вызываться метод ExitButton\_Click.

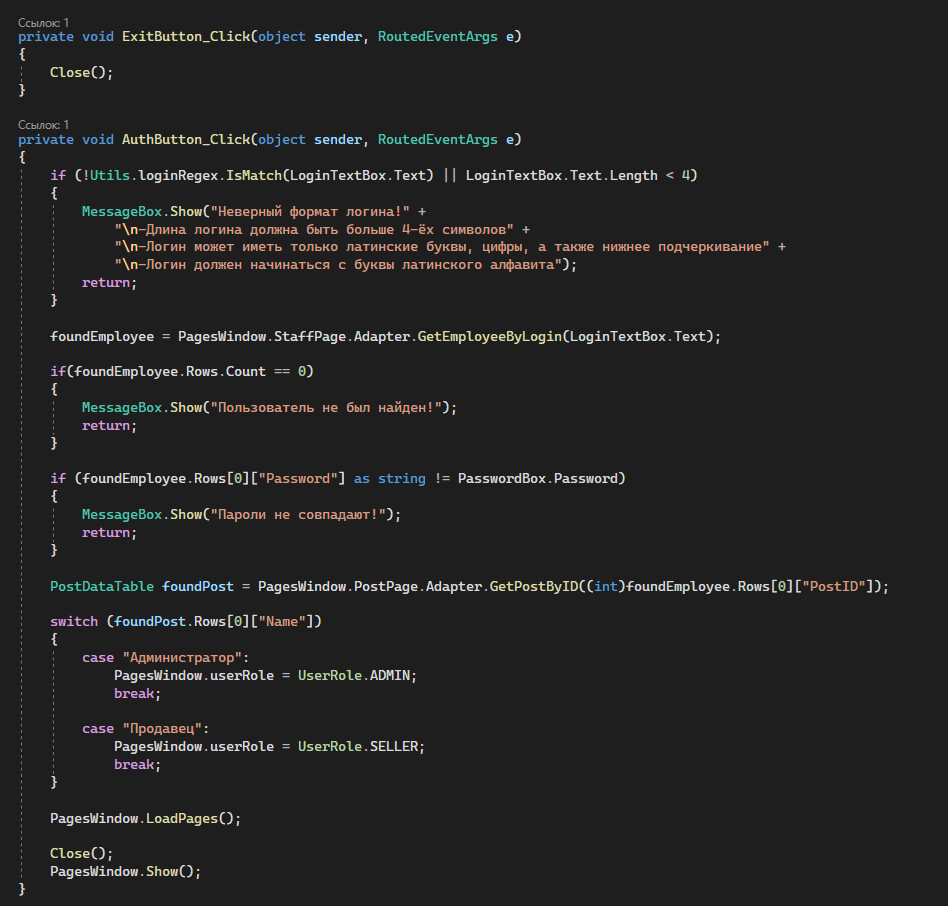


Рис. 102. Основная логика класса авторизации

Далее сделаем окно, в котором будут меняться страницы таблиц.

Объявим поля и реализуем свойства для них:

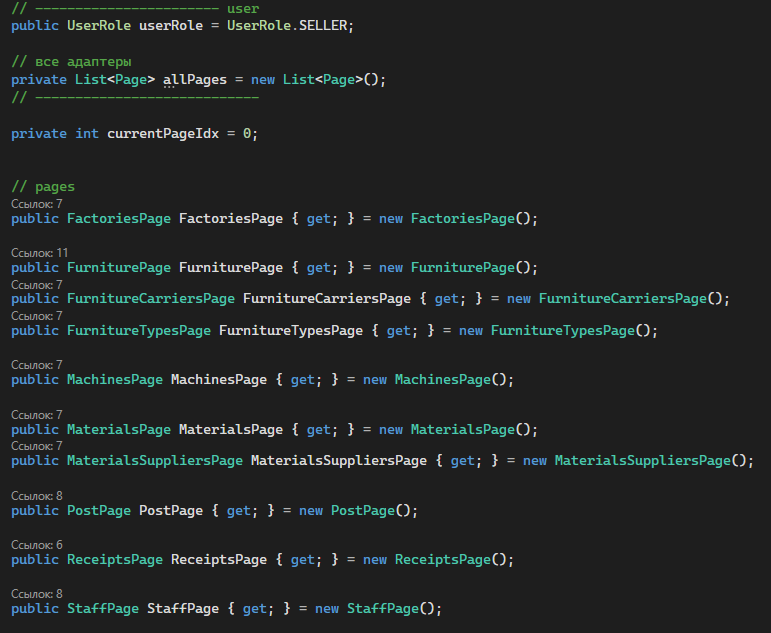


Рис. 103. Поля и свойства класса для страниц

Далее напишем логику для окна. В класс будет всего два метода: один для инициализации необходимых страниц, а другой для перелистывания страниц и обновления содержимого таблиц.

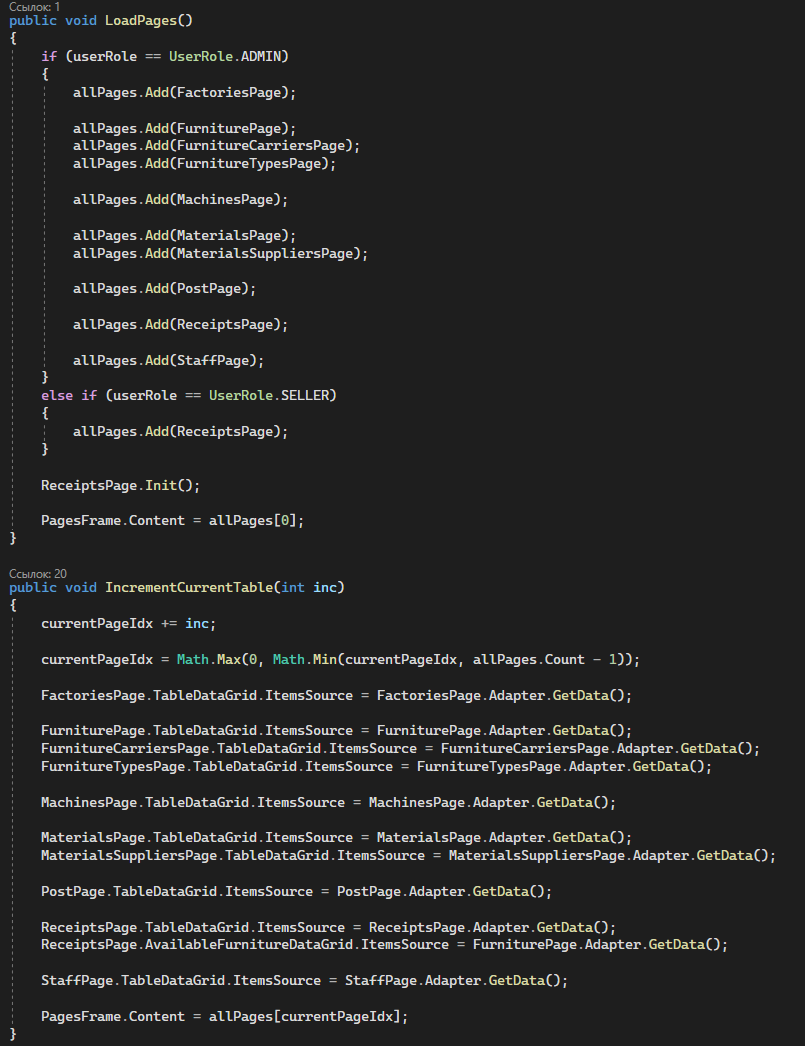


Рис. 104. Логика класса окна для страниц

Далее опишем XAML файл для страницы с таблицей заводов:

