

# Операция Where

[LINQ \(../base/level1/info\\_linq.php\)](#) --- [LINQ to Objects \(../level1/linq\\_index.php\)](#) --- Операция Where

Операция Where используется для фильтрации элементов в последовательность. Операция Where имеет два прототипа, которые описаны ниже:

## Первый прототип Where

### C#

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

Этот прототип Where принимает входную последовательность и делегат метода-предиката, а возвращает объект, который при перечислении проходит по входной последовательности, выдавая элементы, для которых делегат метода-предиката возвращает true.

Поскольку это расширяющий метод, входная последовательность в действительности не передается, поскольку операция Where вызывается с использованием синтаксиса метода экземпляра.

Благодаря расширяющим методам нет необходимости передавать первый аргумент в стандартную операцию запроса, первый аргумент которой помечен модификатором — ключевым словом this, при условии, что операция вызывается на объекте того же типа, что у первого аргумента.

При вызове Where передается делегат метода-предиката. Этот метод-предикат должен принимать тип T в качестве входного, где T — тип элементов, содержащихся во входной последовательности, и возвращать bool. Операция Where вызовет метод-предикат для каждого элемента входной последовательности и передаст ему этот элемент. Если метод-предикат вернет true, то Where выдаст этот элемент в выходную последовательность Where. Если метод-предикат вернет false, то Where этого не сделает.

Пройди тесты

## Второй прототип Where

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

### C#

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

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

Второй прототип Where идентичен первому, но с тем отличием, что он указывает на то, что делегат метода-предиката принимает дополнительный целочисленный аргумент. Этот аргумент будет индексом элемента во входной последовательности.

Нумерация индекса начинается с нуля, поэтому индексом первого элемента будет 0. Последний элемент имеет номер, соответствующий количеству элементов в последовательности минус 1.

Ниже показан пример вызова первого прототипа:

## C#

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

IEnumerable<string> sequence = cars.Where(p => p.StartsWith("F"));

foreach (string s in sequence)
    Console.WriteLine(s);
```



В приведенном примере ограничение последовательности с использованием первого прототипа операции Where осуществляется просто вызовом метода Where на последовательности, с передачей ему лямбда-выражения, возвращающего значение bool, которое указывает, должен ли элемент быть включен в выходную последовательность. В этом примере возвращаются только элементы, начинающиеся со строки "F".

Обратите внимание, что метод-предикат был передан через лямбда-выражение.

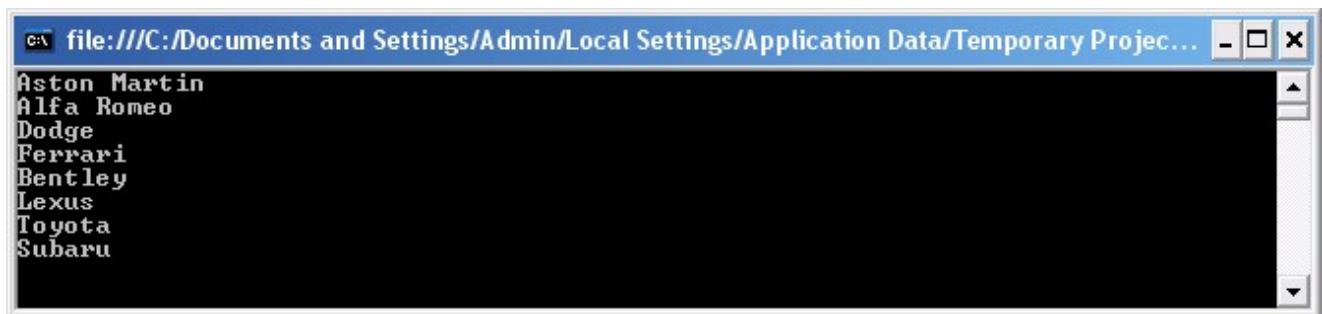
## Пройди тесты

Ниже показан код, вызывающий второй прототип операции Where. Обратите внимание, что в этом примере используется сам элемент, а не только индекс i. Это обеспечит включение в выходную последовательность элементов с нечетными индексами:

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

## C#

```
string[] cars = { "Nissan", "Aston Martin", "Chevrolet", "Alfa Romeo", "Chrysler", "Dodge",  
"BMW",  
"Ferrari", "Audi", "Bentley", "Ford", "Lexus", "Mercedes", "Toyota", "Volvo", "Su  
baru", "Жигули :)"};  
  
IEnumerable<string> sequence = cars.Where((p, i) => (i & 1) == 1);  
  
foreach (string s in sequence)  
    Console.WriteLine(s);
```



1	2	3	Вперед (2_2.php)
---	---	---	------------------



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

**Professor Web (/)**

Наш любимый хостинг (/)

Пройди тесты

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

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