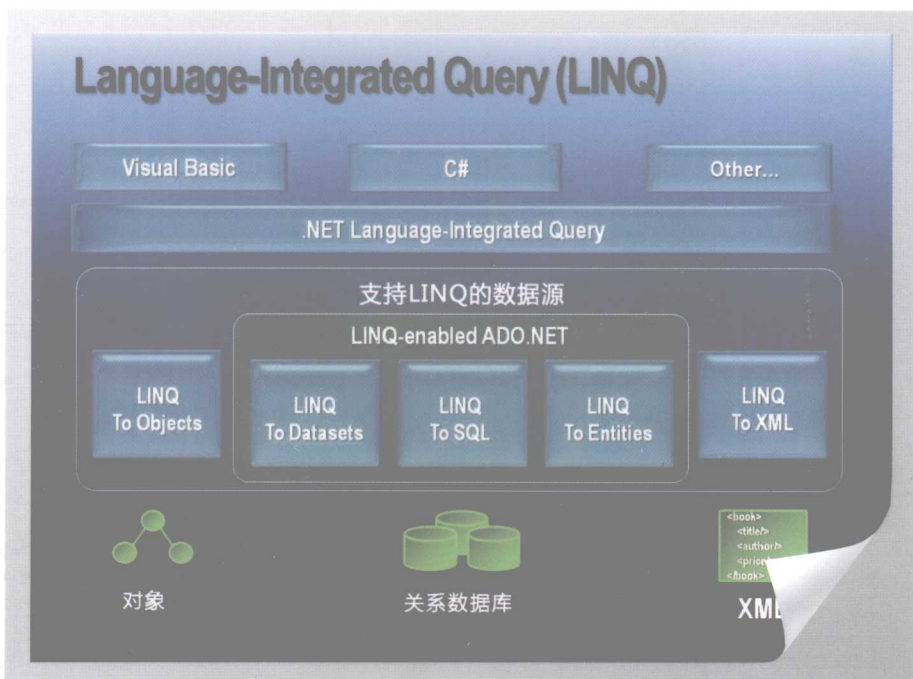




Microsoft®

Visual Studio® 2008



LINQ

入门及应用

王的强 张晓娜 周 静◎编著

- ◎ 本书介绍微软最前沿的**数据查询编程**技术
- ◎ 内容由**160**个知识点实例加**5**个综合应用案例组成
- ◎ 光盘中提供了涉及到的全部**示例源程序项目**
- ◎ 对于将要从事正规商业开发的人员有很强的**示范性**和**指导作用**



清华大学出版社

LINQ 入门及应用

王的强 张晓娜 周 静 编著

清华大学出版社

北 京

内 容 简 介

本书系统、全面地介绍了微软最新推出的 LINQ 数据开发技术。

本书采取以实践为主、以理论为辅的方法,首先对 C#新改进的一些语法做了详细的剖析,让读者领会 C#语法的新特点,为后面的学习打下基础,然后对 LINQ 各个子句和组件进行了实例讲解,通过一个个生动的示例,带领读者由浅入深、系统地学习和掌握 LINQ 技术。

本书对开发步骤的详细介绍,有助于读者更好地掌握 Microsoft Visual Studio 2008 开发环境。各个章节中的综合实例,除了充分演示 LINQ 技术外,还努力引导读者适应面向对象的开发方式。

配书光盘中提供了书中所有的源程序(均为 VS2008 应用项目),全部项目均经过严格的测试并通过。

本书可供相关技术研究人员、广大.NET 应用程序开发人员和用户参考,也可作为高等院校计算机、电子商务以及信息类专业的教材。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

LINQ 入门与应用/王的强,张晓娜,周静编著. —北京:清华大学出版社,2009.5

ISBN 978-7-302-20025-3

I. L… II. ①王… ②张… ③周… III. ①计算机网络—程序设计 ②C 语言—程序设计 IV. TP393
TP312

中国版本图书馆 CIP 数据核字(2009)第 063660 号

责任编辑:张 瑜 宋延清

装帧设计:杨玉兰

责任印制:王秀菊

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者:北京密云胶印厂

装 订 者:三河市新茂装订有限公司

经 销:全国新华书店

开 本:185×260 印 张:27.5 字 数:661 千字

附光盘 1 张

版 次:2009 年 5 月第 1 版

印 次:2009 年 5 月第 1 次印刷

印 数:1~4000

定 价:48.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系
调换。联系电话:(010)62770177 转 3103 产品编号:032135-01