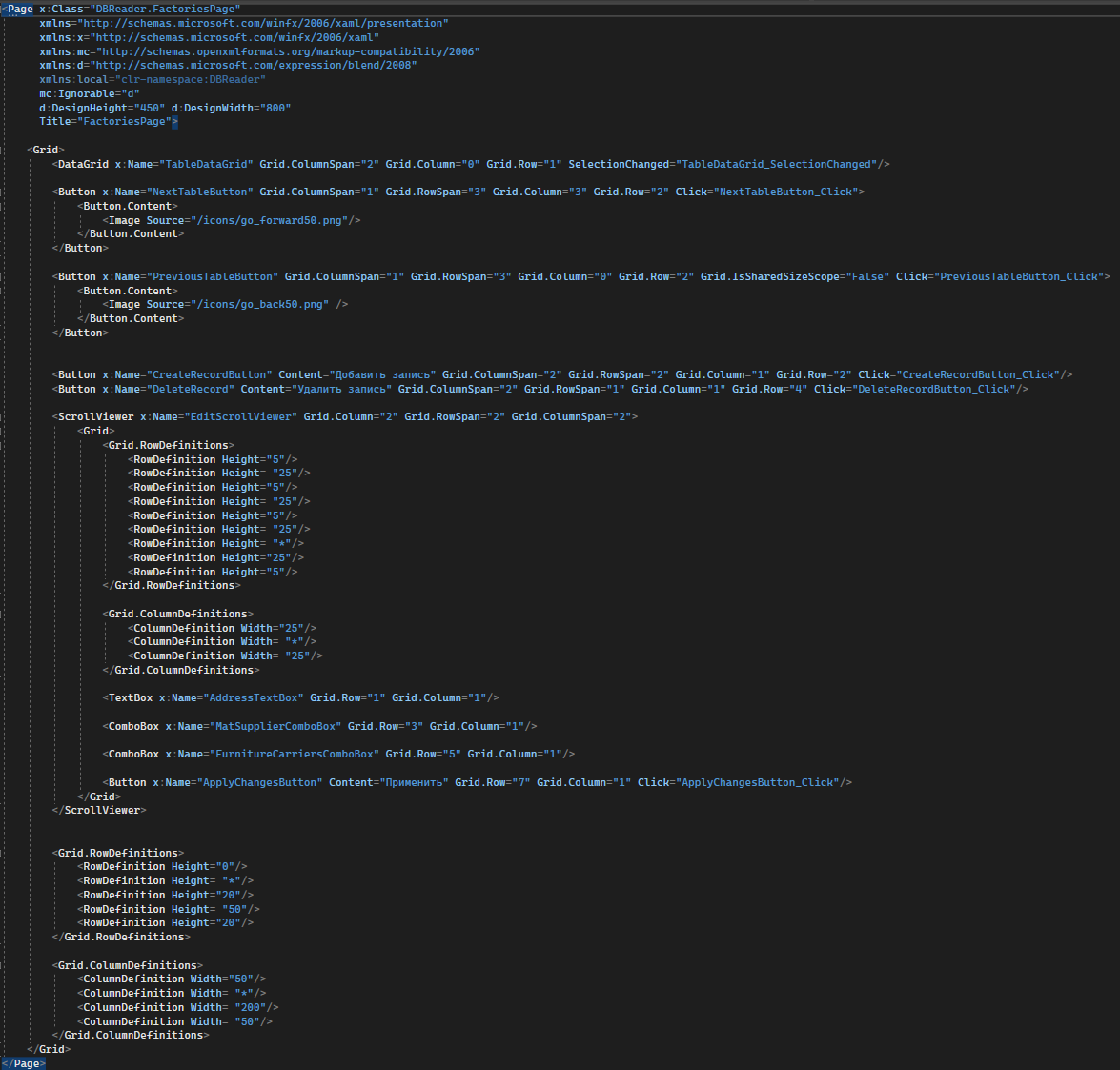


Рис. 105. XAML файл страницы с заводами

Получившийся вид страницы:

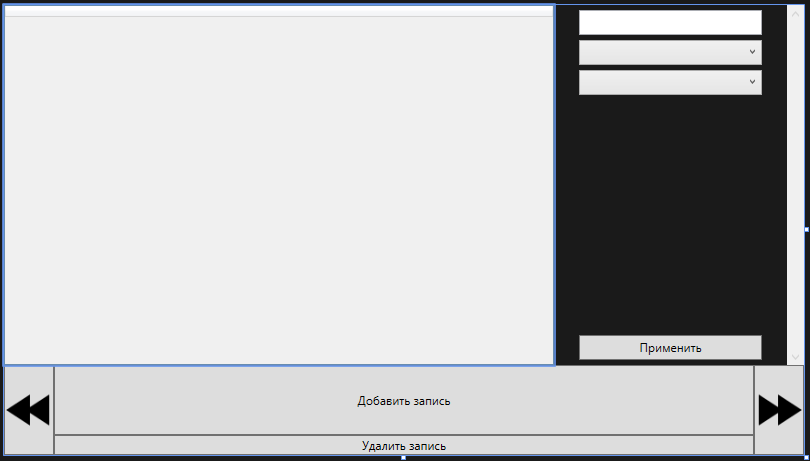


Рис. 104. Получившийся вид страницы заводов

Далее реализуем логику класса страницы.

Данные методы срабатывают при нажатии на кнопки Назад и Вперед и меняют страницы соответственно. Такие методы будут встречаться во всех таблицах, и они будут одинаковыми, поэтому далее описываться не будут.

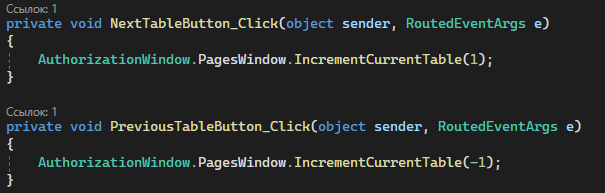


Рис. 105. Методы для изменения текущей страницы

Далее реализуем методы, которые будут создавать пустую запись и удалять все выделенные записи. Они будут срабатывать при нажатии на соответствующие кнопки. Такие методы будут встречаться во всех таблицах, и они будут почти одинаковыми, поэтому далее описываться не будут.



Рис. 106. Методы для удаления записей и создания записи

Далее реализуем для изменения содержимого полей для редактирования записей, а также для применения редактирования к записи. Подобные метод с почти таким же кодом будут встречаться везде, поэтому дальше описываться не будут.

