Операция Concat

Операция Concat соединяет две входные последовательности и выдает одну выходную последовательность. Операция Concat имеет один прототип, который описан ниже:

C#

```
public static IEnumerable<T> Concat<T>(
     this IEnumerable<T> first,
     IEnumerable<T> second);
```

В этом прототипе две последовательности одного типа T — first и second — являются входными. Возвращается объект, который при перечислении проходит по первой последовательности, выдавая каждый ее элемент в выходную последовательности за которым начинается перечисление второй входной последовательности с выдачей каждого ее элемента в ту же выходную последовательность.

Ниже приведен пример использования операции Concat, а также операций Take и Skip:

C#

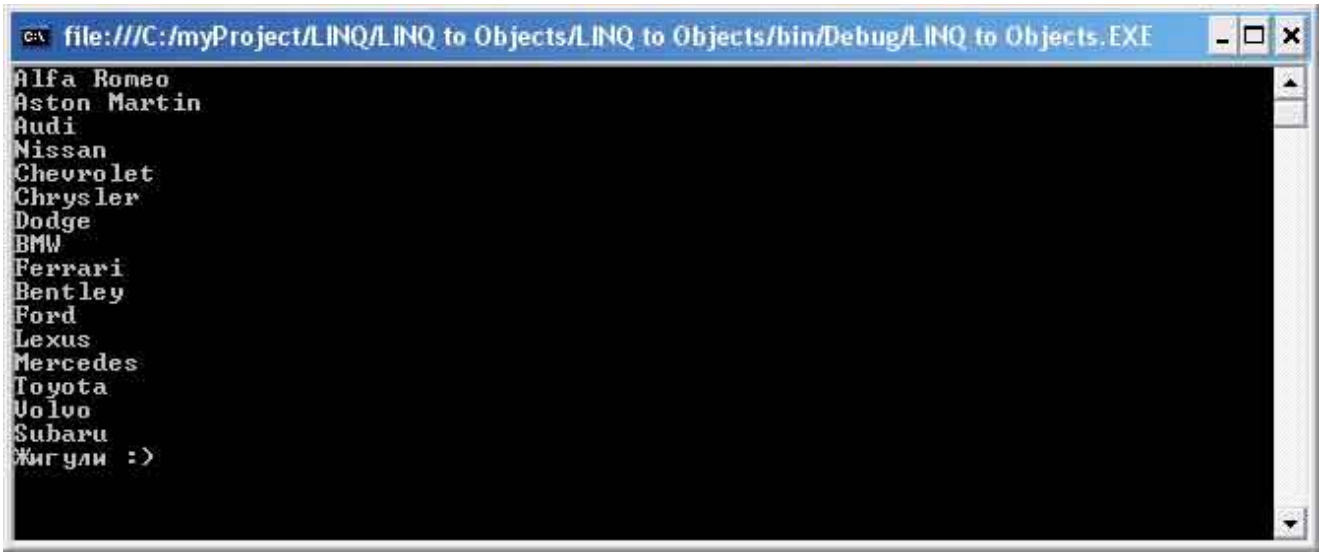
```
string[] cars = { "Alfa Romeo", "Aston Martin", "Audi", "Nissan", "Chevrolet",  "Chrysler",
"Dodge", "BMW",
     "Ferrari",  "Bentley", "Ford", "Lexus", "Mercedes", "Toyota", "Volvo", "Subaru", "Жигу
ли :)"};

IEnumerable<string> auto = cars.Take(5).Concat(cars.Skip(5));

foreach (string str in auto)
         Console.WriteLine(str);
```

Пройди тесты

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

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

Этот код берет пять первых членов из входной последовательности cars и соединяет со всеми, кроме первых пяти входных элементов из последовательности cars. В результате должна получиться последовательность, по содержимому идентичная последовательности cars, следующего вида:



Альтернативный подход к соединению предусматривает вызов операции SelectMany на массиве последовательностей, как показано ниже:

C#

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

IEnumerable<string> auto = new[] {
           cars.Take(5),
           cars.Skip(5)}
           .SelectMany(s => s);
```

В данном примере создается экземпляр массива, состоящего из двух последовательностей: одной, созданной вызовом операции Take на входной последовательности, и другой, созданной вызовом операции Skip на входной последовательности. Обратите внимание, что это подобно предыдущему примеру во всем, за исключением того, что на массиве последовательностей вызывается операция SelectMany. С учетом того, что операция Concat позволяет объединять только две последовательности, при наличии массива последовательностей продемонстрированный прием может оказаться удобнее.

Результат получается тем же, что и при использовании операции Concat.

Назад (2_3.php)	3	4	5	Вперед (2_5.php)

Пройди тесты

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

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