Linq операторы

otus.ru

Проверить, идет ли запись

Напишите «+» в чат, если меня слышно и видно

Тема урока

Linq-операторы

Нилов Павел

Fullstack разработчик компании Волховец

Преподаватель курса C# Professional, C# Basic

Контакты:t.me/@NilovPavel

Правила вебинара

Активно
участвуем

Задаем вопрос
в чат

Вопросы вижу в чате,
могу ответить не сразу

Маршрут вебинара

1. Фильтрация
2. Проекция
3. Операции с наборами
4. Сортировка данных
5. Квантификаторы
6. Секционирование данных
7. Преобразование(конвертация) данных
8. Операции соединения
9. Группировка элементов
10. Ответы на вопросы

Цели вебинара

1. Узнать про все операции, доступные в linq-запросах
2. Узнать какие операторы используются в операциях из п.1
3. Практика в навыках описания запросов

Операции linq

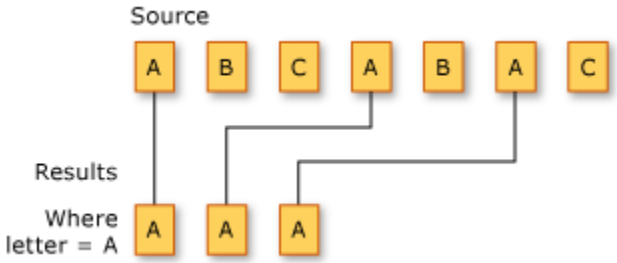
Операции Linq

1. Фильтрация
2. Проекция
3. Наборы данных
4. Сортировка
5. Квантификаторы
6. Секционирование
7. Конвертация данных
8. Операции соединения
9. Группировка данных

Фильтрация

Фильтрация

Фильтрация – это операция, результатом которой будет набор значений, подходящий под определенное условие.



Фильтрация

Фильтрация данных в linq представлена следующими методами:

OfType – фильтрует данные по типу;

Where – фильтрует значения по условию (в декларативном синтаксисе - where).

Проекция

Проекция

Проекция – это операция преобразования объекта в новую форму, которая часто состоит только из этих свойств, которые впоследствии используются.

Проекция

Проекция данных в linq представлена следующими методами:

Select – проецирует значения, которые основаны на функции преобразования (в декларативном синтаксисе также select)

SelectMany – проецирует последовательности значений, основанных на функции преобразования, а затем выравнивает их в одну последовательность.(в декларативном синтаксисе множественный from). Простыми словами: забирает последовательность из элемента коллекции и кладет его в результирующую последовательность.

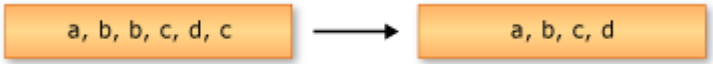
Zip – создает последовательность кортежей из 2-3 указанных последовательностей.

Операции над множествами

Операции над множествами

Под операции над множествами в данном случае понимаются операции запросов, которые создают результирующий набор присутствия или отсутствия эквивалентных элементов в одной или отдельной коллекциях.

Distinct или DistinctBy – возвращает уникальные элементы последовательности. Можно также сказать, что удаляет дубликаты.

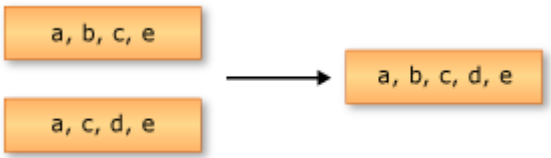


Except или ExceptBy – возвращает набор значений, которые присутствуют в одной коллекции и отсутствуют в другой.

Операции над множествами

Intersect или IntersectBy – возвращает набор значений, которые встречаются в обоих коллекциях.

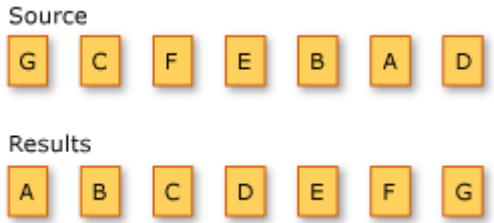
Union или UnionBy – возвращает набор уникальных значений, присутствующий в обоих коллекциях.



Сортировка данных

Сортировка данных

Операция сортировки упорядочивает элементы последовательности на основе одного или нескольких атрибутов.



OrderBy – сортировка значений в возрастающем порядке. В декларативном синтаксисе – оператор orderby или orderby ascending.

OrderByDescending – сортировка значений в убывающем порядке. В декларативном синтаксисе – оператор orderby descending.

Сортировка данных

ThenBy – дополнительная сортировка по возрастанию. В декларативном дополнительные – операторы orderby или orderby ascending.

ThenByDescending – дополнительная сортировка по убыванию. В декларативном синтаксисе дополнительные orderby descending.