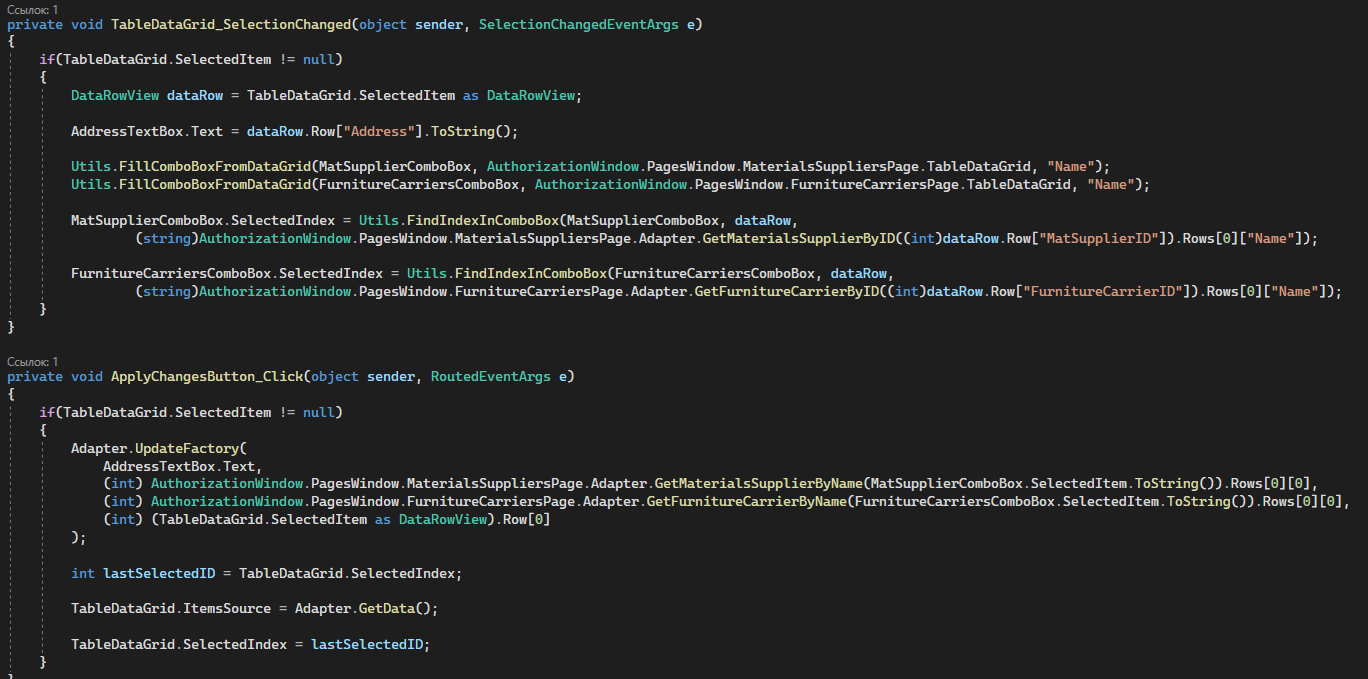


Рис. 107. Методы для изменения записи и применения изменения

Далее опишем окно с перевозчиками мебели. В логике класса почти ничего не поменялось.

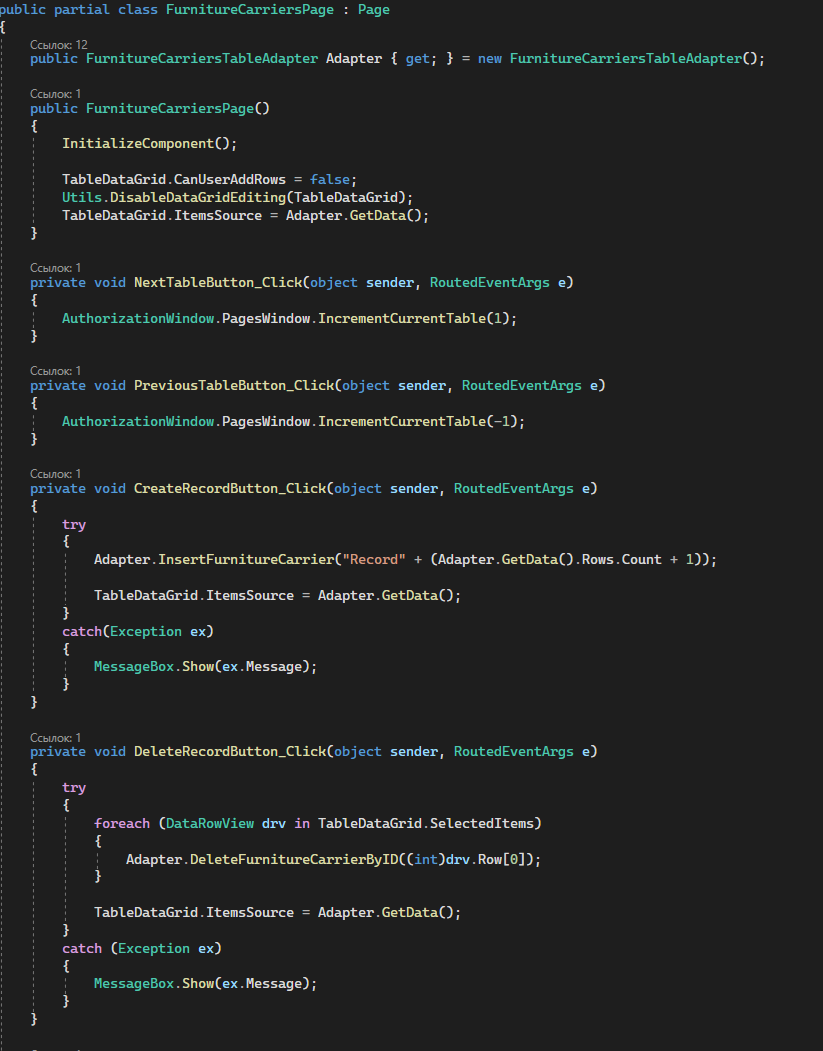


Рис. 108. Код для класса с перевозчиками мебели

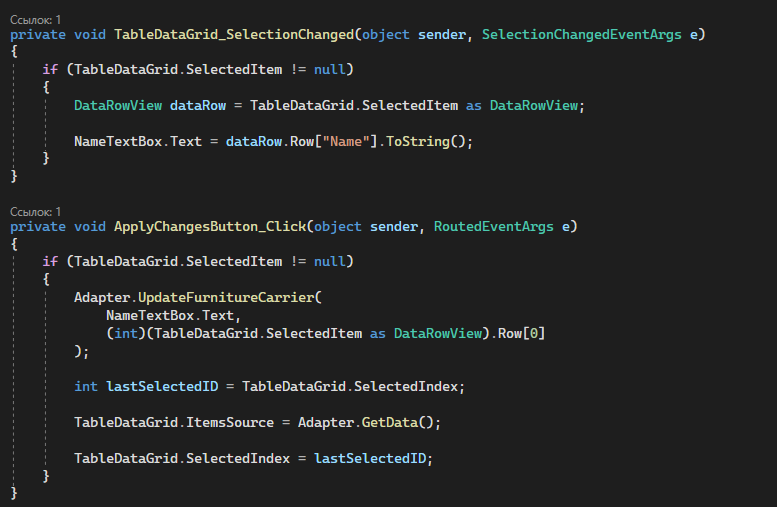


Рис. 109. Код для класса с перевозчиками мебели

Далее опишем окно с мебелью. В логике изменилось немного. Добавилось два метода, которые ограничивают ввод в TextBox, в которые нужно вводить только числа. Подобные ограничения будут встречаться везде.

