

Операции Range, Repeat и Empty

[LINQ \(../base/level1/info_linq.php\)](#) --- [LINQ to Objects \(../level1/linq_index.php\)](#) --- [Операции Range, Repeat и Empty](#)

Операции генерации помогают в генерации последовательностей.

Range

Операция Range генерирует последовательность целых чисел. У операции Range есть один прототип, описанный ниже:

C#

```
public static IEnumerable<int> Range(  
    int start,  
    int count);
```

Последовательность целых чисел генерируется, начиная со значения, переданного в start, протяженностью до count. Обратите внимание, что это не расширяющий метод, а одна из немногих стандартных операций запросов, которые не расширяют IEnumerable<T>.

Если count меньше 0 или сумма start и count минус 1 больше int.MaxValue, генерируется исключение ArgumentOutOfRangeException. Давайте рассмотрим пример использования операции Range:

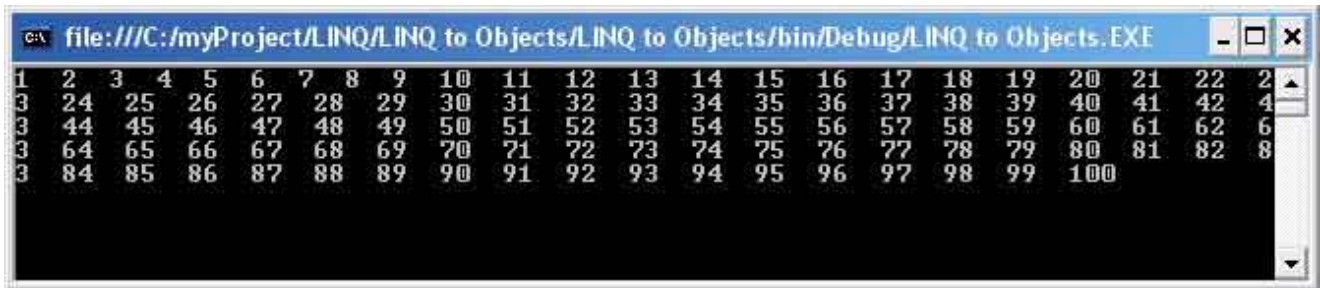
C#

```
IEnumerable<int> nums = Enumerable.Range(1, 100);  
  
foreach (int i in nums)  
    Console.Write(i + " ");
```

Пройди тесты

Опять-таки, следует подчеркнуть, что операция Range не вызывается на последовательности. Это статический метод класса System.Linq.Enumerable. Вывод доказывает отсутствие каких-либо осложнений:

[C# тест \(легкий\) \(https://professorweb.ru/test/c-sharp-test.html\)](#)
[PHP тест \(средний\) \(https://professorweb.ru/test/asp-test.html\)](#)



Repeat

Операция Repeat генерирует последовательность, повторяя указанный элемент заданное количество раз. У операции Repeat есть один прототип, описанный ниже:

C#

```
public static IEnumerable<int> Repeat(
    T element,
    int count);
```

Этот прототип возвращает объект, который при перечислении выдаст count экземпляров элемента T. Обратите внимание, что он является не расширяющим методом, а одной из немногих стандартных операций запросов, которые не расширяют IEnumerable<T>. Если значение аргумента count меньше нуля, генерируется исключение ArgumentOutOfRangeException.

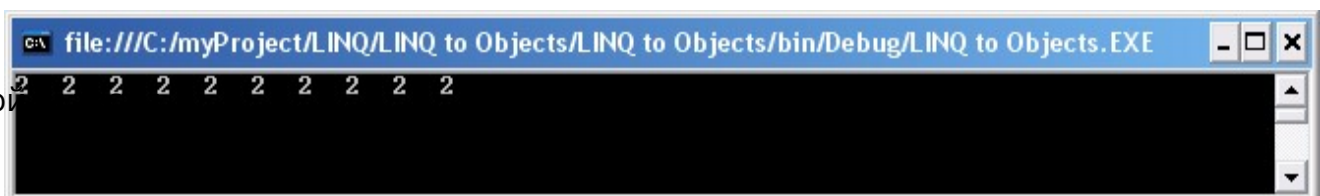
Ниже показан пример, в котором генерируется последовательность из десяти элементов, каждый из которых является числом 2:

C#

```
IEnumerable<int> nums = Enumerable.Repeat(2, 10);

foreach (int i in nums)
    Console.Write(i + " ");
```

Про



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

Empty

Операция Empty генерирует пустую последовательность заданного типа. Операция Empty имеет один прототип, представленный ниже:

C#

```
public static IEnumerable<T> Empty<T>();
```

Этот прототип возвращает объект, при перечислении выдающий последовательность из нуля элементов типа T. Обратите внимание, что это не расширяющий метод, а одна из немногих стандартных операций запросов, которая не расширяет IEnumerable<T>.

Ниже генерируется пустая последовательность типа string с использованием операции Empty и выводится значение Count для сгенерированной последовательности, которое должно быть равно нулю, поскольку последовательность пуста:

C#

```
IEnumerable<string> str = Enumerable.Empty<string>();

foreach (string s in str)
    Console.Write(s);
Console.WriteLine(str.Count());
```

Поскольку последовательность пуста, в цикле foreach отображать нечего, поэтому был предусмотрен вывод количества элементов в последовательности. Данный код выведет значение 0.

Назад (2_11.php)	10	11	12
------------------	----	----	----

Пройди тесты



тест (легкий) (<https://professorweb.ru/test/c-sharp-test.html>)
лучший чит для C# программистов (<https://t.me/professorweb>)

Professor Web (/)
.NET тест (средний) (<https://professorweb.ru/test/asp-test.html>)
Наш любимый хостинг (/)