Reverse - изменение порядка элементов в коллекции на обратный. В декларативном синтаксисе аналогов нет.

Квантификаторы

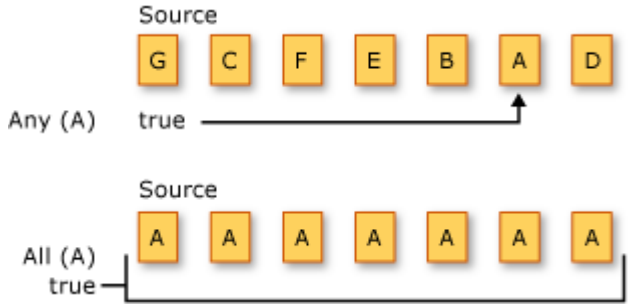
Операции квантификатора

Квантификатор – это операция, которая возвращают значение bool, которое указывает, удовлетворяют ли условию некоторые или все элементы в последовательности.

All - определяет, все ли элементы последовательности удовлетворяют условию.

Any - определяет, удовлетворяют ли условию какие-либо элементы последовательности.

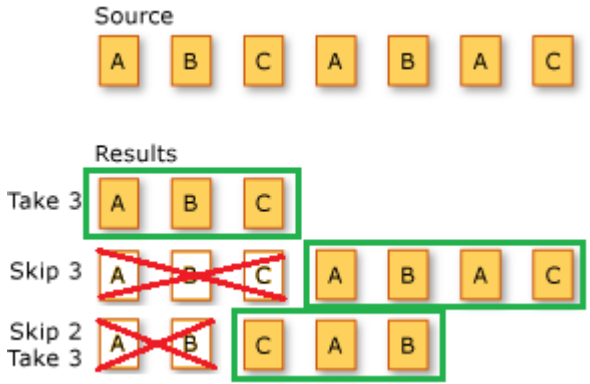
Contains - определяет, содержит ли последовательность указанный элемент.



Секционирование данных

Секционирование данных

Секционирование – это операция разделения входной последовательности на два раздела без изменения порядка элементов, а затем возвращения одного из разделов.



Секционирование данных

Skip - пропускает элементы до указанной позиции в последовательности.

SkipWhile - пропускает элементы на основе функции предиката, пока элемент не удовлетворяет условию.

Take - возвращает элементы на указанную позицию в последовательности.

TakeWhile - принимает элементы на основе функции предиката, пока элемент не удовлетворяет условию.

Chunk - разделяет элементы последовательности на фрагменты указанного максимального размера.

Преобразование типов данных

Преобразование типов данных

Операция преобразования меняет тип входных объектов.

Таблица методов преобразования находится здесь.

https://learn.microsoft.com/en-us/dotnet/csharp/linq/standard-query-operators/converting-data-types#methods

Операции соединения

Операции соединения(Join)

Соединение двух источников данных — это связь объектов в одном источнике данных с объектами, которые имеют общий атрибут в другом источнике данных.

Join – соединяет две последовательности на основании функций селектора ключа и извлекает пары значений. В декларативном синтаксисе дополнительные join … in … on … equals.

GroupJoin – соединяет две последовательности на основании функций селектора ключа и группирует полученные при сопоставлении данные для каждого элемента. В декларативном синтаксисе дополнительные join … in … on … equals … into … .

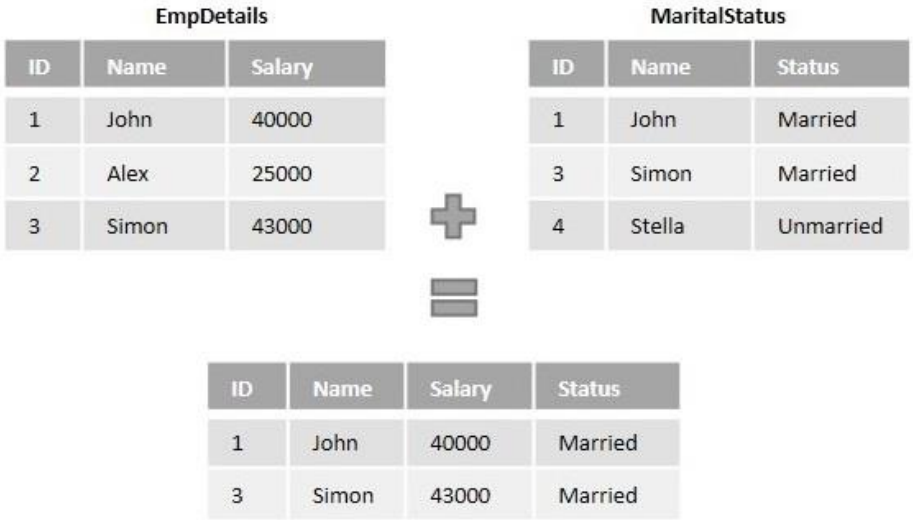
Пример декларативного синтаксиса:

```
from x in set1
join y in set2 on y.Prop2 equals x.Prop1
```

где, свойства y.Prop2 и x.Prop1 являются одной и той же сущностью, скажем идентификатором группы студента.