前言

IT 业界在 2008 年有很多让人激动的事情发生,例如微软公司在 2008 年早些时候,正式发布了 Visual Studio 2008 和 .NET Framework 3.5 版,对 .NET 做了很多重大的改进,其中 LINQ 技术尤为耀眼。

LINQ 的多项革命性特征必将引领信息处理技术走向新的高度。所以众多读者期望能更加深入、透彻地了解 LINQ 所带来的各种编程优势及其对开发工作的促进作用,本书以理论联系实际的讲解风格,通过介绍如何使用 LINQ 技术解决开发过程中经常遇到的问题,来帮助读者更好地理解 LINQ 技术,为在生产环境中使用 LINQ 技术打下基础。

本书共分 7 章,主要内容说明如下。

第 1 章: LINQ 介绍。

叙述 LINQ 的一些概念和使用 LINQ 编程的优势,还简单介绍开发 LINQ 的编程环境 Visual Studio 2008 的安装和配置。在叙述中,用三个实例引领读者了解 LINQ to Objects、LINQ to ADO.NET、LINQ to XML,同时对实例的开发步骤都进行非常细致的讲解,目的是让读者尽快熟悉 Visual Studio 2008 开发环境,为后面的学习打下基础。

第 2 章: C#语言的新特性及相关知识。

首先叙述 .NET 各个版本之间的关系,然后分别讲解 C#语言的一些新特性。在讲解中,尽力避免晦涩难懂的理论叙述,通过大量有趣、实用的例子,演示 C#语言的自动属性、可空类型、泛型、代理、迭代器、隐式类型局部变量、对象集合初始化器、匿名类型、扩展方法、匿名方法、Lambda 表达式等特性。最后的综合实例更是充分利用这些特性,带给读者关于 C#语言的完美体验。

第 3 章: LINQ 基本子句介绍。

这一章讲述 LINQ 表达式的 from 子句、where 子句、select 子句、group 子句、into 子句、排序子句、let 子句、join 子句等基本子句,这些子句都是 LINQ 标准查询运算符的一部分。

第 4 章: 用于集合、文件、字符串对象的 LINQ(LINQ to Objects)。

通过大量实例讲解 LINQ to Objects 的查询运算符, LINQ to Objects 是 LINQ 技术的基础,掌握这些查询运算符也就掌握了 LINQ 技术的核心操作。该章最后的综合实例演示 LINQ 在 ASP.NET 和 WinForm 下的运用,并且顺便让读者感受了一下 ASP.NET 中新增的 Ajax 技术。

第 5 章: 用于数据库访问的 LINQ(LINQ to SQL)。

通过一些实例讲解并演示如何通过 Microsoft Visual Studio 2008 设计器、手动方式、XML 文件来建立 LINQ to SQL 的实体对象;介绍用于映射数据库及进行相关操作的 DataContext 对象、操作数据库单一表格的 Table<TEntity>类、操作数据库一对多关系表格的 EntitySet<TEntity>类。掌握这些知识后,便可以熟练使用 LINQ to SQL 进行数据库开发了。该章最后的综合实例,使用 LINQ to SQL 操作一对多关系的表格,演示实体对象的序列化与反序列化,讲解如何正确绑定 DataGridView 的数据源等,相信通过这个实例,读者对 LINQ

to SQL 的认识会更加深入。

第6章：用于XML访问的LINQ(LINQ to XML)。

通过一些实例讲解并演示 LINQ to XML 的一些概念及基本操作，使用 XML 轴方法处理 XML 元素，使用 LINQ 表达式处理 XML 元素，如何修改、变造、输出 XML 树，其中通过变造 XML 树的实例，读者可以体验到 LINQ 的强大功能。该章最后的综合实例使用 LINQ to SQL 操作一对多关系的表格，统计订单数据，之后使用 LINQ to XML 将实体对象转换为 XML 节点，实现一个较为完整的 RSS 生成类。

第7章：综合应用实例——商品管理解决方案。

在企业开发中，一般都遵从多层结构开发方式。例如：一个解决方案，需要有构建持久保存数据的数据层、实现商业逻辑的逻辑层、与用户沟通/交换数据的表示层，这就是一个典型的 3 层结构系统。那种生成界面之后双击控件直接进行编码的方式，是无法适应企业开发中复杂、繁琐、不断重构的环境的。该章的解决方案实例，分为 3 个项目来实现一个数据层、一个商店模型、一个在线商品浏览页面、一个收银台桌面程序等。在 DLL 类库项目中，演示 LINQ to SQL 技术；在 Windows 窗体项目中，演示 LINQ to Objects 技术；在 ASP.NET 项目中，演示 LINQ to XML 技术。