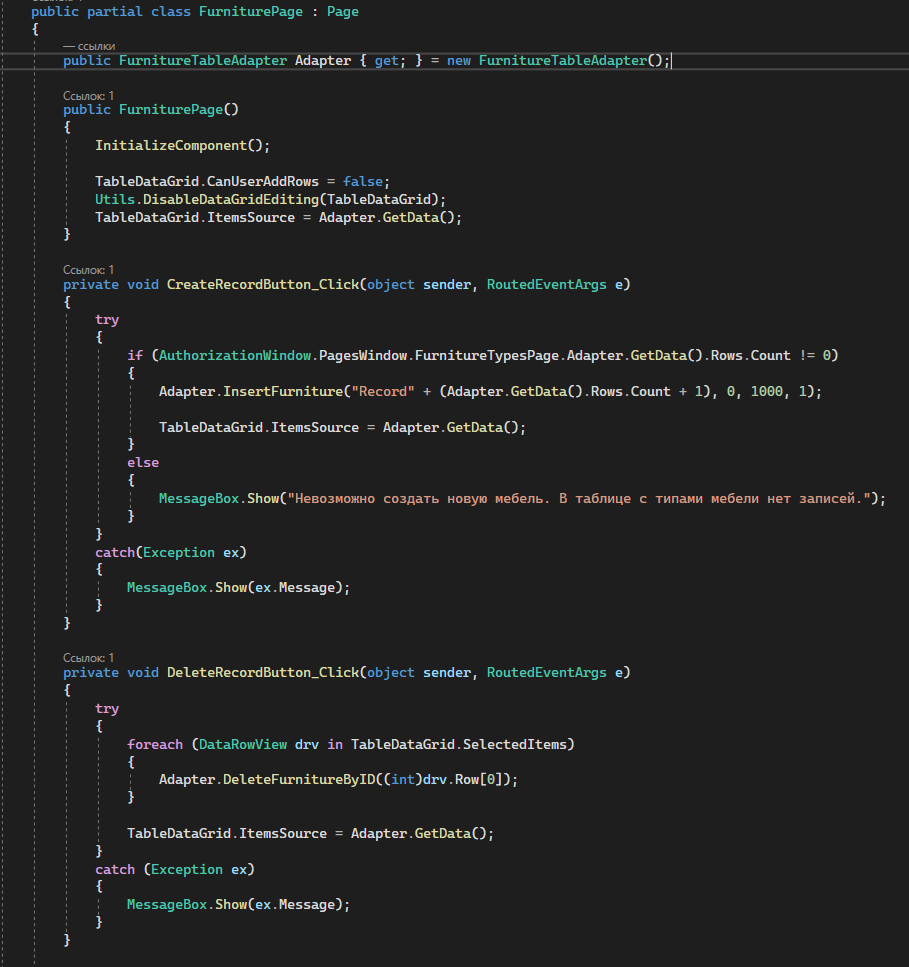


Рис. 110. Код для класса со страницей мебели

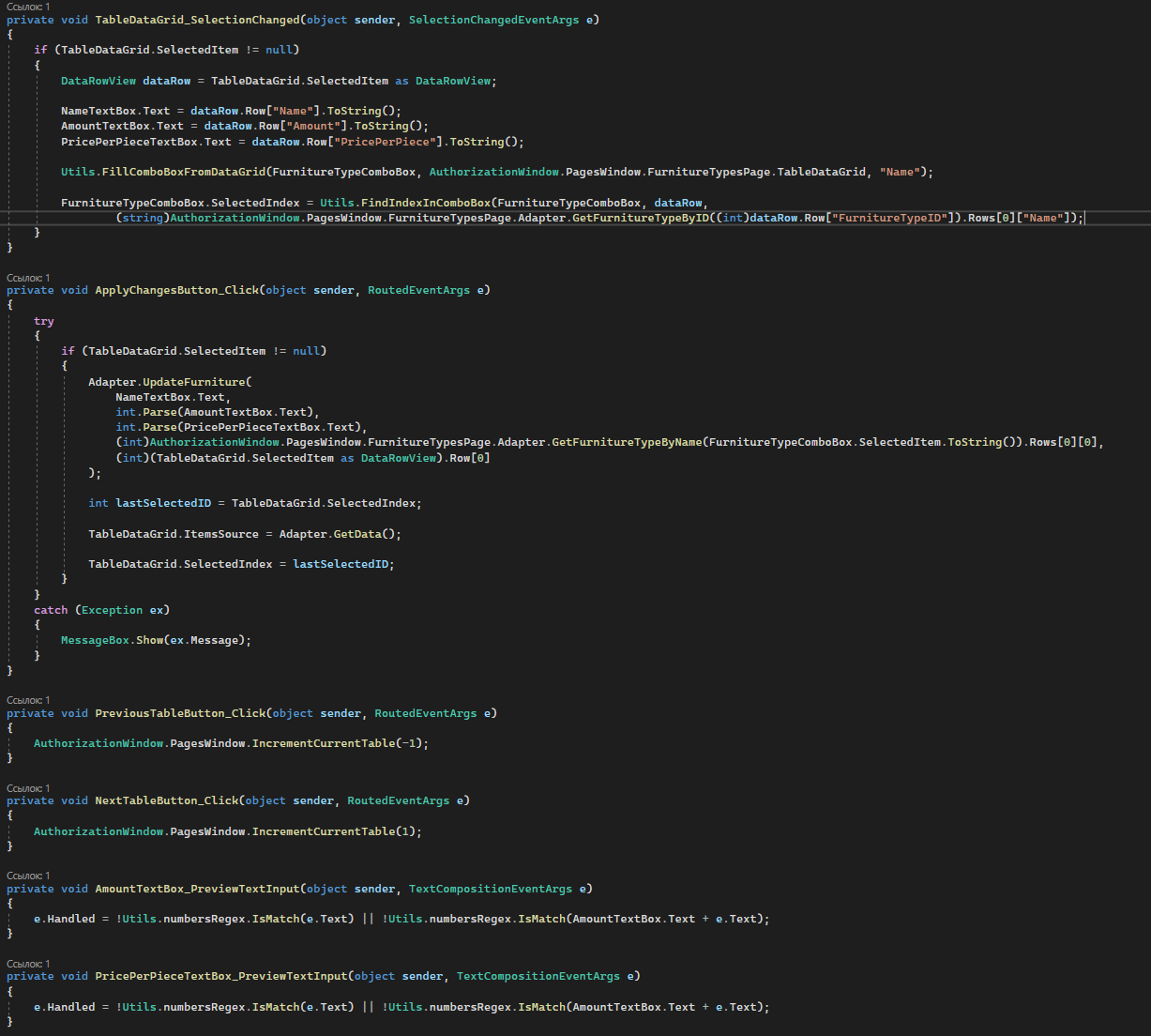


Рис. 111. Код для класса со страницей мебели

Далее опишем логику класса страницы с типами мебели:



Рис. 112. Логика класса страницы с типами мебели

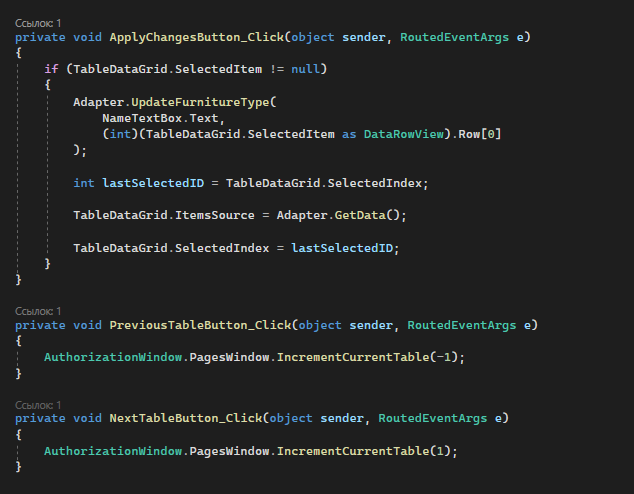


Рис. 113. Логика класса страницы с типами мебели

Далее реализуем логику для класса страницы для станков:



Рис. 114. Логика класса страницы для станков



Рис. 115. Логика класса страницы для станков

Напишем логику класса для страницы материалов:



Рис. 116. Логика класса для страницы с материалами

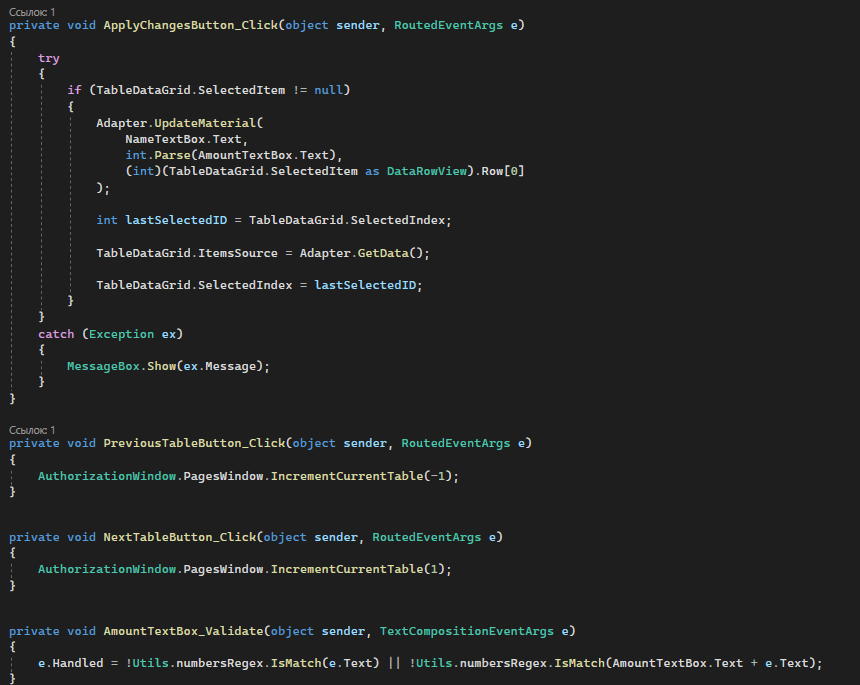


Рис. 117. Логика класса для страницы с материалами

Реализуем логику класса для страницы с поставщиками материалов:

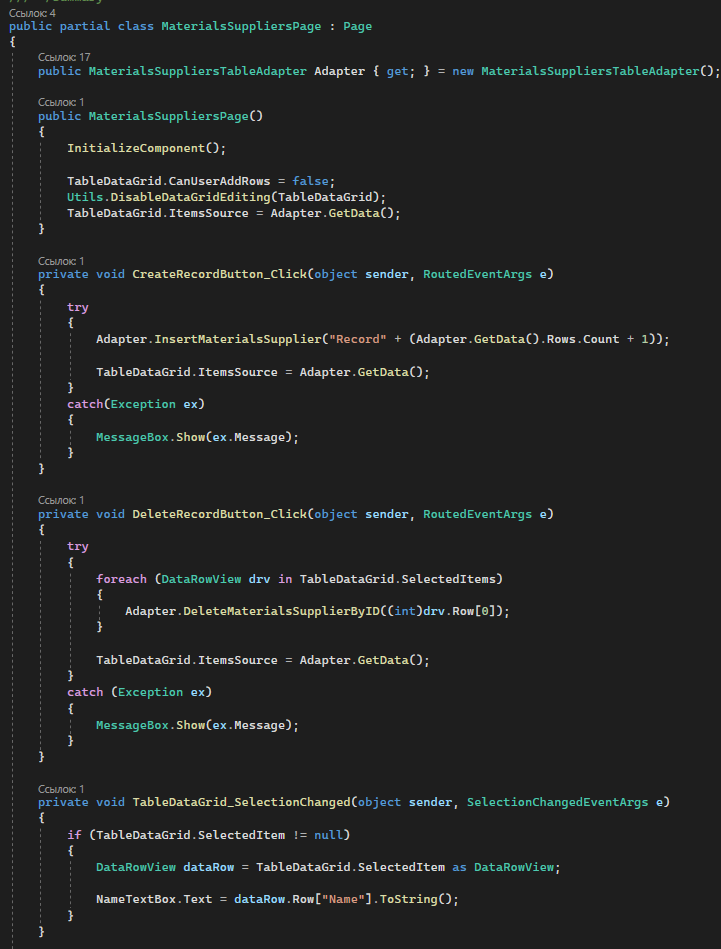


Рис. 118. Логика класса для страницы с поставщиками материалов

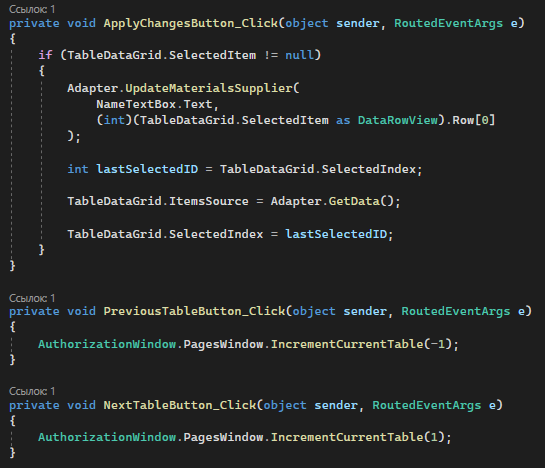


Рис. 119. Логика класса для страницы с поставщиками материла

Здесь уже появляется новый код для сохранения таблицы в файлы и загрузки таблицы из файла. Подобный код будет встречаться еще в одной таблице, поэтому больше описываться не будет.



Рис. 120. Код для экспорта/импорта таблицы

Далее напишем логику класса для страницы с должностями:



Рис. 121. Логика класса для страницы с должностями

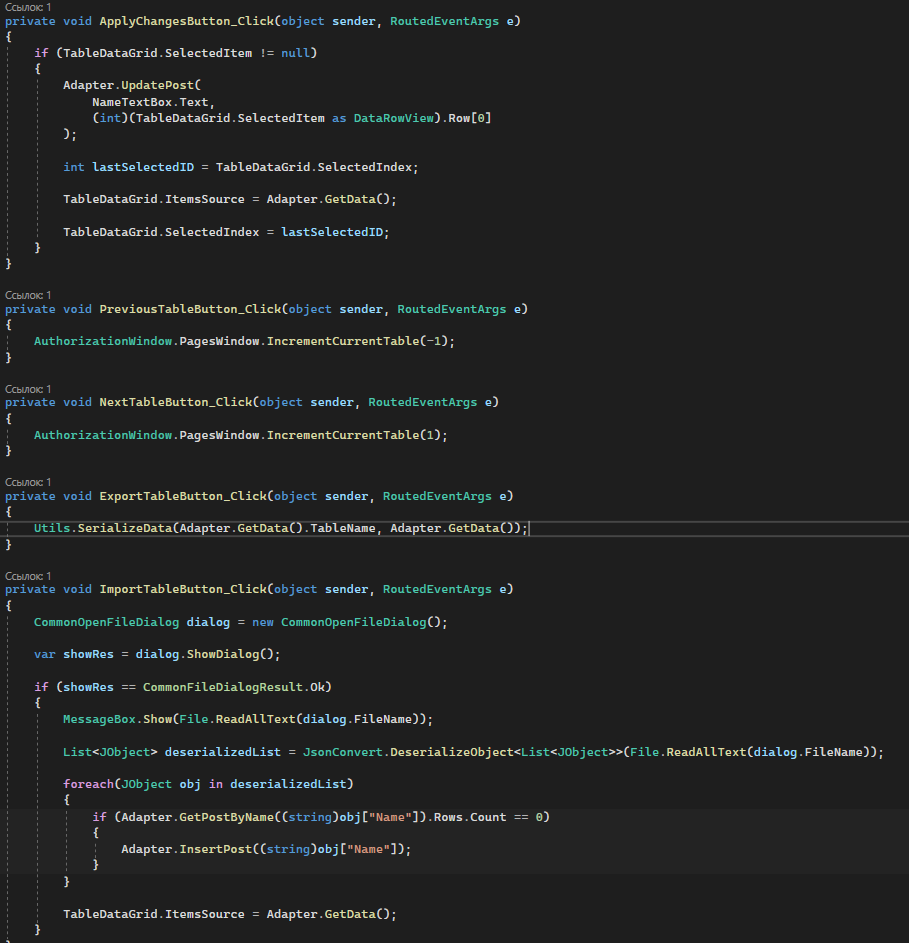


Рис. 122. Логика класса для страницы с должностями

Далее опишем код XAML файла для страницы с чеками:

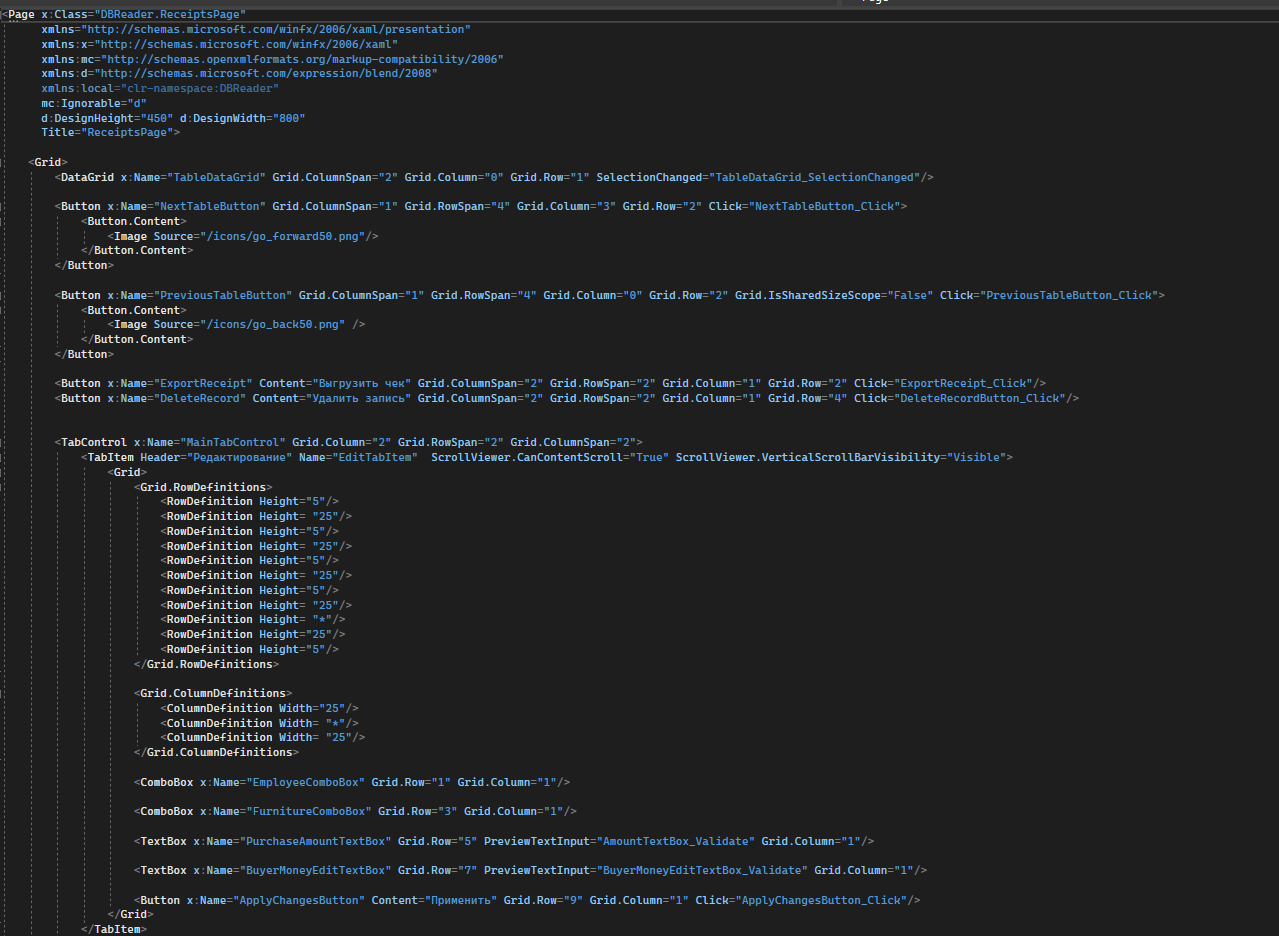


Рис. 123. XAML файл для страницы с чеками

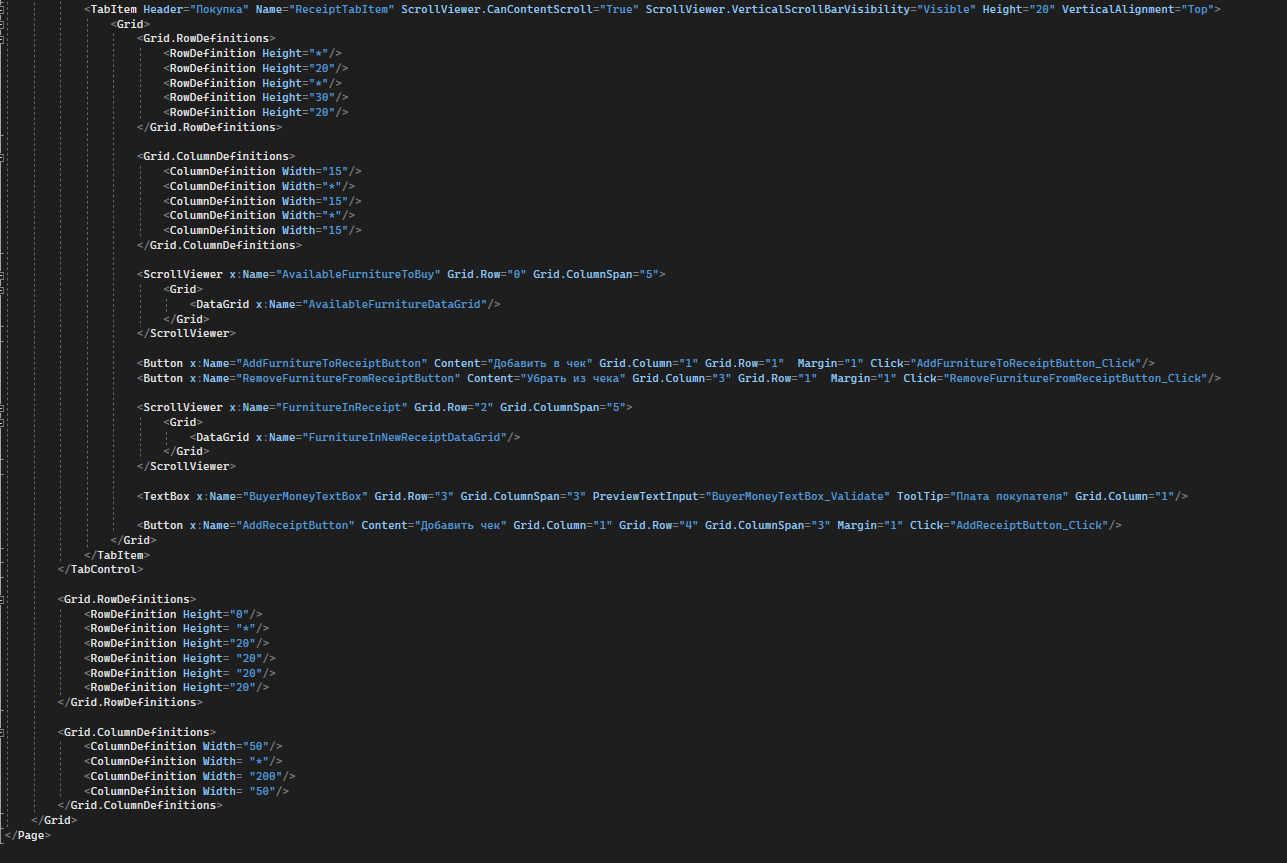


Рис. 124. XAML файл для страницы с чеками

Получившийся вид страницы:

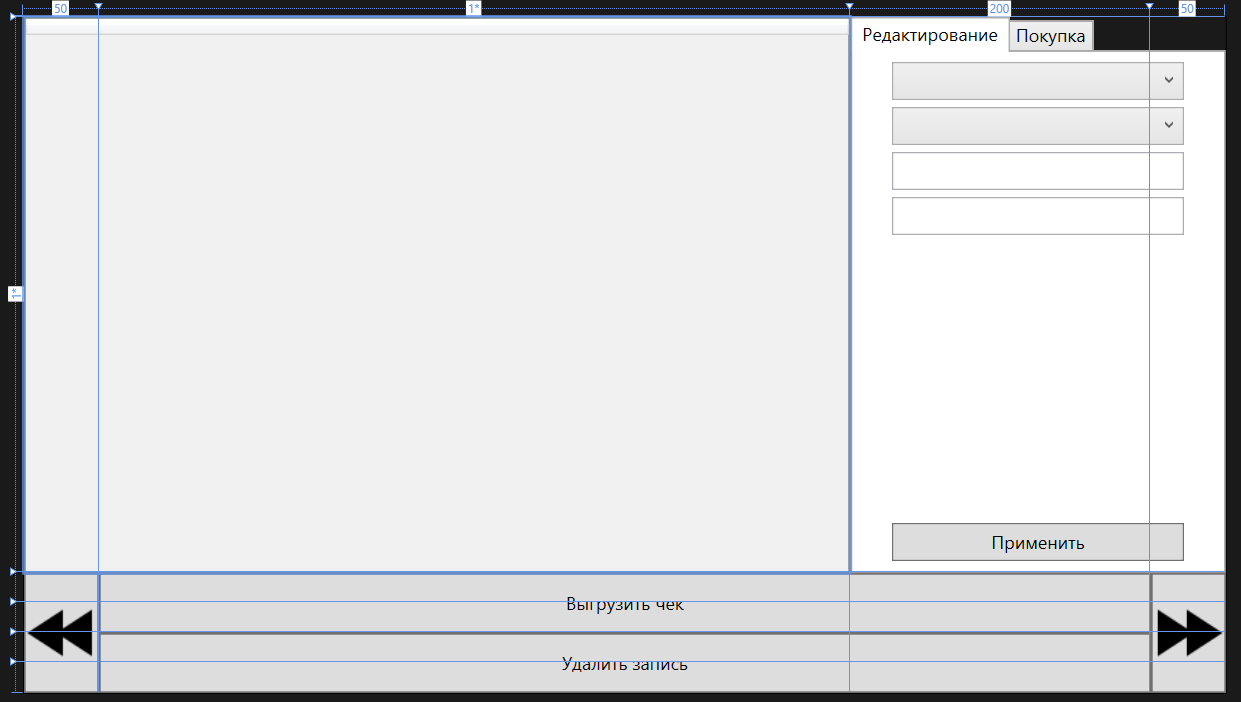


Рис. 125. Вид страницы с чеками

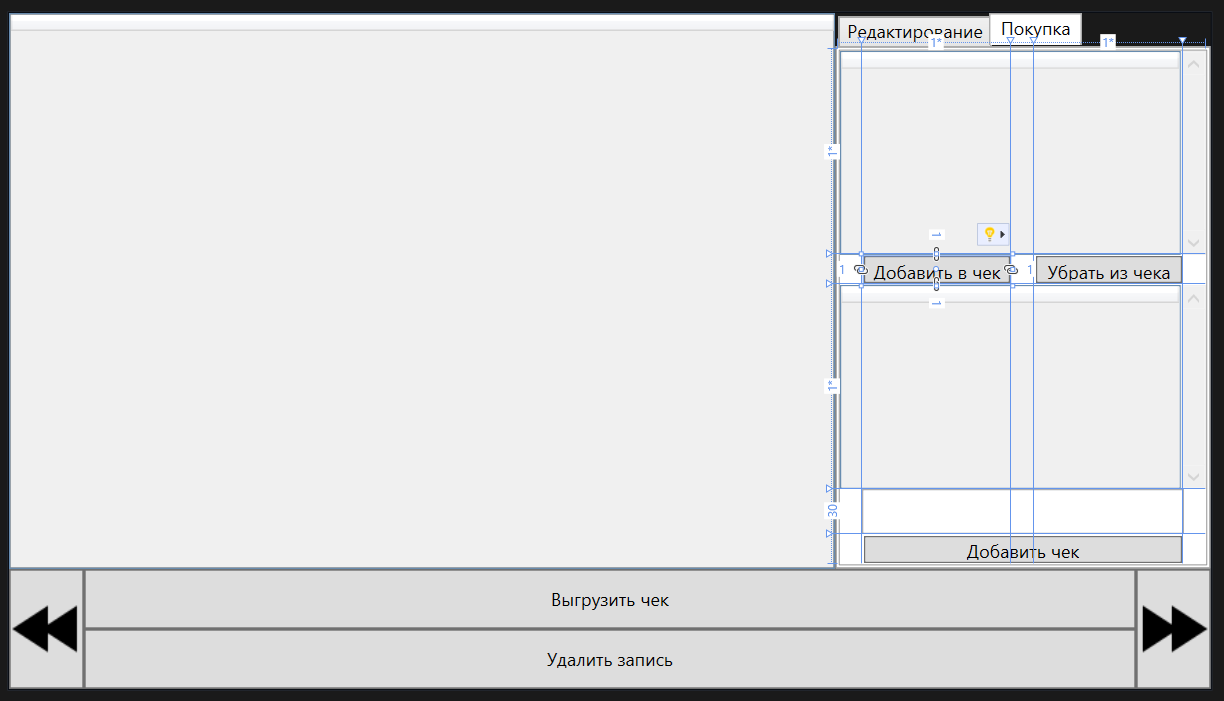


Рис. 126. Вид страницы с чеками

Далее опишем код страницы с чеками. Здесь появляются новые методы и обработки для добавления, удаление товаров в чек, для сохранения нового чека, а также чеков из БД.



