

# Операции Single, SingleOrDefault, ElementAt и ElementOrDefault

[LINQ \(../base/level1/info\\_linq.php\)](#) --- [LINQ to Objects \(../level1/linq\\_index.php\)](#) --- Операции Single, SingleOrDefault, ElementAt и ElementOrDefault

## Single

Операция Single возвращает единственный элемент последовательности или единственный элемент последовательности, соответствующий предикату — в зависимости от используемого прототипа. Эта операция имеет два прототипа, описанные ниже:

### Первый прототип Single

#### C#

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

В случае использования этого прототипа операция Single перечисляет входную последовательность по имени source и возвращает единственный ее элемент.

### Второй прототип Single

Второй прототип Single принимает predicate и выглядит следующим образом:

#### C#

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

Пройди тесты  
Эта версия операции Single возвращает единственный найденный элемент, для которого predicate дает true. Если ни один из элементов не заставляет predicate вернуть true, то операция Single генерирует исключение InvalidOperationException.  
C# тест (легкий) (<https://professorweb.ru/test/c-sharp-test.html>)

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

Если любой из аргументов равен null, генерируется исключение `ArgumentNullException`.

Если последовательность `source` пуста, либо `predicate` ни разу не вернул true или нашел более одного элемента, для которых вернул true, генерируется исключение `InvalidOperationException`.

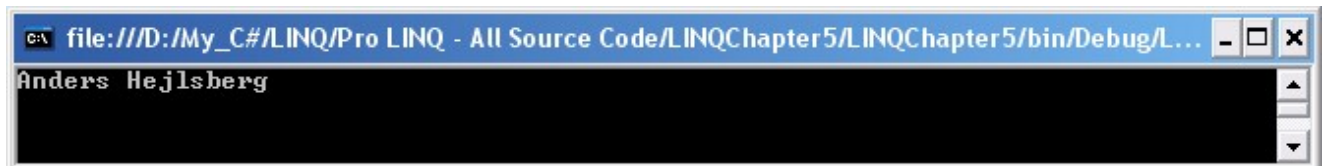
Ниже содержится пример применения первого прототипа `Single`, использующий общий класс `Employee` (`../level2/2_0.php`):

## C#

```
Employee emp = Employee.GetEmployeesArray()
    .Where(e => e.id == 3).Single();

Console.WriteLine("{0} {1}", emp.firstName, emp.lastName);
```

В данном примере вместо ожидания получения последовательности от запроса мне нужна просто ссылка на определенного сотрудника. Операция `Single` очень полезна в такой ситуации — до тех пор, пока вы можете гарантировать, что в переданной ему последовательности есть только один элемент. В данном случае, поскольку я вызвал операцию `Where`, указав уникальный ключ, я в безопасности. И вот результат:



Ниже показан пример кода, использующего второй прототип операции `Single`:

## C#

```
Employee emp = Employee.GetEmployeesArray()
    .Single(e => e.id == 3);

Console.WriteLine("{0} {1}", emp.firstName, emp.lastName);
```

Пройди тесты  
Данный код функционально эквивалентен предыдущему примеру. Вместо вызова операции `Where` для того, чтобы гарантировать наличие единственного элемента в последовательности, можно выполнить ту же операцию фильтрации  
C# тест (легкий) (<https://professorweb.ru/test/c-sharp-test.html>)  
.NET тест (средний) (<https://professorweb.ru/test/asp-test.html>)

последовательности в операции Single. Она должна вернуть только один элемент из входной последовательности, id которого равен 3.

Помните, если любой прототип операции Single не находит элемента для возврата, генерируется исключение InvalidOperationException. Избежать этого позволяет операция SingleOrDefault.

## SingleOrDefault

Операция SingleOrDefault подобна Single, но отличается поведением в случае, когда элемент не найден. Эта операция имеет два прототипа, описанные ниже:

### Первый прототип SingleOrDefault

#### C#

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

Эта версия SingleOrDefault возвращает единственный элемент, найденный во входной последовательности. Если последовательность пуста, возвращается default (T). Для ссылочных и допускающих null типов значением по умолчанию является null. Если найдено более одного элемента, генерируется исключение InvalidOperationException.

### Второй прототип SingleOrDefault

Второй прототип операции SingleOrDefault позволяет передать predicate, который определяет, какой элемент должен возвращаться.

#### C#

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

Ниже представлен пример вызова первого прототипа SingleOrDefault, когда элемент не найден. Для этого понадобилась пустая последовательность. Будет произведен вызов операции Where с указанием несуществующего значения ключа в качестве условия фильтра:

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

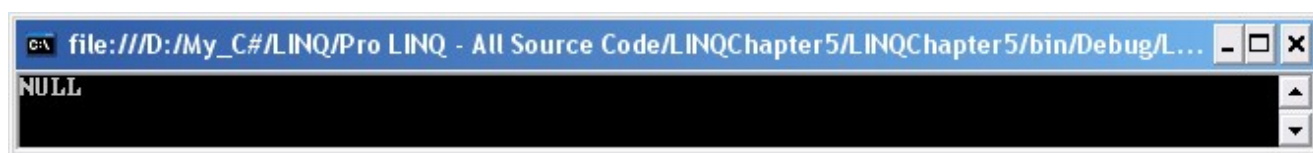
#### C#

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

```
Employee emp = Employee.GetEmployeesArray()
    .Where(e => e.id == 5).SingleOrDefault();

Console.WriteLine(emp == null ? "NULL" :
    string.Format("{0} {1}", emp.firstName, emp.lastName));
```

