Операции First, FirstOrDefault, Last и LastOrDefault

LINQ (../../base/level1/info_linq.php) --- LINQ to Objects (../level1/linq_index.php) --- Операции First, FirstOrDefault, Last и LastOrDefault

Операции элементов позволяют извлекать отдельные элементы из входной последовательности.

First

Операция First возвращает первый элемент последовательности или первый элемент последовательности, соответствующий предикату — в зависимости от использованного прототипа.

Эта операция имеет два прототипа, описанные ниже:

Первый прототип First

C#

```
public static T First<T>(
        this IEnumerable<T> source);
```

При использовании этого прототипа операции First выполняется перечисление входной последовательности source и возвращается ее первый элемент.

Второй прототип First

Второй прототип операции First позволяет передать ему предикат:

C#

```
public static T First<T>(
    this IEnumerable<T> source,
    Func<T, bool> predicate);
```

Эта версия операции First возвращает первый найденный ею элемент, для которого predicate дал true. Если ни один из элементов не заставил predicate вернуть true, то операция First генерирует исключение InvalidOperationException.

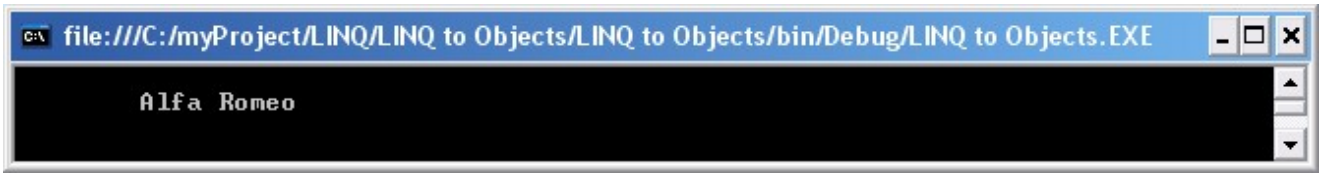
Пример первого прототипа First приведен ниже:

C#

```
string[] cars = { "Alfa Romeo", "Aston Martin", "Audi", "Nissan", "Chevrolet",  "Chrysler",
"Dodge", "BMW",
                            "Ferrari",  "Bentley", "Ford", "Lexus", "Mercedes", "Toyota",
"Volvo", "Subaru", "Жигули :)"};

string auto = cars.First();
Console.WriteLine("\n\t" + auto);
```

Вот результат:



Может возникнуть вопрос: чем отличается эта операция от вызова операции Take (../level2/2_3.php) с параметром 1? Отличие в том, что Take возвращает последовательность элементов, даже если она состоит всего из одного элемента. Операция First всегда возвращает в точности один элемент либо генерирует исключение, если возвращать нечего.

Ниже приведен пример использования второго прототипа операции First:

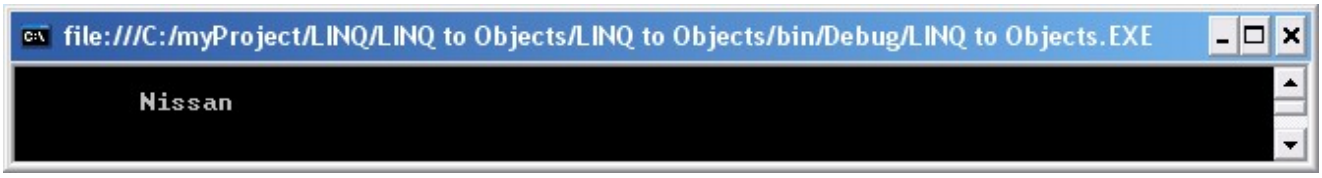
C#

```
string[] cars = { "Alfa Romeo", "Aston Martin", "Audi", "Nissan", "Chevrolet",  "Chrysler",
"Dodge", "BMW",
                            "Ferrari",  "Bentley", "Ford", "Lexus", "Mercedes", "Toyota",
"Volvo", "Subaru", "Жигули :)"};

string auto = cars.First(p => p.StartsWith("N"));
Console.WriteLine("\n\t" + auto);
```

Пройди тесты

Он должен вернуть первый элемент из входной последовательности, начинающийся со строки "N". Вот результат:



Вспомните, что если любой из прототипов операции First не находит элемент, который нужно возвратить, генерируется исключение InvalidOperationException. Чтобы избежать этого, используйте операцию FirstOrDefault.

FirstOrDefault

Операция FirstOrDefault подобна First во всем, кроме поведения, когда элемент не найден. Эта операция имеет два прототипа, описанные ниже:

Первый прототип FirstOrDefault

C#

```
public static T FirstOrDefault<T> (
        this IEnumerable<T> source);
```

Эта версия прототипа FirstOrDefault возвращает первый элемент, найденный во входной последовательности. Если последовательность пуста, возвращается default(T). Для ссылочных и допускающих null типов значением по умолчанию является null.

Второй прототип FirstOrDefault

Второй прототип операции FirstOrDefault позволяет передать predicate, определяющий, какой элемент следует возвратить.

