|  |
| --- |
| **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.

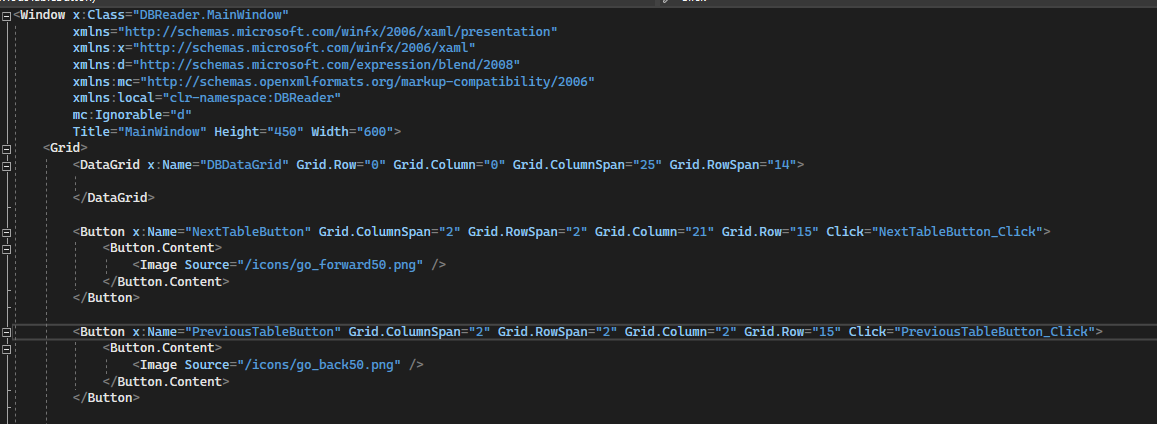


Рис. 1. Добавление кнопок и вида таблиц

Затем сделаем грид, необходимый для адаптивной верстки.

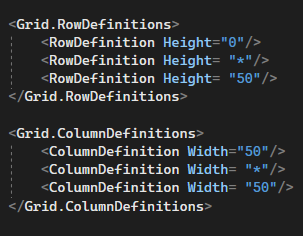


Рис. 2. Грид для адаптивной верстки

1. Создание базы данных, таблиц и SQL запросов

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

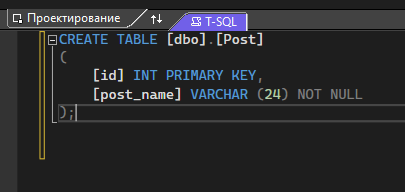


Рис. 3. SQL запрос для создания таблицы с должностями

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

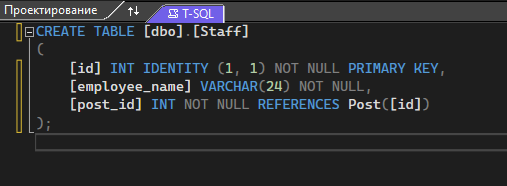


Рис. 4. Создание таблицы с сотрудниками

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

В итоге получились две таблицы:

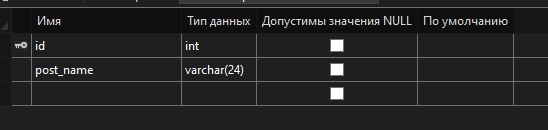


Рис. 5. Таблица должностей

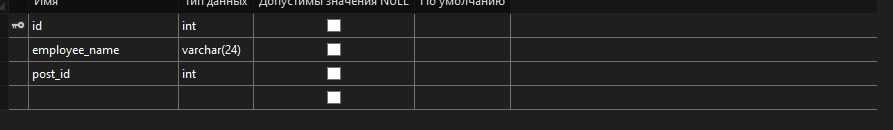


Рис. 6. Таблица сотрудников

1. Написание логики приложения

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

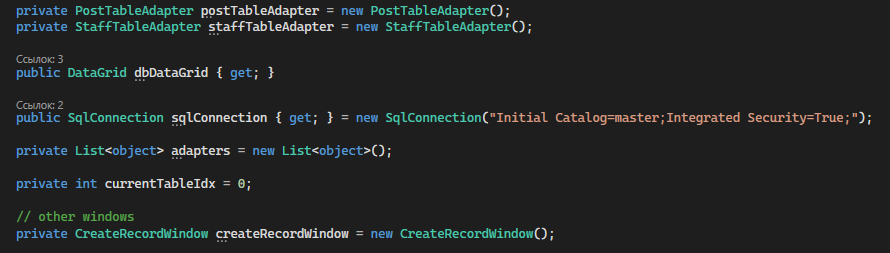


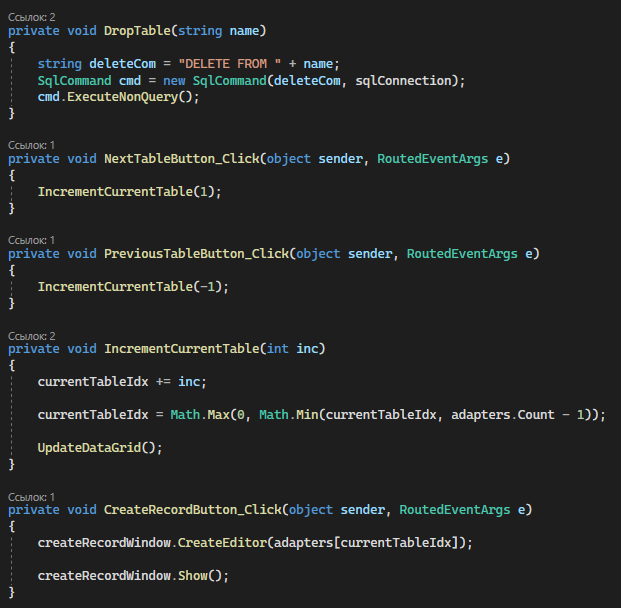
Рис. 7. Объявление и инициализация переменных

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:

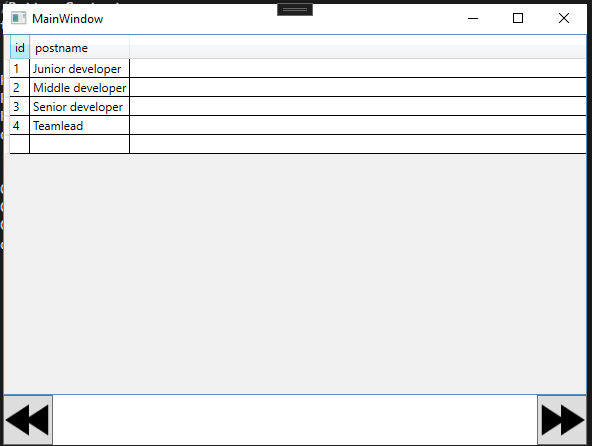


Рис. 10. Результат разработки

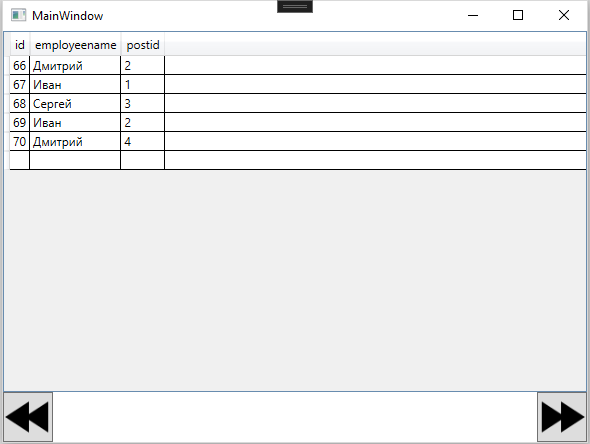


Рис. 11. Результат разработки

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.

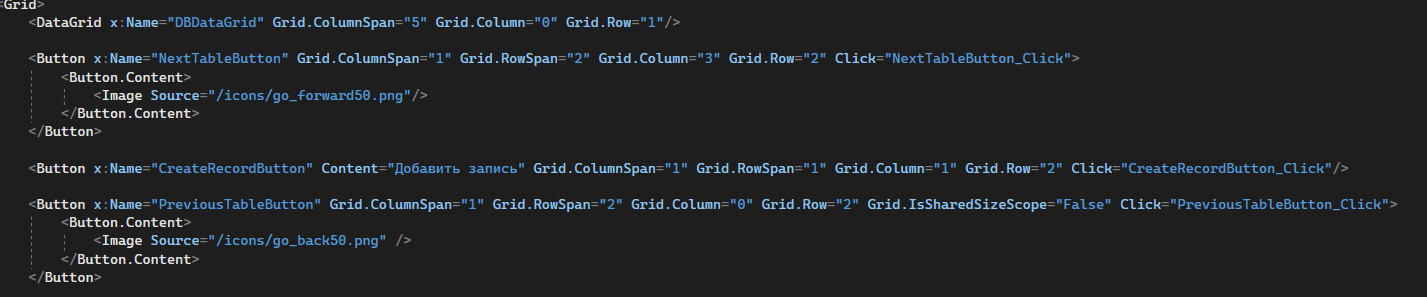


Рис. 11. Оформление интерфейса

Далее оформим грид, необходимый для адаптивной верстки.

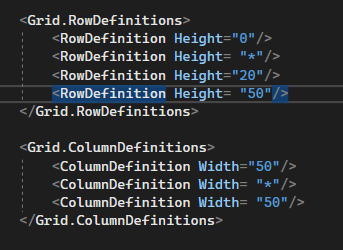


Рис. 12. Оформление грида

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.