本书由王的强、张晓娜、周静编著。参加编写的还有：王文伟、张元兰、赵蕾、王曦悦、王芹、王永生、王少琨、赵蓓、胡海青、胡艺凡、赵滨、张启荣、赵钢、王黛涵、王广成、王晓泳、秦伟斌、刘勇等。

由于作者水平有限，本书的内容难免会有纰漏和不足之处，恳请各位同仁、专家和读者批评指正。

作者
2009 年 4 月

目 录

第 1 章 LINQ 介绍.....	1	3.2.2 复合 from 子句.....	69
1.1 什么是 LINQ	1	3.2.3 多个 from 子句.....	71
1.2 为什么要用 LINQ	2	3.3 where 子句.....	73
1.3 搭建 LINQ 开发环境	2	3.3.1 常见的 where 子句查询.....	73
1.3.1 安装 Visual Studio 2008	2	3.3.2 在 where 子句中使用自定义函数	75
1.3.2 配置 Visual Studio 2008	4	3.3.3 动态谓词的筛选	76
1.4 初识 LINQ 技术	5	3.4 select 子句	78
1.4.1 初识 LINQ to Objects	5	3.4.1 输出查询结果	78
1.4.2 初识 LINQ to ADO.NET	8	3.4.2 对查询结果进行投影.....	78
1.4.3 初识 LINQ to XML	21	3.5 group 子句.....	80
1.5 小结.....	26	3.5.1 IGrouping<TKey, TElement> 泛型接口	80
第 2 章 C#语言的新特性及相关知识	27	3.5.2 分组查询	81
2.1 .NET 各个版本之间的关系	27	3.6 into 子句	83
2.2 C#语言的新特性	28	3.7 排序子句	86
2.2.1 自动属性和属性访问器的保护级别	28	3.7.1 OrderBy 和 OrderByDescending.....	86
2.2.2 可空类型.....	29	3.7.2 ThenBy 和 ThenByDescending.....	87
2.2.3 泛型.....	32	3.8 let 子句	90
2.2.4 代理.....	35	3.9 join 子句.....	91
2.2.5 迭代器与 yield 关键字	41	3.10 小结	94
2.2.6 隐式类型的局部变量.....	44	第 4 章 用于集合、文件、字符串对象的 LINQ(LINQ to Objects).....	95
2.2.7 对象和集合初始化器.....	45	4.1 LINQ 标准查询运算符概述	95
2.2.8 匿名类型.....	49	4.2 Enumerable 类的介绍	95
2.2.9 扩展方法.....	51	4.3 用于延期执行的 Enumerable 类方法成员	97
2.2.10 匿名方法和 Lambda 表达式.....	53	4.3.1 Take 方法	98
2.3 综合实例: C#高亮编辑及编译运行器.....	56	4.3.2 TakeWhile 方法.....	99
2.4 小结.....	64	4.3.3 Skip 方法	101
第 3 章 LINQ 基本子句介绍	65	4.3.4 SkipWhile 方法	103
3.1 概述.....	65		
3.2 from 子句	66		
3.2.1 单个 from 子句	66		