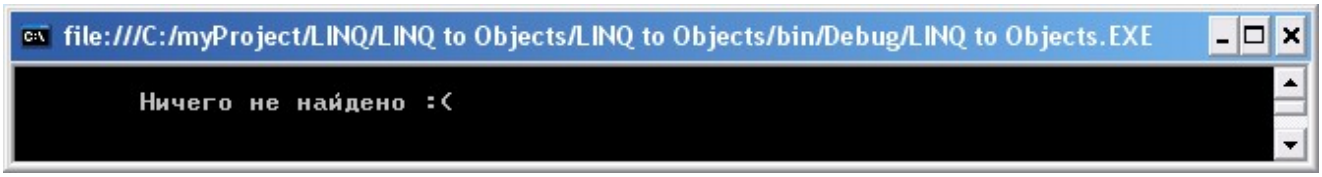
C#

```
public static T FirstOrDefault<T>(
       this IEnumerable<T> source,
       Func<T, bool> predicate);
```

Код ниже, содержит пример использования первого прототипа FirstOrDefault, где элемент не найден. Для этого понадобилась пустая последовательность, которая была создана с помощью Take(0):

```
string[] cars = { "Alfa Romeo", "Aston Martin", "Audi", "Nissan", "Chevrolet",  "Chrysler",
"Dodge", "BMW",
                            "Ferrari",  "Bentley", "Ford", "Lexus", "Mercedes", "Toyota",
"Volvo", "Subaru", "Жигули :)"};

string auto = cars.Take(0).FirstOrDefault();
Console.WriteLine(auto == null ? "\n\tНичего не найдено :(" : auto);
```



Last

Операция Last возвращает последний элемент последовательности или последний элемент, соответствующий предикату — в зависимости от используемого прототипа. Эта операция имеет два прототипа, которые описаны ниже:

Первый прототип Last

C#

```
public static T Last<T>(
       this IEnumerable<T> source);
```

В случае этого прототипа операция Last перечисляет входную последовательность по имени source и возвращает ее последний элемент.

Второй прототип Last

Второй прототип Last позволяет передать predicate и рассматривается ниже:

C#

```
public static T Last<T>(
      this IEnumerable<T> source,
      Func<T, bool> predicate) ;
```

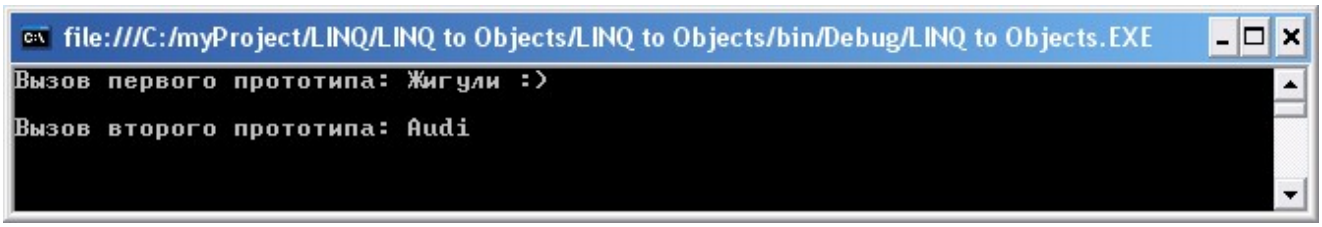

Эта версия операции Last возвращает последний из найденных элементов, для которых predicate вернет true. Ниже показан пример использования этих прототипов:

C#

```
string[] cars = { "Alfa Romeo", "Aston Martin", "Audi", "Nissan", "Chevrolet",  "Chrysler",
"Dodge", "BMW",
                            "Ferrari",  "Bentley", "Ford", "Lexus", "Mercedes", "Toyota",
"Volvo", "Subaru", "Жигули :)"};

string auto = cars.Last();
Console.WriteLine("Вызов первого прототипа: " + auto);

string auto1 = cars.Last(p => p.StartsWith("A"));
Console.WriteLine("\nВызов второго прототипа: " + auto1);
```



Помните, что если любому из двух прототипов операции Last возвращать нечего, генерируется исключение InvalidOperationException. Чтобы избежать этого, используйте операцию LastOrDefault.

LastOrDefault

Операция LastOrDefault подобна Last во всем, за исключением поведения в случае, когда элемент не найден. Эта операция имеет два прототипа, анологичные операции Last.

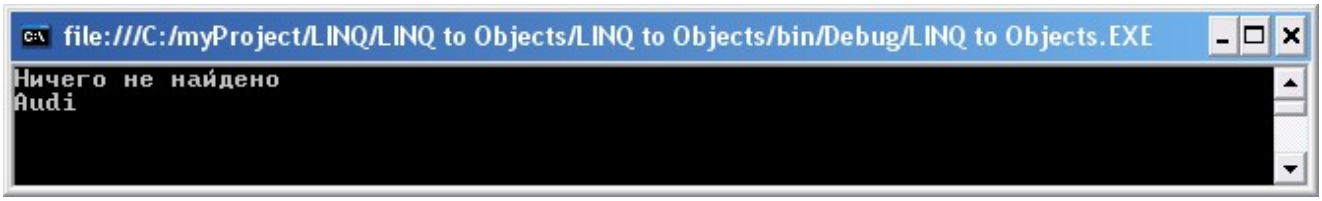
Ниже показан пример использования обоих прототипов LastOrDefault:

C#

```
string[] cars = { "Alfa Romeo", "Aston Martin", "Audi", "Nissan", "Chevrolet",  "Chrysler",
"Dodge", "BMW",
                            "Ferrari",  "Bentley", "Ford", "Lexus", "Mercedes", "Toyota",
"Volvo", "Subaru", "Жигули :)"};

            string auto = cars.Take(0).LastOrDefault();
            Console.WriteLine(auto == null ? "Ничего не найдено" : auto);

            string auto1 = cars.LastOrDefault(p => p.StartsWith("A"));
            Console.WriteLine(auto1 == null ? "Ничего не найдено" : auto1);
```



Назад (3_4.php)	4	5	6	Вперед (3_6.php)

Пройди тесты

C# тест (легкий) (https://professorweb.ru/test/c-sharp-test.html)

.NET тест (средний) (https://professorweb.ru/test/asp-test.html)