Лабораторная работа №18
Разработка приложения для редактирования данных

1 Цель работы

1.1 Научиться выполнять вставку, обновление и удаление записей средствами EF Core;

1.2 Научиться обеспечивать обратную связь при редактировании данных.

2 Литература

2.1 https://metanit.com/sharp/efcore/1.4.php Основные операции с данными. CRUD

3 Подготовка к работе

3.1 Повторить теоретический материал (см.п.2).

3.2 Изучить описание лабораторной работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

Для работы с фильмами создать классы контекста БД, модели данных, сервиса. В окнах применять сервисы для работы с БД. Тип приложения: WPF.

5.1 Удаление данных из таблицы БД.

5.1.1 Реализовать отображение в главном окне списка фильмов в табличном виде (элемент должен быть доступен только на чтение).

5.1.2 Добавить кнопки «Добавить» и «Удалить».

5.1.3 При нажатии на «Удалить» должно выполняться удаление из таблицы БД выбранной записи. Для получения из DataGrid выбранных записей использовать SelectedItems. После удаления обновлять источник данных для DataGrid.

5.1.4 Реализовать перехват исключений.

5.2 Обратная связь с пользователем при удалении записей.

5.2.1 Перед удалением запрашивать у пользователя подтверждение удаления выбранных строк, используя MessageBox с иконкой вопроса, кнопками «Да»/«Нет», заголовком «Удаление» и следующим текстом вопроса (вместо *N* должно отображаться количество выбранных записей):

Вы уверены, что хотите удалить *N* записей?

5.2.2 В случае успешного удаления записей информировать об этом пользователя, используя MessageBox с иконкой информации, кнопкой «ОК», заголовком «Информация» и следующим текстом:

Данные успешно удалены.

5.2.3 В случае возникновения исключения при удалении записи информировать об этом пользователя, используя MessageBox с иконкой ошибки, кнопкой «ОК», заголовком «Ошибка» и следующим текстом (вместо *описание* должно отображаться сообщение исключения):

Не удалось удалить записи. Причина: *описание*.

5.3 Вставка записи в таблицу БД.

5.3.1 Добавить в приложение окно работы с фильмом и реализовать переход к ней с главной формы при нажатии на кнопку «Добавить».

Добавить на форму элементы управления для ввода данных:

- поля ввода для указания названия, длительности, года выхода фильма,
- выпадающий список для указания возрастного рейтинга,
- многострочное поле ввода для указания описания.

5.3.2 При загрузке окна реализовать заполнение выпадающего списка данными из БД (получить из таблицы Фильм как неповторяющиеся значения и добавить отдельно еще 2 возрастных рейтинга).

5.3.3 Выполнить привязку данных в окне к данным закрытого поля *фильм* и указать поле как DataContext.

5.3.4 При нажатии на кнопку «Сохранить» требуется сохранить данные в БД. У текстовых данных требуется обрезать пробелы по краям строки.

5.3.5 Реализовать перехват исключений. После сохранения закрывать окно и обновлять источник данных для DataGrid.

5.4 Редактирование записи в таблице БД.

5.4.1 Изменить в форме работы с фильмом конструктор, добавив в него параметр типа Фильм (по умолчанию — null). Присвоить полю *фильм* переданный в конструкторе параметр.

5.4.2 На главной форме создать у DataGrid обработчик MouseDoubleClick: Полученное значение передать в конструкторе форме работы с фильмом. Пример получения выбранного элемента:

var объект = элементОтображенияТаблицы.SelectedItem as Тип;

5.4.3 Внести изменения в метод сохранения: если идентификатор *фильма* равен 0, вызывать метод вставки, иначе – метод добавления записи.

5.5 Обратная связь с пользователем при добавлении/изменении записи.

5.5.1 В случае успешного добавления записи информировать об этом пользователя, используя MessageBox с иконкой информации, кнопкой «ОК», заголовком «Информация» и следующим текстом:

Данные успешно сохранены.

5.5.2 Проверить на корректность данные, указанные пользователем (заполненность обязательных полей ввода, допустимость числовых значений, выбранные в выпадающих списках значения.

Каждое некорректное значение добавлять в строку со списком ошибок.

Строку с ошибками формировать, используя StringBuilder, например:

```
StringBuilder errors = new();
if (book.title == null)
   errors.AppdendLine("название обязательно должно быть указано");
```

Все ошибки вывести пользователю, используя MessageBox с иконкой ошибки, кнопкой «ОК», заголовком «Ошибка» и текстом обнаруженных ошибок заполнения.

5.5.3 В случае возникновения исключения при сохранении записи информировать об этом пользователя, используя MessageBox с иконкой ошибки, кнопкой «ОК»,

заголовком «Ошибка» и следующим текстом (вместо *описание* должно отображаться сообщение исключения):

Не удалось сохранить запись. Причина: *описание*.

6 Порядок выполнения работы

6.1 Выполнить все задания из п.5.в одном проекте LabWork18.
6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист
7.2 Цель работы
7.3 Ответы на контрольные вопросы
7.4 Вывод

8 Контрольные вопросы

8.1 Для чего используются методы Add() и AddRange() в EF Core?
8.2 Для чего используются методы Update() в EF Core?
8.3 Для чего используются методы Remove() и RemoveRange() в EF Core?
8.4 Как сохранить изменения в БД, используя EF Core?
8.5 Как изменить значения полей объекта?
8.6 Какое значение по умолчанию присваивается идентификатору нового объекта?
8.7 Как передать объект с одной формы на другую?