LINQ

Язык интегрированных запросов в C# 2010

ДЛЯ ПРОФЕССИОНАЛОВ

Pro LINQ

Language Integrated Query in C# 2010

Adam Freeman and
Joseph C. Rattz, Jr.

Apress®

LINQ

Язык интегрированных запросов в C# 2010

ДЛЯ ПРОФЕССИОНАЛОВ

Адам Фримен
Джозеф Раттц-мл.

Москва • Санкт-Петербург • Киев
2011

ББК 32.973.26-018.2.75
Ф88
УДК 681.3.07

Издательский дом "Вильямс"
Зав. редакцией *С.Н. Тригуб*
Перевод с английского *Н.А. Мухина*
Под редакцией *Ю.Н. Артеменко*

По общим вопросам обращайтесь в Издательский дом "Вильямс" по адресу:
info@williamspublishing.com, http://www.williamspublishing.com

Фримен, Адам, **Раттц-мл.**, Джозеф С.

Ф88 LINQ: язык интегрированных запросов в C# 2010 для профессионалов. : Пер. с англ. — М. : ООО "И.Д. Вильямс", 2011. — 656 с. : ил. — Парал. тит. англ.

ISBN 978-5-8459-1701-0 (рус.)

ББК 32.973.26-018.2.75

Научно-популярное издание
Адам Фримен, Джозеф С. Раттц-мл.

LINQ: язык интегрированных запросов в C# 2010 для профессионалов

Верстка *Т.Н. Артеменко*
Художественный редактор *В.Г. Павлютин*

Подписано в печать 24.01.2011. Формат 70×100/16.
Гарнитура Times. Печать офсетная.
Усл. печ. л. 52,89. Уч.-изд. л. 37,6.
Тираж 1500 экз. Заказ № 0000.

Отпечатано по технологии CtP
в ОАО "Печатный двор" им. А. М. Горького
197110, Санкт-Петербург, Чкаловский пр., 15.

ООО "И. Д. Вильямс", 127055, г. Москва, ул. Лесная, д. 43, стр. 1

ISBN 978-5-8459-1701-0 (рус.)
ISBN 978-1430-22653-6 (англ.)

Оглавление

Содержание

Об авторах

Адам Фримен — профессионал в области информационных технологий, который занимал ведущие должности в ряде компаний, последняя из которых — директор по развитию технологий и главный операционной директор в глобальном банке. Он написал несколько книг по Java и .NET и давно интересуется параллельными вычислениями.

Джозеф Раттц-мл. неосознанно начал свою карьеру в разработке программного обеспечения в 1990 г., когда друг попросил его помочь в написании текстового редактора ANSI под названием ANSI Master для компьютера Commodore Amiga. Вскоре за этим последовала игра в палача (The Gallows — "Виселица"). От этих программ, написанных на компилируемом Basic, он перешел к программированию на C, в поисках более высокой скорости и мощи. После этого Джо разрабатывал приложения для *JumpDisk* — журнала Amiga на дисках, а также для журнала *Amiga World*. Из-за того, что ему пришлось работать в маленьком городе в относительной изоляции, Джо изучил все неправильные способы написания кода. Это обучение происходило при попытках усовершенствовать плохо написанные приложения, в процессе которых он осознал важность написания хорошо сопровождаемого кода. Впервые познакомившись с отладчиком уровня исходного кода, он влюбился в него с первого взгляда.

Двумя годами позже Джо получил свою первую работу в качестве разработчика программного обеспечения в Policy Management Systems Corporation, как программист начального уровня, разрабатывающий клиент-серверное приложение системы страхования для OS/2 и Presentation Manager. С годами он добавил к своему багажу знаний C++, Unix, Java, ASP, ASP.NET, C#, HTML, DHTML и XML, разрабатывая приложения для SCT, DocuCorp, IBM, Комитета по проведению олимпийских игр в Атланте, CheckFree, NCR, EDS, Delta Technology, Radiant Systems и Genuine Parts Company. Джо нравились творческие аспекты дизайна пользовательского интерфейса, и он осознал необходимость дисциплины при разработке программного обеспечения серверной стороны. Но когда у него была такая возможность, его любимым времяпрепровождением была отладка кода.

Сегодня Джо можно найти в Genuine Parts Company — родительской компании NAPA — в подразделении Automotive Parts Group Information Systems, где он трудится над своим детищем — веб-сайтом Storefront. Этот сайт обслуживает хранилища NAPA, предоставляя их счета и данные в сети систем AS/400. Связаться к Джо можно через его веб-сайт по адресу `www.linqdev.com`.

