

Операция DefaultIfEmpty

[LINQ \(../base/level1/info_linq.php\)](#) --- [LINQ to Objects \(../level1/linq_index.php\)](#) --- Операция DefaultIfEmpty

Операции элементов позволяют извлекать элементы из входной последовательности.

Операция DefaultIfEmpty возвращает последовательность, содержащую элемент по умолчанию, если входная последовательность пуста. У операции DefaultIfEmpty есть два прототипа, которые описаны ниже:

Первый прототип DefaultIfEmpty

C#

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

Этот прототип операции DefaultIfEmpty возвращает объект, который при перечислении входной последовательности source выдает каждый ее элемент, если только последовательность не окажется пустой — тогда возвращается последовательность из одного элемента default (T). Для ссылочных и допускающих null типов значением по умолчанию является null.

Обратите внимание, что в отличие от всех прочих операций типа элементов, DefaultIfEmpty возвращает последовательность типа IEnumerable<T> вместо самого типа T. Существуют еще дополнительные операции типа элементов, но они не включены в эту статью, поскольку не являются отложенными.

Второй прототип DefaultIfEmpty

Второй прототип позволяет указать значение по умолчанию.

C#

Пройди тесты

```
public static IEnumerable<T> DefaultIfEmpty<T>(
    this IEnumerable<T> source, T defaultValue);
```

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

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

Эта операция полезна для всех других операций, которые генерируют исключения в случае пустой входной последовательности. Вдобавок эта операция удобна в сочетании с операцией GroupJoin для производства *левосторонних внешних соединений* (*left outer joins*).

Ниже показан пример прототипа DefaultIfEmpty с пустой последовательностью. В данном примере не применяется операция DefaultIfEmpty, чтобы посмотреть, что происходит. Производится поиск в массиве Cars названия Porsche, возвращается первый элемент, и если он не будет равен null, выводится сообщение:

C#

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

try
{
    string nissan = cars.Where(n => n.Equals("Porsche")).First();

    if (nissan != null)
        Console.WriteLine("Автомобиль Porsche найден");
    else
        Console.WriteLine("Автомобиль Porsche не найден");
}
catch (Exception ex)
{
    Console.WriteLine(ex.Message);
}
```



В приведенном коде запрос не нашел элемента, эквивалентного "Porsche", поэтому операции First была передана пустая последовательность. Операции First не нравятся пустые последовательности, поэтому она генерирует исключение.

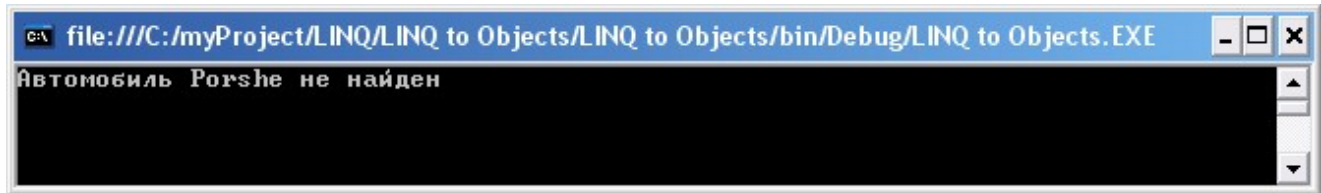
В примере ниже используется тот же самый код, но на этот раз между операциями Where и First вставлена операция DefaultIfEmpty. Таким образом, вместо пустой последовательности операции First будет передана последовательность, содержащая элемент null:
Пройди тесты

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

C# .NET тест (средний) (<https://professorweb.ru/test/asp-test.html>)

```
...
string nissan = cars.Where(n => n.Equals("Porsche")).DefaultIfEmpty().First();
```

Теперь результат выглядит так:



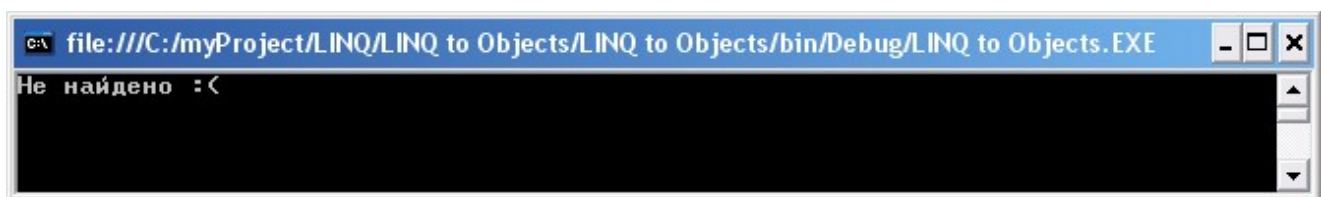
В примере применения второго прототипа DefaultIfEmpty, приведенном ниже, разрешено указывать значение по умолчанию для пустой последовательности:

C#

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

string porsche = cars
    .Where(n => n.Equals("Porsche"))
    .DefaultIfEmpty("Не найдено :(")
    .First();

Console.WriteLine(porsche);
```



Назад (2_10.php)	10	11	12	Вперед (2_12.php)
------------------	----	----	----	-------------------

Пройди тесты



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

Лучший чат для C# программистов (<https://t.me/professorweb>)

ИЕТ тест (средний) (<https://professorweb.ru/test/asp-test.html>)

Наш любимый хостинг (/)

Пройди тесты

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

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