В коде запрашивается сотрудник с id равным 5. Поскольку известно, что такого нет, возвращается пустая последовательность. В отличие от операции Single, SingleOrDefault нормально обрабатывает пустые последовательности. Вот результат работы примера:



## ElementAt

Операция ElementAt возвращает элемент из исходной последовательности по указанному индексу. Эта операция имеет один прототип, описанный ниже:

### C#

```
public static T ElementAt<T>(
    this IEnumerable<T> source,
    int index);
```

Если последовательность реализует `IList<T>`, то интерфейс `IList` используется для извлечения индексированного элемента непосредственно. Если же последовательность не реализует `IList<T>`, то последовательность перечисляется до тех пор, пока не будет достигнут указанный индексом элемент. *Исключение* `ArgumentOutOfRangeException` генерируется, если индекс меньше нуля либо больше или равен количеству элементов в последовательности.

Ниже приведен пример вызова единственного прототипа операции ElementAt:

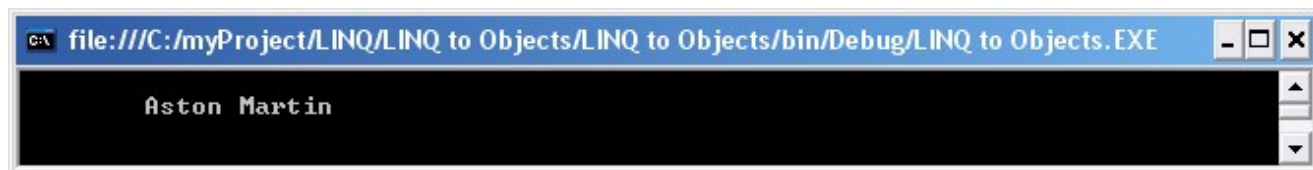
Пройди тесты **C#**

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

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

```
string[] cars = { "Alfa Romeo", "Aston Martin", "Audi", "Nissan", "Chevrolet", "Chrysler",  
"Dodge", "BMW",  
                "Ferrari", "Bentley", "Ford", "Lexus", "Mercedes", "Toyota",  
"Volvo", "Subaru", "Жигули :)"};  
  
string auto = cars.ElementAt(1);  
Console.WriteLine("\n\t" + auto);
```

Здесь указано, что требуется элемент, индекс которого равен 1, т.е. второй по счету. И вот результат этого запроса:



## ElementOrDefault

Операция ElementOrDefault возвращает из исходной последовательности элемент, имеющий указанный индекс местоположения. Эта операция имеет один прототип, описанный ниже:

### C#

```
public static T ElementOrDefault<T>(
    this IEnumerable<T> source,
    int index);
```

Если последовательность реализует `IList<T>`, то интерфейс `IList` используется для извлечения индексированного элемента непосредственно. Если же последовательность не реализует `IList<T>`, то последовательность перечисляется до тех пор, пока не будет достигнут указанный индексом элемент. Если заданный индекс меньше нуля либо больше или равен количеству элементов в последовательности возвращается `default(T)`. Для ссылочных и допускающих null типов значение по умолчанию равно null. Это поведение отличает его от операции `ElementAt`.

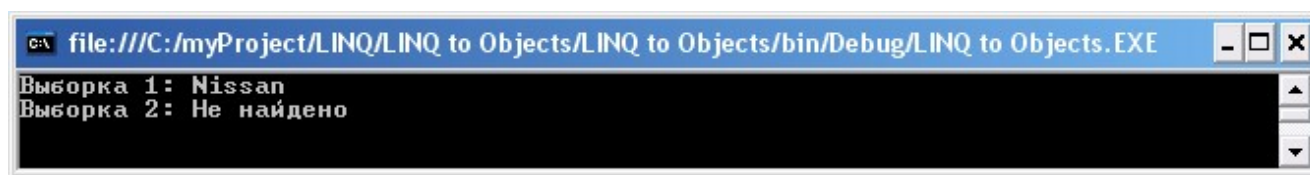
Ниже содержится пример вызова операции `ElementOrDefault`:  
Пройди тесты

### C#

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

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

```
string[] cars = { "Alfa Romeo", "Aston Martin", "Audi", "Nissan", "Chevrolet", "Chrysler",  
"Dodge", "BMW",  
                "Ferrari", "Bentley", "Ford", "Lexus", "Mercedes", "Toyota",  
"Volvo", "Subaru", "Жигули :)"};  
  
string auto = cars.ElementAtOrDefault(3);  
Console.WriteLine("Выборка 1: " + auto);  
  
auto = cars.ElementAtOrDefault(-100);  
Console.WriteLine("Выборка 2: ");  
Console.WriteLine(auto == null ? "Не найдено" : auto);
```



Назад (3_5.php)	5	6	7	Вперед (3_7.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>)