4.3.5	Take、TakeWhile、Skip、SkipWhile 小结.....	105	4.4.23	Max 方法.....	194
4.3.6	Reverse 方法.....	107	4.5	综合实例.....	198
4.3.7	Distinct 方法.....	108	4.5.1	使用 Ajax 技术的 ASP.NET 项目: 敏感词过滤.....	198
4.3.8	Union 方法.....	111	4.5.2	WinForm 项目: 文件查找和分组.....	202
4.3.9	Concat 方法.....	113	4.6	小结.....	208
4.3.10	Intersect 方法.....	114	第 5 章	用于数据库访问的 LINQ (LINQ to SQL).....	209
4.3.11	Except 方法.....	116	5.1	LINQ to SQL 概述.....	209
4.3.12	Range 方法.....	119	5.2	对象模型的建立.....	210
4.3.13	Repeat 方法.....	120	5.2.1	数据库对象的建立.....	211
4.3.14	Empty 方法.....	121	5.2.2	使用 Visual Studio 2008 建立实体类.....	213
4.3.15	DefaultIfEmpty 方法.....	122	5.2.3	手动建立实体类.....	215
4.3.16	Cast 方法.....	125	5.2.4	使用 XML 映射文件.....	221
4.3.17	OfType 方法.....	126	5.3	用于数据库连接的 DataContext 对象的成员介绍.....	225
4.3.18	AsEnumerable 方法.....	127	5.3.1	DataContext 构造函数.....	225
4.4	用于立即执行的 Enumerable 类方法成员.....	129	5.3.2	CreateDatabase 方法.....	227
4.4.1	ToArray 方法.....	129	5.3.3	DatabaseExists 方法.....	229
4.4.2	ToList 方法.....	131	5.3.4	DeleteDatabase 方法.....	231
4.4.3	ToDictionary 方法.....	132	5.3.5	GetTable 方法.....	232
4.4.4	ToLookup 方法.....	140	5.3.6	SubmitChanges 方法.....	234
4.4.5	SequenceEqual 方法.....	150	5.3.7	GetChangeSet 方法.....	235
4.4.6	First 方法.....	153	5.3.8	Refresh 方法.....	237
4.4.7	FirstOrDefault 方法.....	155	5.3.9	ExecuteQuery 方法.....	240
4.4.8	Last 方法.....	157	5.3.10	ExecuteCommand 方法.....	243
4.4.9	LastOrDefault 方法.....	159	5.3.11	Translate 方法.....	244
4.4.10	Single 方法.....	162	5.3.12	ChangeConflicts 属性.....	251
4.4.11	SingleOrDefault 方法.....	164	5.3.13	Transaction 属性.....	251
4.4.12	ElementAt 方法.....	167	5.3.14	ObjectTrackingEnabled 属性.....	251
4.4.13	ElementAtOrDefault 方法.....	168	5.3.15	Log 属性.....	252
4.4.14	All 方法.....	169	5.4	用 LINQ 操作数据库单一表格.....	257
4.4.15	Any 方法.....	170	5.4.1	Table<TEntity>类.....	257
4.4.16	Contains 方法.....	173	5.4.2	Attach 方法.....	257
4.4.17	Count 方法.....	175	5.4.3	AttachAll 方法.....	264
4.4.18	LongCount 方法.....	177			
4.4.19	Aggregate 方法.....	179			
4.4.20	Sum 方法.....	183			
4.4.21	Average 方法.....	186			
4.4.22	Min 方法.....	190			

5.4.4	GetOriginalEntityState 方法	267	6.3.4	删除元素的轴方法	340
5.4.5	GetModifiedMembers 方法	269	6.3.5	替换元素的轴方法	342
5.4.6	InsertOnSubmit 方法	270	6.3.6	根据指定的值自动添加、 删除、修改子元素	345
5.4.7	InsertAllOnSubmit 方法	272	6.3.7	获取元素属性的轴方法	346
5.4.8	DeleteOnSubmit 方法	274	6.3.8	替换元素属性的轴方法	349
5.4.9	DeleteAllOnSubmit 方法	276	6.3.9	删除元素的属性的轴方法	350
5.5	用 LINQ 操作数据库一对多表格	278	6.3.10	根据指定的值自动添加、 删除、修改元素的属性	352
5.5.1	EntitySet<TEntity>类	278	6.3.11	处理批注的轴方法	353
5.5.2	Add 方法	279	6.4	使用 LINQ 表达式处理 XML 元素	357
5.5.3	AddRange 方法	282	6.4.1	筛选 XML 元素	357
5.5.4	Insert 方法	284	6.4.2	对 XML 元素进行排序	359
5.5.5	Remove 方法	285	6.4.3	对 XML 元素进行计算	360
5.5.6	RemoveAt 方法	292	6.4.4	剔除 XML 树中符合条件的 元素	363
5.5.7	Clear 方法	294	6.4.5	变造 XML 树	365
5.6	综合实例：客户产品订单处理 系统	296	6.5	使用 LINQ to XML 序列化 XML 树	369
5.6.1	数据类	297	6.5.1	输出 XML 树到字符串	369
5.6.2	序列化与反序列化	299	6.5.2	输出 XML 树到 TextWriter 对象	371
5.6.3	控制器类	307	6.5.3	输出 XML 树到文件	372
5.6.4	界面的数据绑定	310	6.6	综合实例：生成并输出 RSS 聚合 内容	373
5.7	小结	315	6.6.1	数据类	375
第 6 章	用于 XML 访问的 LINQ (LINQ to XML)	317	6.6.2	建立 RSS 对象	376
6.1	LINQ to XML 概述	317	6.6.3	控制器类	381
6.2	LINQ to XML 的基本操作	319	6.6.4	界面	382
6.2.1	创建 XML 文档	319	6.6.5	运行结果	383
6.2.2	创建 XML 元素	320	6.7	小结	385
6.2.3	创建 XML 属性	321	第 7 章	综合应用实例：商品管理解决 方案	386
6.2.4	创建 XML 注释	322	7.1	概述	386
6.2.5	创建 XML 声明	323	7.2	数据库设计	387
6.2.6	创建 XML 文档类型	324	7.3	使用 LINQ to SQL 的数据层、商品 管理模型	389
6.2.7	创建 XML 处理指令	325	7.3.1	建立数据层	390
6.2.8	创建 XML CData 数据	326			
6.2.9	创建 XML 命名空间	327			
6.3	使用轴方法处理 XML 元素	328			
6.3.1	XElement 类	328			
6.3.2	检索元素的轴方法	331			
6.3.3	添加元素的轴方法	339			