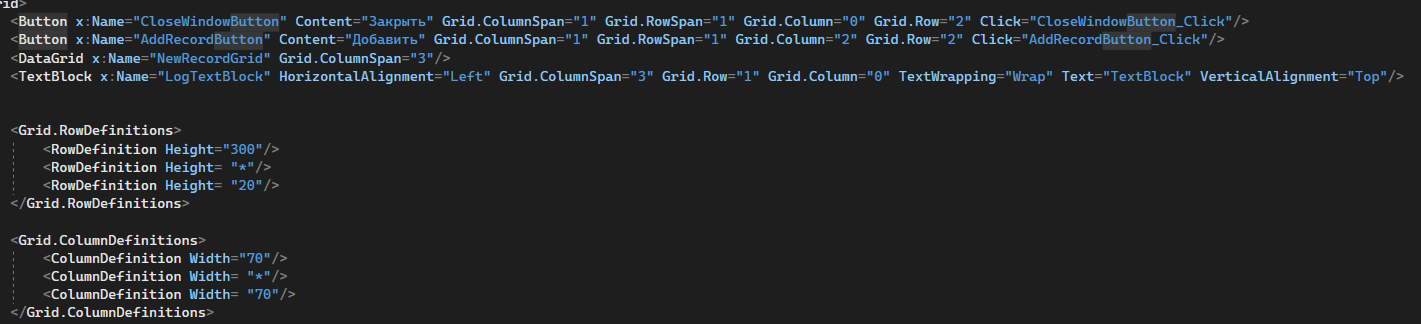


Рис. 13. Оформление второго окна

* + - 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.

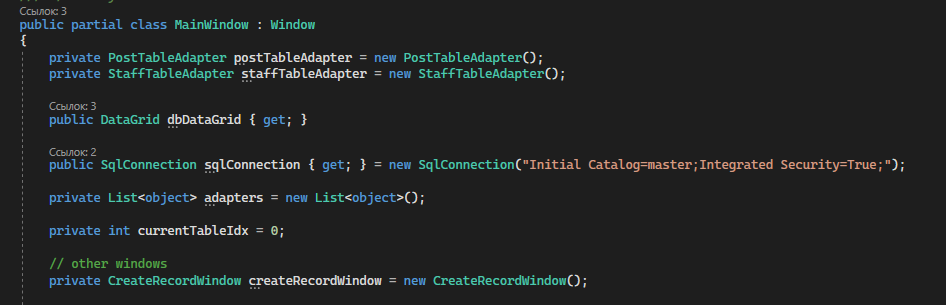


Рис. 14. Поля класса MainWindow

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.

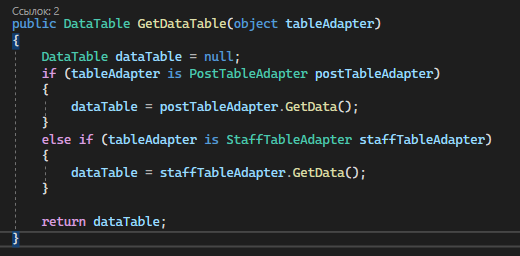


Рис. 16. Метод для получения DataTable

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

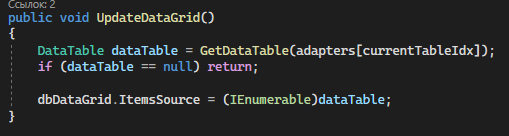


Рис. 17. Метод для обновления DataGrid

* + - 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

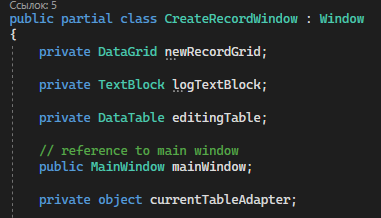


Рис. 18. Поля класса CreateRecordWindow

Далее напишем конструктор, в котором найдем необходимые элементы.