О техническом рецензенте

Фабио Клаудио Феррачати — старший консультант и аналитик-разработчик, имеющий дело с технологиями Microsoft. Он работает в компании Brain Force (`www.brainforce.com`), в ее итальянском подразделении (`www.brainforce.it`). Является сертифицированным разработчиком решений Microsoft для .NET, сертифицированным разработчиком приложений Microsoft для .NET, сертифицированным профессионалом Microsoft, а также плодовитым автором и техническим рецензентом. В течение последних десяти лет он написал множество статей для итальянских и международных изданий, а также является соавтором более десятка книг на различные компьютерные темы.

Благодарности

Мы хотели бы поблагодарить Джона Скита (Jon Skeet), Джадсона Уайта (Judson White) и всех остальных сотрудников издательства Apress за совместную работу над этой книгой. В частности, благодарим Мери Тобин (Mary Tobin) за отслеживание работ и Эвана Бакингема (Ewan Buckingham) за рудактирование книги. Мы также выражаем благодарность Ким Уимпсетт (Kim Wimpsett) и Фабио Феррачати (Fabio Ferracchiati) за их усилия, которые сделали эту книгу намного лучше, чем она была в начале.

Адам Фримен и Джозеф Раттц-мл.

От издательства

Вы, читатель этой книги, и есть главный ее критик и комментатор. Мы ценим ваше мнение и хотим знать, что было сделано нами правильно, что можно было сделать лучше и что еще вы хотели бы увидеть изданным нами. Нам интересно услышать и любые другие замечания, которые вам хотелось бы высказать в наш адрес.

Мы ждем ваших комментариев и надеемся на них. Вы можете прислать нам бумажное или электронное письмо, либо просто посетить наш веб-сервер и оставить свои замечания там. Одним словом, любым удобным для вас способом дайте нам знать, нравится или нет вам эта книга, а также выскажите свое мнение о том, как сделать наши книги более интересными для вас.

Посылая письмо или сообщение, не забудьте указать название книги и ее авторов, а также ваш обратный адрес. Мы внимательно ознакомимся с вашим мнением и обязательно учтем его при отборе и подготовке к изданию последующих книг.

Наши координаты:

E-mail: info@williamspublishing.com

WWW: http://www.williamspublishing.com

Информация для писем из:

России: 127055, г. Москва, ул. Лесная, д. 43, стр. 1

Украины: 03150, Киев, а/я 152

Исходный код примеров

Исходный код примеров, рассмотренных в книге, доступен для загрузки на сайте издательства по адресу http://www.williamspublishing.com.

ЧАСТЬ I

LINQ: язык интегрированных запросов в C# 2010

В этой части

ГЛАВА 1

Знакомство с LINQ

Листинг 1.1. Программа “Hello LINQ”

```
using System;
using System.Linq;
string[] greetings = {"hello world", "hello LINQ", "hello Apress"};
var items =
  from s in greetings
  where s.EndsWith("LINQ")
  select s;
foreach (var item in items)
  Console.WriteLine(item);
```

На заметку! Код из листинга 1.1 был добавлен в проект, созданный с помощью шаблона консольного приложения в Visual Studio 2010. Если это еще не сделано, добавьте директиву `using` для пространства имен `System.Linq`.

Запуск приведенного выше кода по нажатию <Ctrl+F5> выдаст следующий вывод в окно консоли:

```
hello LINQ
```

Сдвиг парадигмы

Вы почувствовали только что, как *ваш* мир сдвинулся с места? Как разработчик .NET, вы должны были это почувствовать. То, что продемонстрировано в тривиальном примере программы из листинга 1.1, похоже на запрос на языке структурированных запросов (Structured Query Language — SQL) к массиву строк[1]. Взгляните на конструкцию `where`. Она выглядит так, будто использовался метод `EndsWith` объекта `string`, потому что так оно и есть. Может возникнуть вопрос: а как насчет типа переменной `var`? Выполняет ли по-прежнему компилятор C# контроль типов? Ответ — да, он проверяет статически типы во время компиляции. Какое средство или средства C# позволяют все это? Ответ: Microsoft Language Integrated Query (язык интегрированных запросов Microsoft), иначе называемый *LINQ*.

[1] Важно отметить, что порядок следования конструкций в запросе противоположен типичному SQL. К тому же добавлена часть `s in` запроса, которая представляет ссылку на набор элементов, содержащихся в источнике, которым в данном случае является массив строк `"hello world"`, `"hello LINQ"` и `"hello Apress"`.

Запрос к XML

В то время как пример из листинга 1.1 достаточно тривиален, пример в листинге 1.2 начинает отражать потенциальную мощь, которую вручает LINQ в руки разработчика .NET. Он демонстрирует легкость, с которой можно взаимодействовать и опрашивать данные XML (Extensible Markup Language — расширяемый язык разметки) с помощью API-интерфейса LINQ to XML. Обратите внимание, как на основе данных XML конструируется объект по имени `books`, с которым впоследствии можно взаимодействовать программно.