7.3.2 收银台及商店模型	396	7.5.1 自定义 Web 控件	418
7.4 WinForm 项目	406	7.5.2 商品查询、分页显示页面	419
7.4.1 收银台界面自定义控件	406	7.5.3 Webservice 数据耦合层	425
7.4.2 控制台窗体	411	7.6 小结	427
7.5 ASP.NET 项目	418		

第 1 章 LINQ 介绍

本章将简要介绍 LINQ 的一些概念,并通过实例讲解 LINQ 的三大组件,它们是 LINQ to Objects、LINQ to ADO.NET 和 LINQ to XML。

在演示例子中,多处用到 C#语言的新特性,读者可以体验到这些新特性和 LINQ 技术给编程方式带来的巨大变革。

本章的实例都有充分、细致的开发步骤,并且对用到 LINQ 技术的部分代码都做了标注,读者可以根据自己的情况选择阅读。

1.1 什么是 LINQ

LINQ 的英文全称是 Language Integrated Query,中文翻译为“语言集成查询”。LINQ 作为一种查询技术,首先要解决封装数据源的问题,在.NET 框架库中,大致使用三大组件来实现这个封装,分别是 LINQ to Objects、LINQ to ADO.NET、LINQ to XML。

它们与.NET 语言的关系如图 1.1 所示。

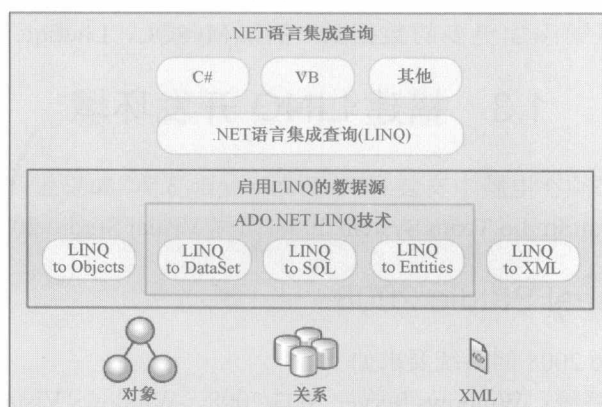


图 1.1 LINQ 和.NET 语言的关系

LINQ 在实际编程中大致是下面这种方式:

```
1 // 建立一个内存数组作为数据源
2 string[] values = { "肖青璇", "董巧巧", "萧玉霜", "秦仙儿", "洛凝" };
3 // 检索字数是2的名字
4 var value = from v in values where v.Length == 2 select v;
5 //输出查询到的结果
6 foreach (var n in value)
7     Console.WriteLine(n);
```

这是一个常见的 LINQ 表达式,它用来查询字数等于2的名字

第 2 行使用 C#语言的新特性——数组集合初始化器，初始化了一个字符串数组。

第 4 行使用 LINQ 技术对内存数组对象进行查询，其中 `var` 关键字也是 C#语言的一个新特性，叫作“局部隐藏类型变量”。

第 6~7 行用迭代器 `foreach` 语句，遍历查询结果并输出显示。

1.2 为什么要用 LINQ

在以前，看到“查询”这个词，一般首先想到的是关系型数据库的查询，为了使用数据库，则必须学习 SQL 查询语言，在编程的时候常用字符串来拼凑 SQL 命令，这样做是非常不安全的，它给 SQL 注入攻击提供了土壤，而开发环境 IDE 也不能对 SQL 命令提供智能感应提示(IntelliSense)，所以书写 SQL 命令是一件很枯燥的事情。