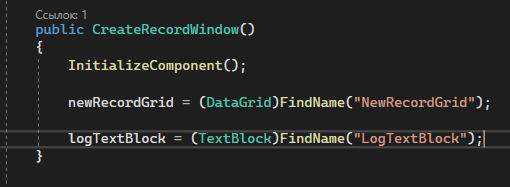


Рис. 19. Конструктор класса CreateRecordWindow

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

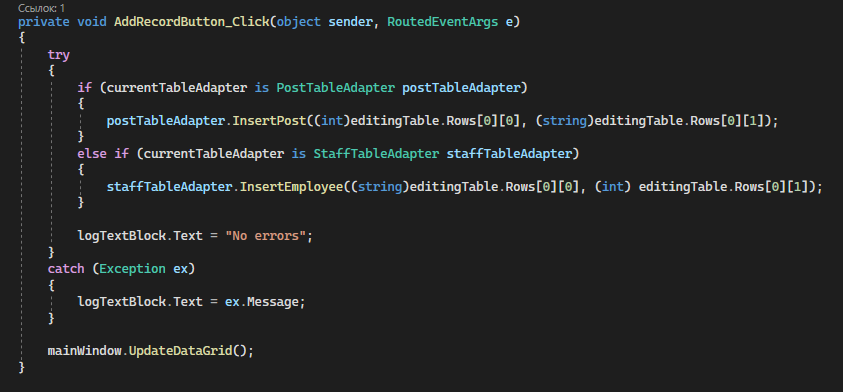


Рис. 20. Методы для формирования и выполнения SQL запроса

Напишем метод для создания редактора новой записи.

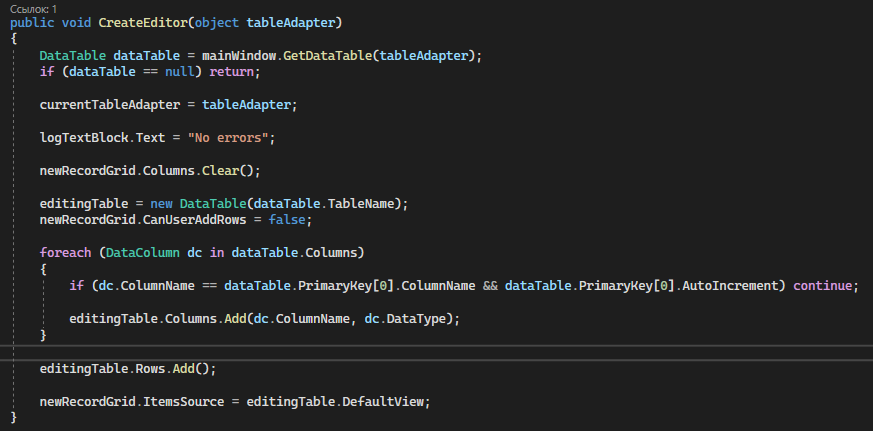


Рис. 21. Метод для создания редактора

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

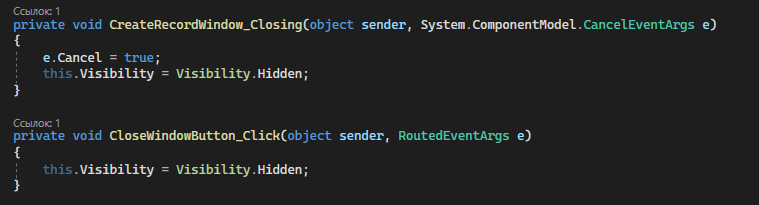


Рис. 22. Метод для скрытия окна

4. Результаты:

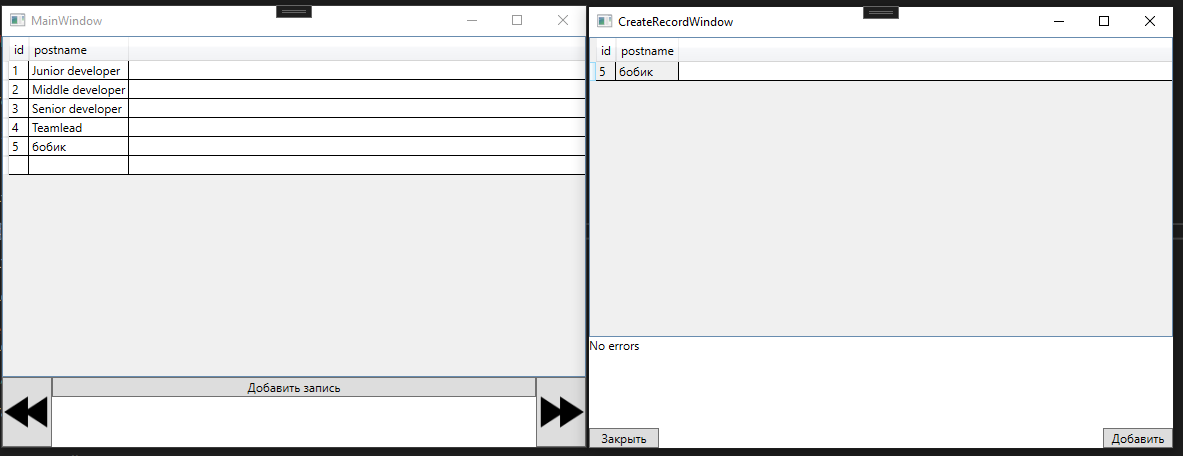


Рис. 23. Результаты

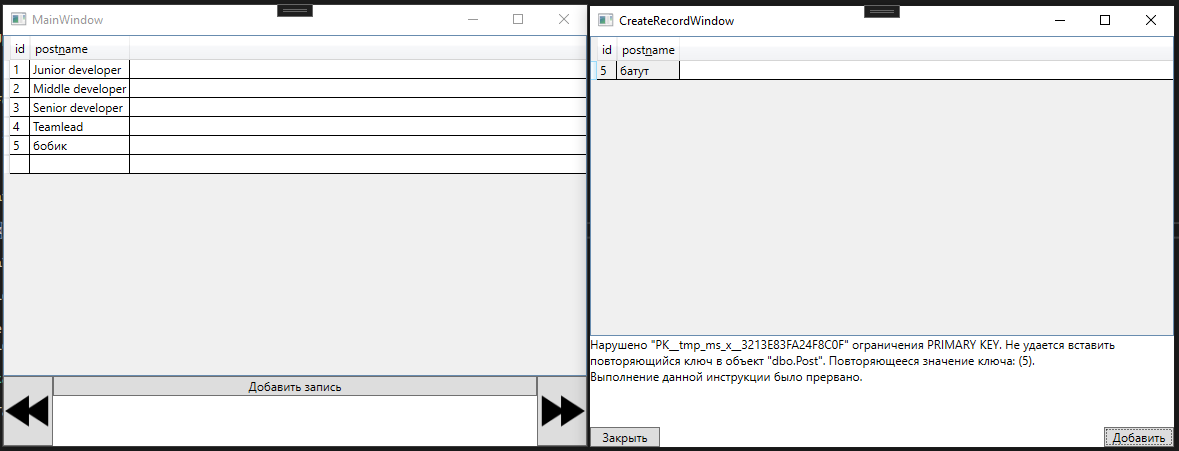
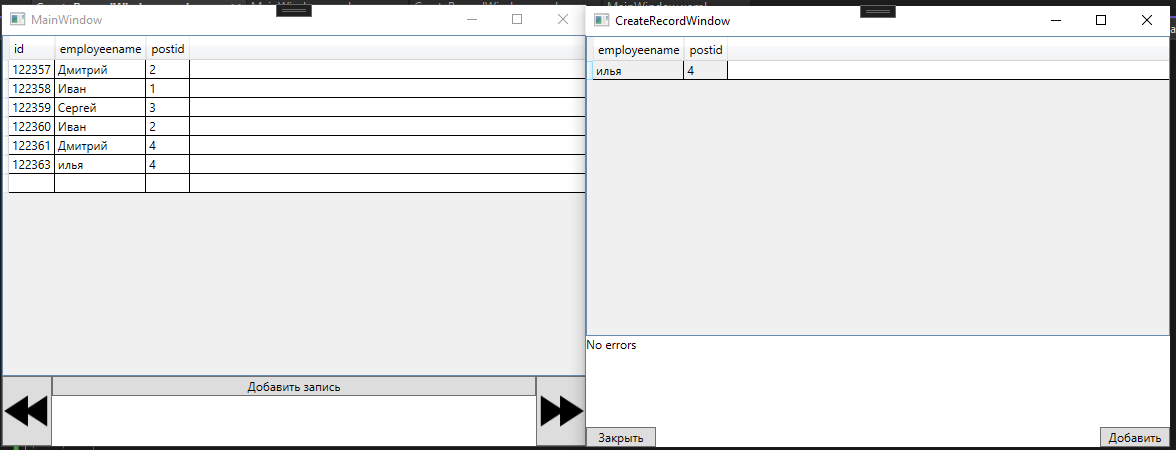


Рис. 24. Результаты

Рис. 25. Результаты

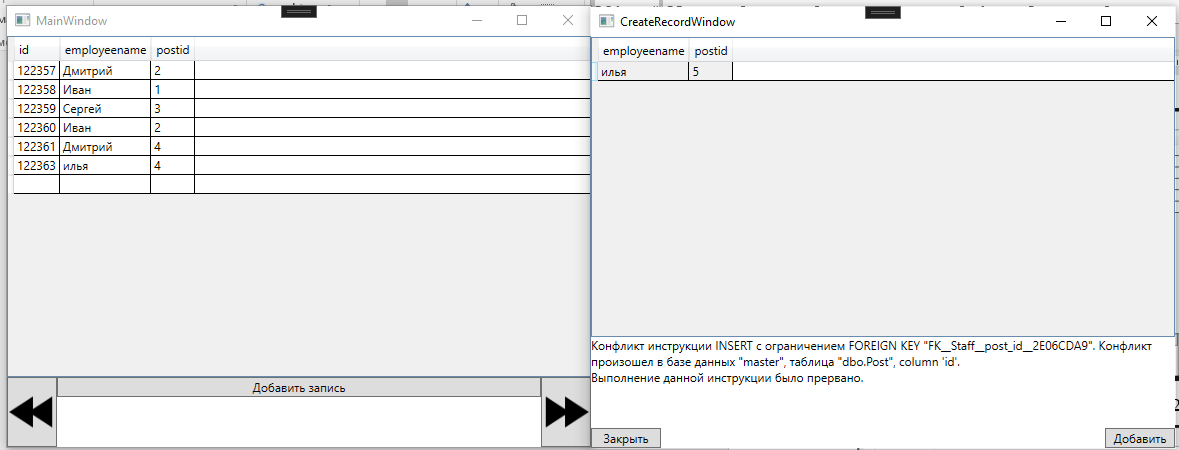


Рис. 26. Результаты

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.