Рис. 127. Код класса страницы с чеками



Рис. 128. Код класса страницы с чеками

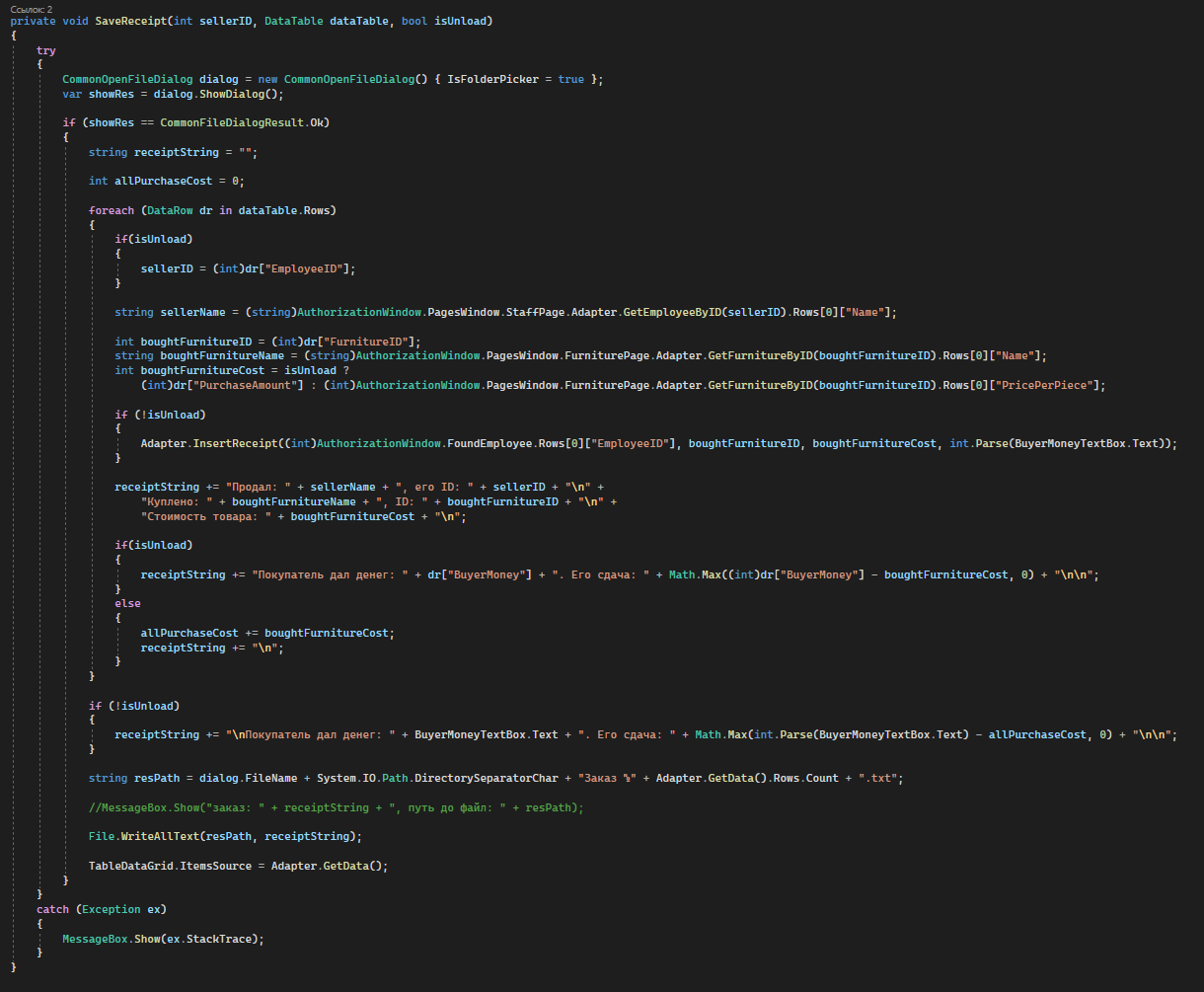


Рис. 129. Код класса страницы с чеками

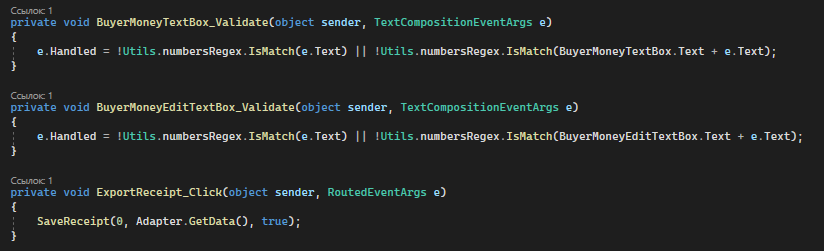


Рис. 130. Код класса страницы с чеками

Далее опишем код для страницы с сотрудниками:

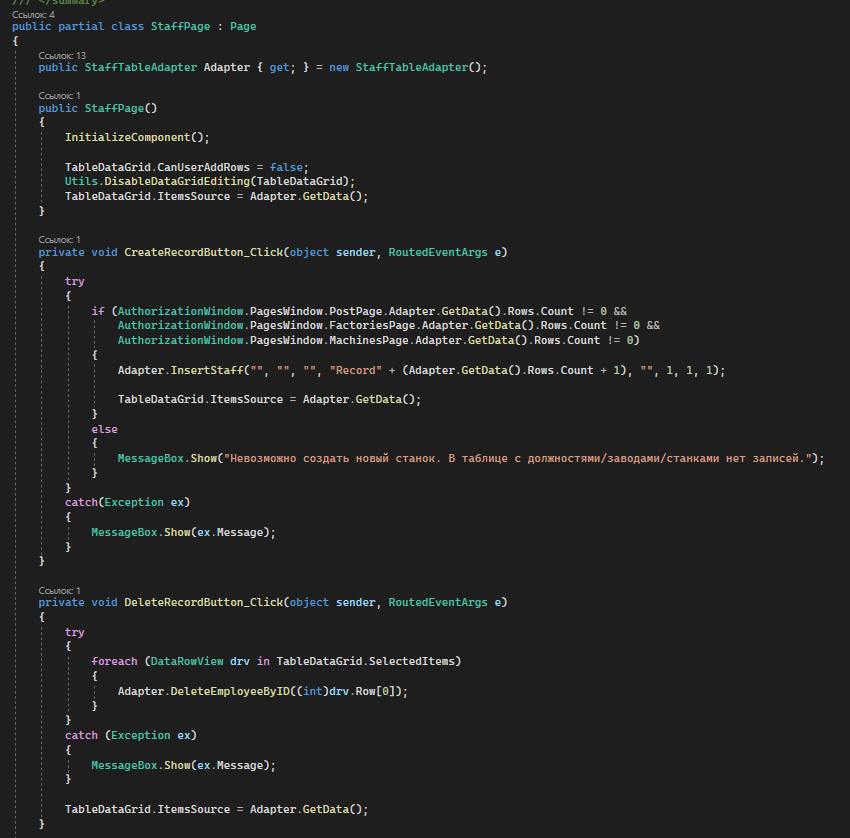


Рис. 131. Код класса страницы с сотрудниками

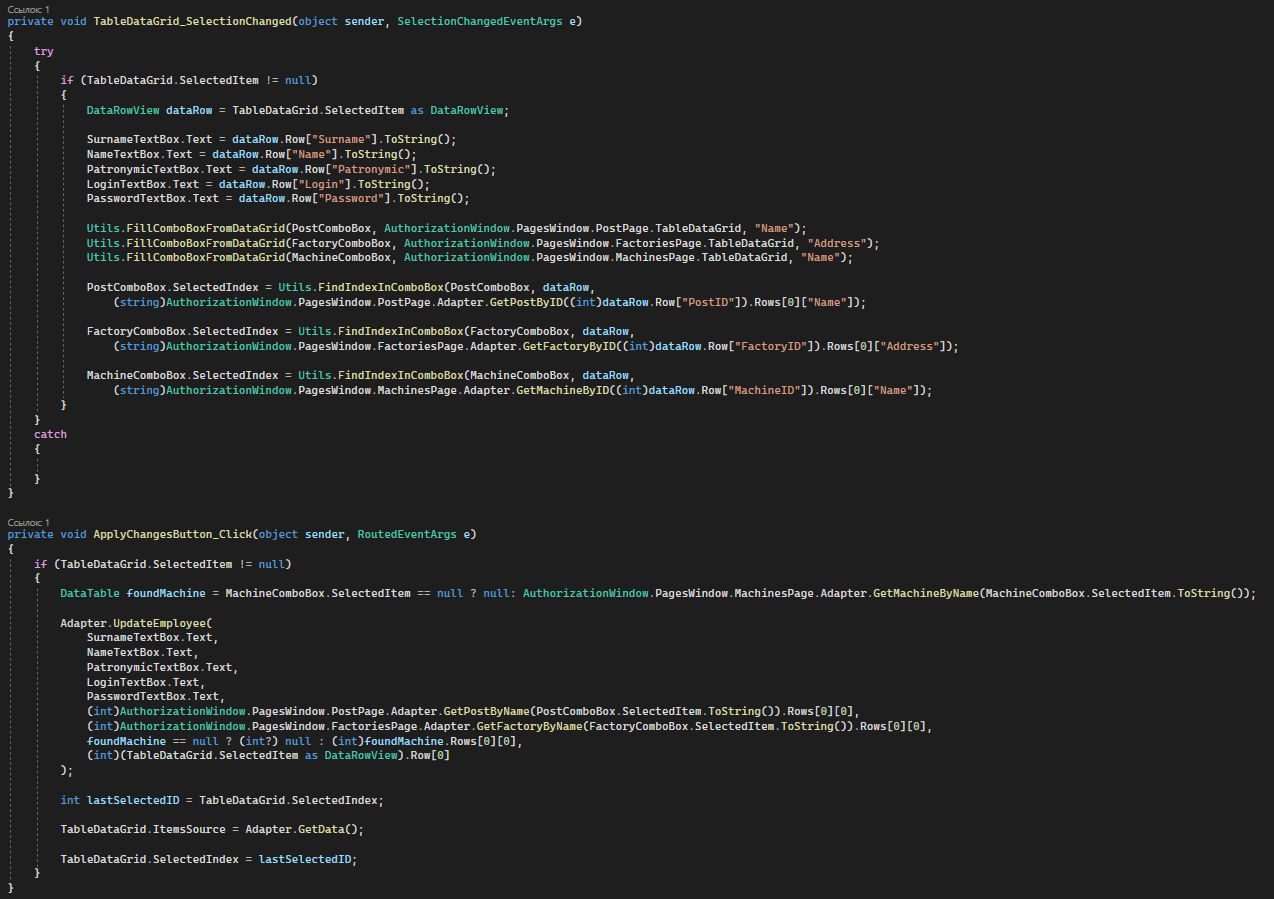


Рис. 131. Код класса страницы с сотрудниками



Рис. 131. Код класса страницы с сотрудниками

Вывод: разработали информационную систему для компании по производству мебели, а также базу данных для неё.