

Лабораторная работа №17

Разработка приложения для фильтрации, поиска и сортировки данных

1 Цель работы

1.1 Научиться создавать приложение для доступа к БД средствами EF Core.

1.2 Научиться выполнять сортировку, фильтрацию и分页 вывод данных, используя LINQ-запросы.

2 Литература

2.1 <https://metanit.com/sharp/efcore/> – гл.1,6

3 Задание

3.1 Создание библиотеки для работы с БД

3.1.1 Создать проект DatabaseLibrary типа Библиотека классов.

3.1.2 Используя Scaffold-DbContext, добавить в нее контекст БД и классы модели данных таблиц Посетитель и Билет, Жанр и Фильм. Все классы структурировать (создать директории для контекста, и моделей).

3.1.3 Добавить в проект директорию для сервисов, в которой создать классы сервисов для каждой модели. В каждый сервис добавить:

- закрытое поле на чтение для хранения контекста БД,
- асинхронные методы получения значения по идентификатору и набора всех значений.

3.1.4 Отключить отслеживание изменений и включить загрузку связанных данных через навигационные свойства.

3.1.5 Реализовать проверку работы методов в консольном приложении или оконном приложении, создав объекты сервиса и вызвав методы получения всех данных.

В приложении WPF для указания источника данных можно использовать следующий код: элементОтображенияСписков.ItemsSource = НаборДанных;

3.2 Пагинация данных

3.2.1 Добавить класс пагинации с автосвойствами для хранения размера страницы (по умолчанию – 5) и номера страницы (по умолчанию – 1).

3.2.2 Добавить параметр для пагинации в методы вывода всех данных и применить его.

3.2.3 Реализовать проверку работы методов в консольном приложении или оконном приложении.

3.3 Сортировка данных

3.3.1 Добавить класс сортировки с автосвойствами для хранения названия столбца и указания, требуется ли фильтровать по возрастанию (значение по умолчанию – true).

3.2.2 Добавить параметр для сортировки в методы вывода всех данных и применить его.

3.2.3 Реализовать проверку работы методов в консольном приложении или оконном приложении.

3.4 Фильтрация данных

3.4.1 Добавить класс фильтрации данных фильма с автосвойствами:

- название,
- часть названия,
- год выхода минимальный,
- год выхода максимальный.

Все параметры должны допускать null.

3.4.2 Добавить параметр для сортировки в метод вывода всех данных о фильме и применить его. Для удобства можно создать отдельный метод, в котором выполнять фильтрацию переданных данных и возвращать отфильтрованный результат.

3.4.3 Реализовать проверку работы метода в консольном приложении или оконном приложении с разными значениями фильтров.

3.5 Проекция данных

3.5.1 Добавить в библиотеку директорию DTOs, в которой создать DTO классы для представления информации:

- о фильме (идентификатор, название, длительность),
- о билете (номер билета, код сеанса, ряд, место, номер телефона посетителя),
- о пользователе (номер телефона, количество купленных билетов),
- о жанрах фильма (идентификатор фильма, название фильма, список жанров).

3.5.2 Добавить в соответствующие сервисы методы для получения данных в формате DTO:

- в сервис фильмов: о фильме и жанрах фильма,
- в сервис пользователя: о пользователе,
- в сервис билета: о билетах.

3.5.3 Реализовать проверку работы метода в консольном приложении или оконном приложении.

4 Порядок выполнения работы

4.1 Выполнить все задания из п.3 в приложении на C#.

4.2 Ответить на контрольные вопросы.

5 Содержание отчета

5.1 Титульный лист

5.2 Цель работы

5.3 Ответы на контрольные вопросы

5.4 Вывод

6 Контрольные вопросы

6.1 Для чего используются метод `OrderBy`, `OrderByDescending`, `ThenBy`, `ThenByDescending` и в чем их отличие?

6.2 Для чего используются методы `Take` и `Skip` и как они применяются при пагинации?

6.3 Для чего используется метод `Where`?

6.4 Какие логические операторы могут применяться при составлении условий?

6.5 Как проверить, что значение есть в списке?

6.6 Как проверить, что строка начинается с определенного текста?

6.7 Как проверить, что строка содержит определенный текст?

6.8 Как указать в `Select` список требуемых данных?