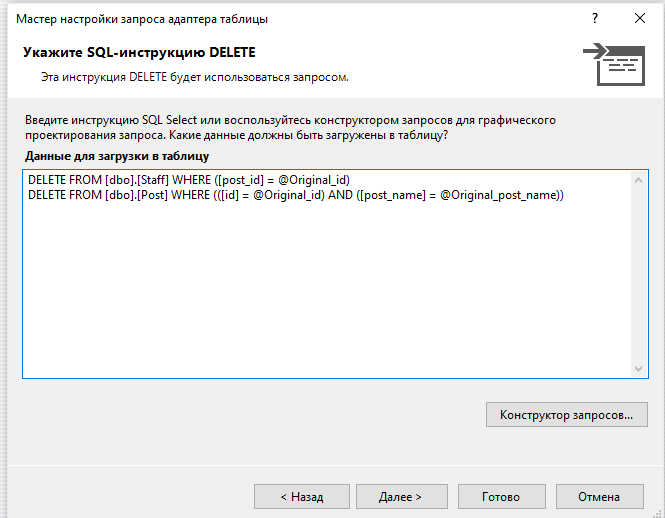


Рис. 27. Удаление записи из таблицы

Далее добавим новый запрос на удаление сотрудника из таблицы сотрудников.

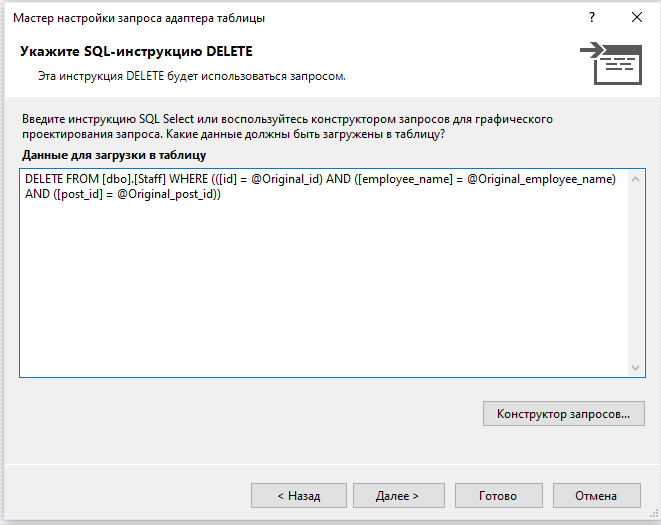


Рис. 28. Запрос на удаление записи

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

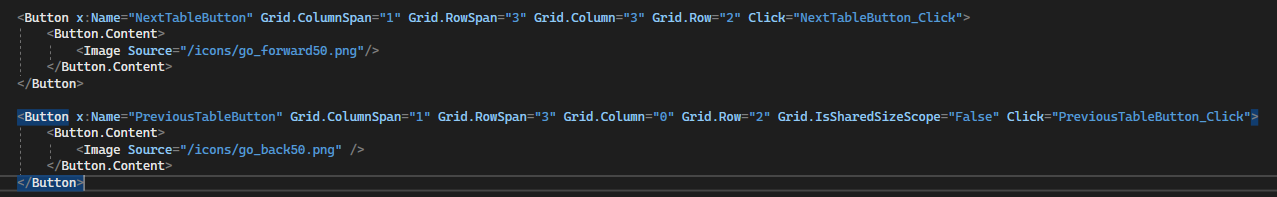


Рис. 30. Настройка кнопок

Изменим грид.