Листинг 1.2. Простой запрос к XML-разметке с использованием LINQ to XML

```
using System;
using System.Linq;
using System.Xml.Linq;
XElement books = XElement.Parse(
  @"<books>
    <book>
      <title>Pro LINQ: Language Integrated Query in C# 2010</title>
      <author>Joe Rattz</author>
    </book>
    <book>
      <title>Pro .NET 4.0 Parallel Programming in C#</title>
      <author>Adam Freeman</author>
    </book>
    <book>
      <title>Pro VB 2010 and the .NET 4.0 Platform</title>
      <author>Andrew Troelsen</author>
    </book>
  </books>");
var titles =
  from book in books.Elements("book")
  where (string) book.Element("author") == "Joe Rattz"
  select book.Element("title");
foreach(var title in titles)
  Console.WriteLine(title.Value);
```

На заметку! Код в листинге 1.2 требует добавления к ссылкам проекта сборки `System.Xml.Linq.dll`, если это еще не сделано. Также обратите внимание, что добавлена директива `using` для пространства имен `System.Xml.Linq`.

Запуск предыдущего кода нажатием <Ctrl+F5> приводит к выводу следующих данных в окно консоли:

```
Pro LINQ: Language Integrated Query in C# 2010
```

Обратите внимание на то, как данные XML были разобраны для помещения в объект `XElement`. Объект `XmlDocument` нигде не создавался. Среди преимуществ LINQ to XML — расширения, которые он привносит в XML API. Теперь вместо того, чтобы сосредоточивать все вокруг `XmlDocument`, как того требует W3C Document Object Model (DOM) XML API, интерфейс LINQ to XML позволяет разработчику взаимодействовать на уровне элемента, используя класс `XElement`.

На заметку! В дополнение к средствам запросов, LINQ to XML предоставляет более мощный и простой способ использования интерфейса для работы с данными XML.

Здесь также применялся SQL-подобный синтаксис для опроса данных XML, как если бы это была база данных.

Запрос к базе данных SQL Server

В следующем примере демонстрирует использование LINQ to SQL для опроса таблиц базы данных. В коде из листинга 1.3 выполняется запрос к стандартной базе данных примеров Microsoft Northwind.

Листинг 1.3. Простой запрос XML с использованием LINQ to XML

```
using System.Linq;
using System.Data.Linq;
using nwind;
Northwind db = new Northwind(@"Data Source=.\SQLEXPRESS;Initial Catalog=Northwind");
var custs =
  from c in db.Customers
  where c.City == "Rio de Janeiro"
  select c;
foreach (var cust in custs)
  Console.WriteLine("{0}", cust.CompanyName);
```

На заметку! Код в листинге 1.3 требует добавления сборки `System.Data.Linq.dll` к списку ссылок проекта, если это не сделано ранее. Также обратите внимание, что добавлена директива `using` для пространства имен `System.Xml.Linq`.

В коде была добавлена директива `using` для пространства имен `nwind`. В этом примере должна использоваться утилита командной строки SQLMetal или Object Relational Designer, чтобы сгенерировать сущностные классы для целевой базы данных, которой в данном случае является база данных примеров Microsoft Northwind. В главе 12 показано, как это делать с SQLMetal. Сгенерированные сущностные классы создаются в пространстве имен `nwind`, которое указано при их генерации. После этого в проект добавляется сгенерированный SQLMetal исходный модуль, а также директива `using` для пространства имен `nwind`.

На заметку! Для корректной установки соединения может понадобиться изменить строку соединения, переданную конструктору Northwind в листинге 1.3. В разделе "`DataContext()` и `[Your] DataContext()`" главы 16 описаны разные способы подключения к базе данных.

Запуск предыдущего кода нажатием <Ctrl+F5> приводит к выводу следующих данных в окно консоли:

```
Hanari Carnes
Que Delícia
Ricardo Adocicados
```

В этом примере демонстрируется опрос таблицы `Customers` из базы данных Northwind на предмет списка заказчиков из Рио-де-Жанейро (Rio de Janeiro). Хотя может показаться, что здесь не происходит ничего нового или особенного, чего нельзя было бы получить существующими средствами, все же есть серьезные отличия. Важнее всего то, что этот запрос интегрирован в язык, а это значит, что получается поддержка уровня языка, включающая проверку синтаксиса и средство IntelliSense. Ушли в прошлое те дни, когда запрос SQL записывался в строку, и ошибку невозможно было обнаружить вплоть до выполнения кода. Вы хотите сделать конструкцию `where` зависящей от поля