



Сортировка

Последнее обновление: 31.10.2015



Для упорядочивания полученных из бд данных по возрастанию служит метод **OrderBy** или оператор **orderby**. Например, отсортируем объекты по возрастанию по свойству Name:

```
using(PhoneContext db = new PhoneContext())
{
    var phones = db.Phones.OrderBy(p=>p.Name);
    foreach (Phone p in phones)
        Console.WriteLine("{0}.{1} - {2}", p.Id, p.Name, p.Price);
}
```

В результате Entity Framework будет генерировать следующее выражение SQL, которое будет упорядочивать данные:

```
SELECT [Extent1].[Id] AS [Id],
       [Extent1].[Name] AS [Name],
       [Extent1].[Price] AS [Price],
       [Extent1].[CompanyId] AS [CompanyId]
FROM [dbo].[Phones] AS [Extent1]
ORDER BY [Extent1].[Name] ASC
```

В качестве альтернативы методу OrderBy можно использовать оператор orderby:

```
var phones = from p in db.Phones
              orderby p.Name
              select p;
foreach (Phone p in phones)
    Console.WriteLine("{0}.{1} - {2}", p.Id, p.Name, p.Price);
```

Для сортировки по убыванию применяется метод **OrderByDescending()**:

```
var phones = db.Phones.OrderByDescending(p=>p.Name);
```

Если нам надо отсортировать данные сразу по нескольким критериям, то мы можем применить методы **ThenBy()** (для сортировки по возрастанию) и **ThenByDescending()**. Например, отсортируем результат проекции по двум столбцам:

```
var phones = db.Phones
    .Select(p => new { Name = p.Name, Company = p.Company.Name, Price = p.Price })
    .OrderBy(p => p.Price)
    .ThenBy(p=>p.Company);
```

[Назад](#) [Содержание](#) [Вперед](#)



[Вконтакте](#) | [Twitter](#) | [Google+](#) | [Канал сайта на youtube](#) | [Помощь сайту](#)

Контакты для связи: metanit22@mail.ru

Copyright © metanit.com, 2012-2017. Все права защищены.