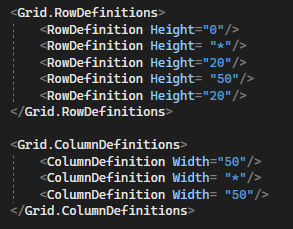


Рис. 31. Изменение грида

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

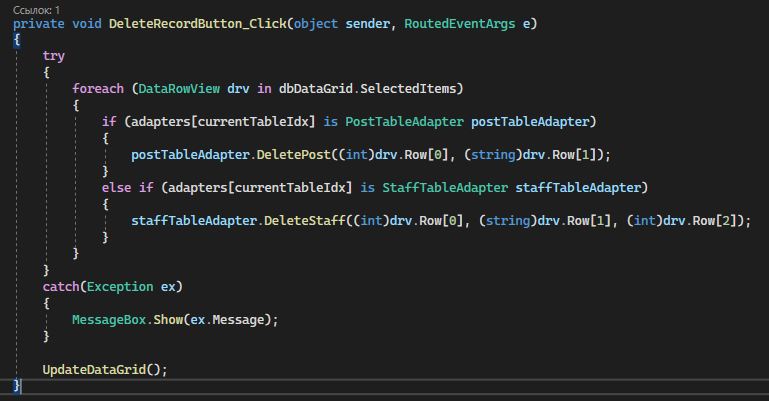


Рис. 32. Логика удаления записи в таблице

Результаты

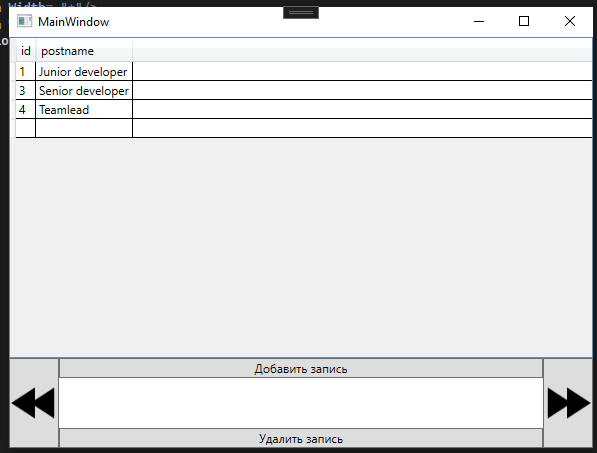


Рис. 33. Результаты

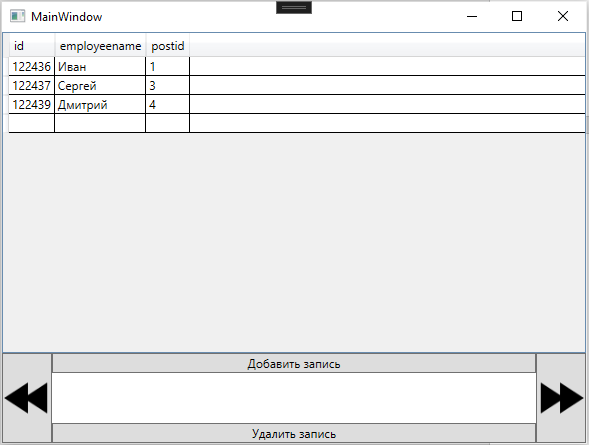


Рис. 34. Результаты

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.

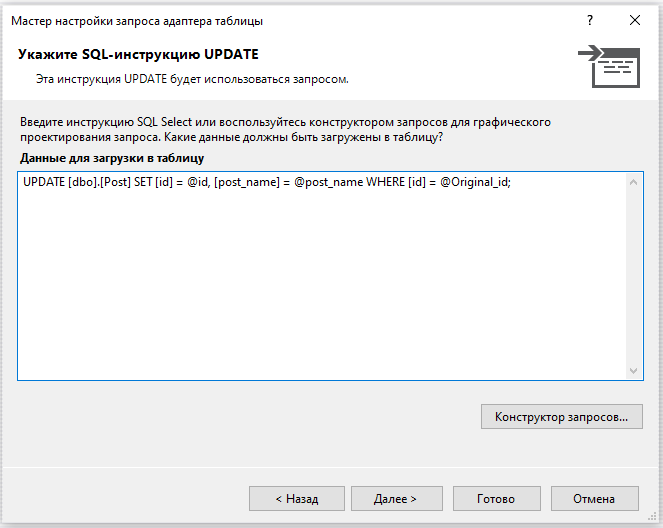


Рис. 35. SQL запрос на обновление должности по ID

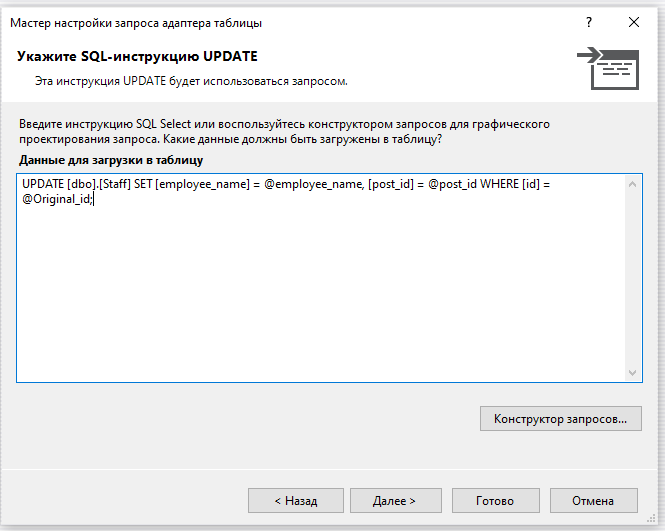


Рис. 36. SQL запрос на обновление сотрудника

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.

