**АРХАНГЕЛЬСКИЙ КОЛЛЕДЖ ТЕЛЕКОММУНИКАЦИЙ**

**ИМ. Б.Л. РОЗИНГА (ФИЛИАЛ) СПбГУТ**

**(АКТ (ф) СПбГУТ)**

**Отчеты по лабораторным и практическим работам**

**МДК 11.01**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Студент | ИСПП-35 | |  |  | *Пожидаев Г.С.* |
|  | (Группа) | | (Подпись) | (Дата) | (И.О. Фамилия) |
| Преподаватель | | |  |  | *Маломан Ю. С.* |
|  | |  | (Подпись) | (Дата) | (И.О. Фамилия) |

Архангельск 2024

# **Лабораторная работа №17**

**Разработка приложения для фильтрации, поиска и сортировки данных**

1. **Цель работы** 
   1. Научиться создавать приложение для доступа к БД средствами EF Core.
   2. Научиться выполнять сортировку, фильтрацию и постраничный вывод данных, используя LINQ-запросы.
2. **Контрольные вопросы**

Вопрос: Для чего используются метод OrderBy, OrderByDescending, ThenBy, ThenByDescending и в чем их отличие?

Методы OrderBy, OrderByDescending, ThenBy, ThenByDescending используются для сортировки последовательностей элементов [1].

Основные отличия:

1. OrderBy и OrderByDescending:

- Сортируют по умолчанию по возрастанию (OrderBy) или убыванию (OrderByDescending).

- Если элементы равны, порядок не гарантируется.

2. ThenBy и ThenByDescending:

- Позволяют добавить дополнительные критерии сортировки после основного.

- Работают только если перед ними был вызван OrderBy или OrderByDescending.

Пример использования:

// Сортировка по возрастанию имени, затем по возрастанию возраста

var sortedUsers = users.OrderBy(u => u.Name).ThenBy(u => u.Age);

// Сортировка по убыванию возраста, затем по возрастанию имени

var sortedUsersDesc = users.OrderByDescending(u => u.Age).ThenBy(u => u.Name);

Вопрос: Для чего используются методы Take и Skip и как они применяются при пагинации?

Методы Take и Skip используются для ограничения количества элементов в результате запроса [1].

- Take(n): возвращает первые n элементов последовательности.

- Skip(n): пропускает первые n элементов последовательности.

Применение при пагинации:

int pageSize = 10;

int pageNumber = 2;

var paginatedItems = items.Skip((pageNumber - 1) \* pageSize).Take(pageSize);

Это позволяет реализовать страницную навигацию по большим наборам данных.

Вопрос: Для чего используется метод Where?

Метод Where используется для фильтрации элементов последовательности на основе заданных условий [1].

Пример:

var filteredUsers = users.Where(u => u.Age > 18 && u.Country == "USA");

Вопрос: Какие логические операторы могут применяться при составлении условий?

Основные логические операторы в LINQ:

- AND (&&): объединяет два условия

- OR (||): объединяет два условия

- NOT (!): инверсирует условие

Пример:

var result = users.Where(u => u.Age > 18 || (u.Age >= 65 && u.Country == "USA"));

Вопрос: Как проверить, что значение есть в списке?

Для проверки наличия значения в списке можно использовать метод Contains:

bool containsValue = list.Contains(valueToCheck);

Вопрос: Как проверить, что строка начинается с определенного текста?

Для проверки начала строки можно использовать метод StartsWith:

bool startsWithSpecificText = string.StartsWith(specificText);

Вопрос: Как проверить, что строка содержит определенный текст?

Для проверки наличия подстроки в строке используется метод Contains:

bool containsSpecificText = string.Contains(specificText);

Вопрос: Как указать в Select список требуемых данных?

Метод Select используется для проекции (выбора) нужных данных из исходной последовательности [1]:

var selectedData = source.Select(item => new { item.Id, item.Name, item.Price });

Этот метод позволяет выбрать только необходимые поля или вычислить новые значения на основе существующих.

LINQ предоставляет мощные инструменты для работы со структурированными данными, позволяя легко фильтровать, сортировать, агрегировать и трансформировать данные в цепочках запросов.

1. **Вывод**
   1. В ходе лабораторной работы мы научились создавать приложение для доступа к БД средствами EF Core.
   2. Научились выполнять сортировку, фильтрацию и постраничный вывод данных, используя LINQ-запросы.