

Операции ToArray и ToList

[LINQ \(../base/level1/info_linq.php\)](#) --- [LINQ to Objects \(../level1/linq_index.php\)](#) --- Операции ToArray и ToList

Следующие операции преобразования предоставляют простой и удобный способ преобразования последовательностей в другие типы коллекций.

ToArray

Операция ToArray создает массив типа T из входной последовательности типа T. Эта операция имеет один прототип, описанный ниже:

C#

```
public static T[] ToArray<T>(
    this IEnumerable<T> source);
```

Эта операция берет входную последовательность source с элементами типа T и возвращает массив элементов типа T.

Для примера, демонстрирующего работу операции ToArray, понадобится последовательность типа IEnumerable<T>. Последовательность этого типа создается вызовом операции Enumerable.Range(), которая была описана ранее. Имея такую последовательность, можно вызывать операцию ToArray для создания массива, как показано ниже:

C#

```
IEnumerable<int> item = Enumerable.Range(1, 20);

Console.WriteLine("Начальный тип: " + item);

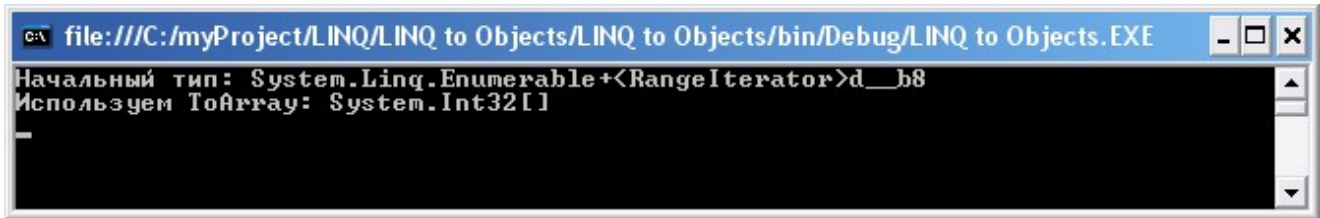
int[] arr = item.ToArray();
```

Пройди тесты

```
Console.WriteLine("Используем ToArray: " + arr);
```

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

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



Эта операция часто бывает удобной для кэширования последовательности, чтобы она не могла измениться до начала ее перечисления. К тому же, поскольку эта операция не является отложенной и выполняется немедленно, множество перечислений, созданных на одном массиве, всегда будут видеть одни и те же данные.

ToList

Операция ToList создает List типа T из входной последовательности типа T. Эта операция имеет один прототип, описанный ниже:

C#

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

Данная операция принимает последовательность по имени source элементов типа T и возвращает список List элементов типа T. Ниже демонстрируется применение операции ToList:

C#

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

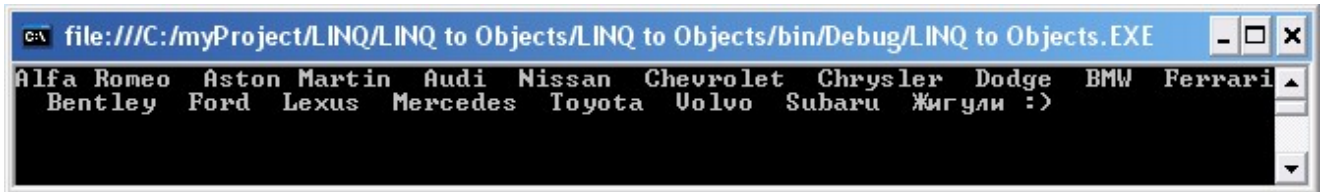
List<string> auto = cars.ToList();

foreach (string s in auto)
    Console.Write(s + " ");
```

В приведенном коде используется массив, состоящий из названий машин. Этот массив преобразуется в список List<string>. Вот результат:

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

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



```
file:///C:/myProject/LINQ/LINQ to Objects/LINQ to Objects/bin/Debug/LINQ to Objects.EXE
Alfa Romeo  Aston Martin  Audi  Nissan  Chevrolet  Chrysler  Dodge  BMW  Ferrari
Bentley  Ford  Lexus  Mercedes  Toyota  Volvo  Subaru  Жигули :>
```

Эта операция часто полезна для кэширования последовательности, чтобы она не могла измениться перед ее перечислением. Также, поскольку эта операция не является отложенной и выполняется немедленно, множество перечислений на созданном списке `List<T>` всегда видят одинаковые данные.

1	2	3	Вперед (3_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>)