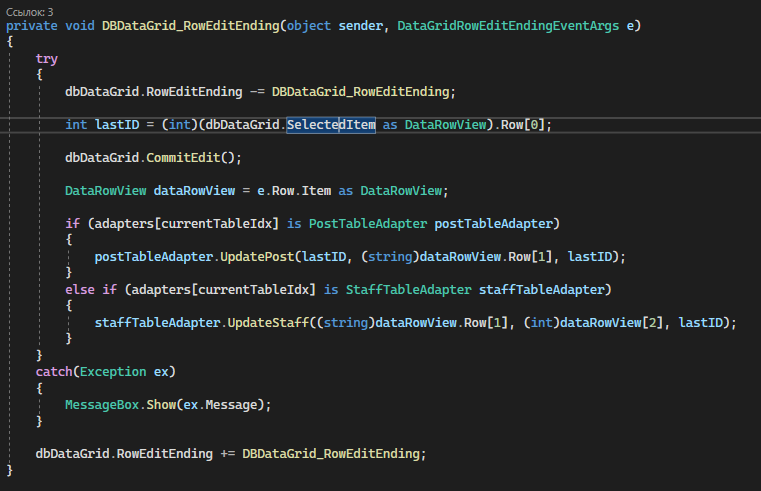


Рис. 37. Обработчик

1. Результаты:

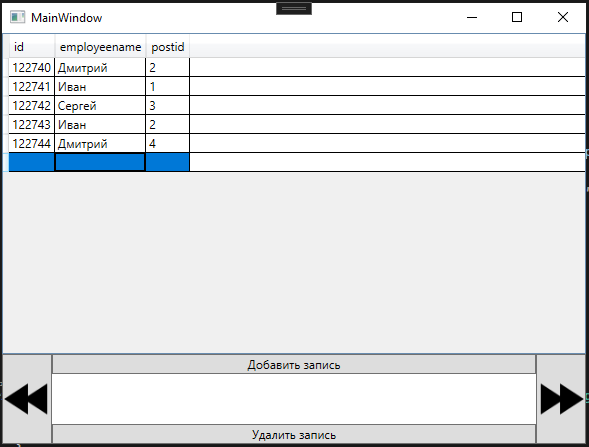


Рис. 38. До изменения строчки с ID 122744

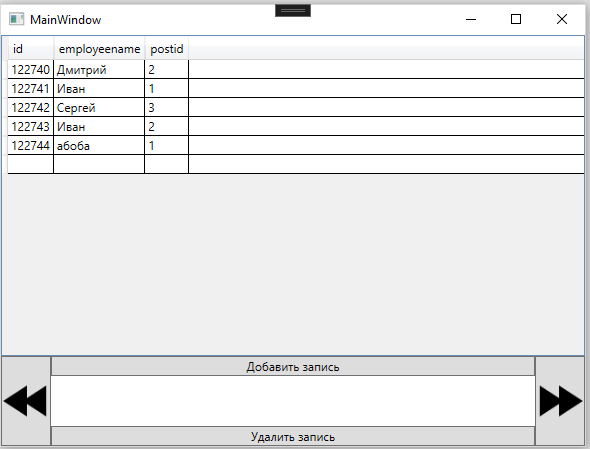


Рис. 39. После изменения строчки с ID 122744

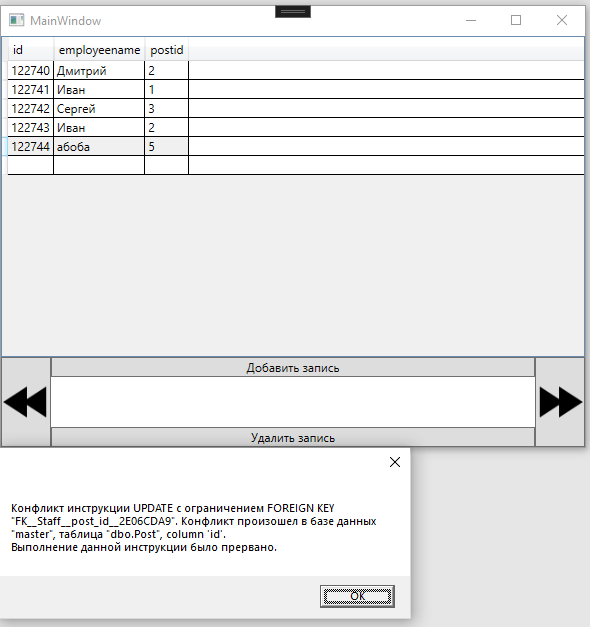


Рис. 40. Обработанная ошибка

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