Операции соединения(Join)

Пример:



EmpDetails

ID	Name	Salary
1	John	40000
2	Alex	25000
3	Simon	43000

MaritalStatus

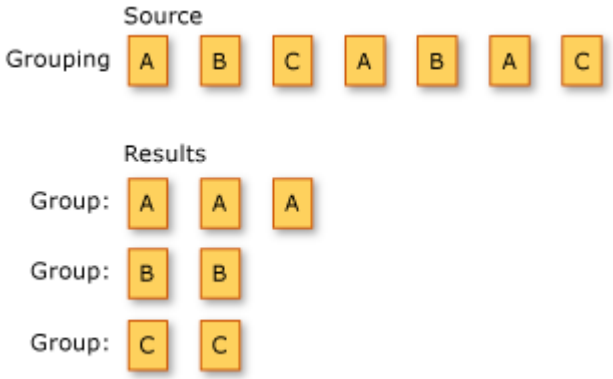
ID	Name	Status
1	John	Married
3	Simon	Married
4	Stella	Unmarried

ID	Name	Salary	Status
1	John	40000	Married
3	Simon	43000	Married

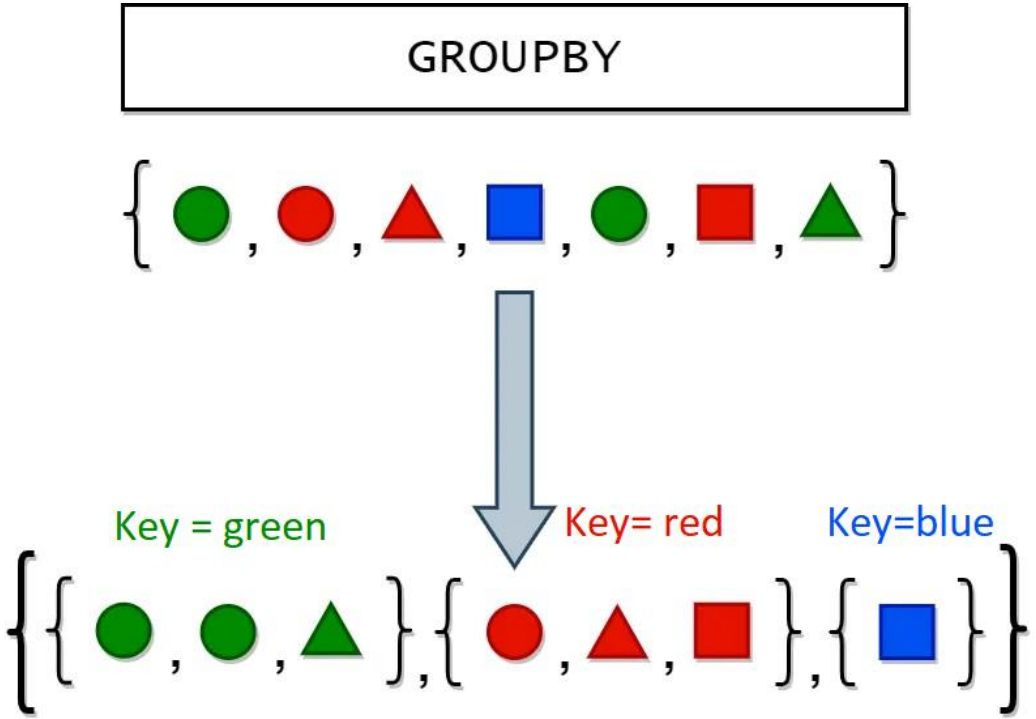
Группировка данных

Группирование данных

Группировка – это операция объединения данных в группы таким образом, чтобы у элементов в каждой группе был общий атрибут. На следующем рисунке показаны результаты операции группирования последовательности символов. Ключ для каждой группы — это символ.



Группирование данных



Группирование данных

GroupBy – группирует элементы с общим атрибутом. Объект представляет каждую IGrouping<TKey,TElement> группу. В декларативном синтаксисе group … by или group … by … into

ToLookup – вставляет элементы в Lookup<TKey,TElement> (словарь "один ко многим") в зависимости от функции выбора ключа.

Описание всех методов

Документация

Подробное описание всех операторов linq Вы сможете найти: тут

Ответы на вопросы

Рефлексия

Цели вебинара

1. Узнать про все операции, доступные в linq-запросах
2. Узнать какие операторы используются в операциях из п.1
3. Практика в навыках описания запросов

Вопросы?

Задаем
вопросы в чат

Ставим “–”,
если вопросов нет