SQL 命令在编程和编译时刻一般是不会有错误检测的，只能在运行时刻，靠数据库系统给出有限的调试信息进行调试，所以调试出错的 SQL 命令更是一件非常痛苦的事情。至于编程查询 XML 文档和内存中的集合对象则更是非常繁琐和效率低下。

LINQ 则是为了解决上述编程中经常遇到的问题而出现的，它在对象领域和数据领域之间架起了一座桥梁。

LINQ 使用开发者熟悉的各种语言关键字和运算符，针对 SQL Server、XML 文档、内存中支持 `IEnumerable` 或泛型 `IEnumerable<T>` 接口的任意对象集合进行查询，并且还可以用 .NET 提供的扩展框架添加更多的数据源，例如 MySQL、LiteSql、DB2 等。

1.3 搭建 LINQ 开发环境

要使用 LINQ，必须在电脑中安装 .NET Framework 3.5，如果想开发 LINQ 程序，则推荐安装 Microsoft Visual Studio Team System 2008(简称 Visual Studio 2008)。

1.3.1 安装 Visual Studio 2008

安装 Visual Studio 2008 的系统要求如下。

- 支持的操作系统：Windows Server 2003/2008、Windows Vista、Windows XP。
- 最低配置：CPU 主频为 1.6GHz、内存容量为 384MB，显示分辨率为 1024×768，硬盘转速为 5400r/min。
- 建议配置：2.2GHz 或更快的 CPU，1024MB 或更大的内存，1280×1024 显示分辨率，7200r/min 或更快的硬盘。在 Windows Vista 上运行的配置要求为 2.4GHz 的 CPU、768MB 的内存。

Microsoft Visual Studio Team System 2008 可以从微软的网站下载 90 天的全功能试用版，安装试用版以后，可以根据需要输入购买的序列号，直接升级成为正式版。

下载之后，使用虚拟光驱或刻录成 DVD 光盘进行安装，安装过程如下。

(1) 插入光盘，安装程序自动运行或双击光盘中的“`Autorun.exe`”文件手动运行安装程序后，出现如图 1.2 所示的安装对话框。

(2) 单击“安装 Visual Studio 2008”链接，开始安装，出现如图 1.3 所示的界面。



图 1.2 Visual Studio 2008 安装对话框

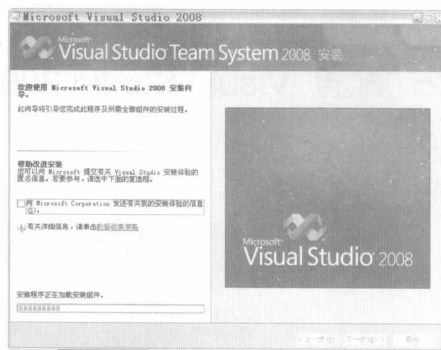


图 1.3 开始安装 Visual Studio 2008

(3) 安装程序检测后, 出现如图 1.4 所示的界面, 阅读许可条款后, 选中“我已阅读并接受许可条款”, 然后单击“下一步”按钮, 出现如图 1.5 所示的界面。

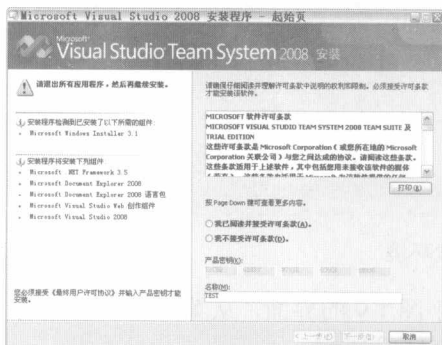


图 1.4 许可条款界面

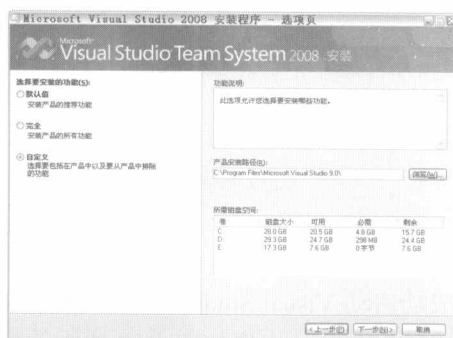


图 1.5 安装程序选项

(4) 选中“自定义”单选按钮, 如果有必要的话可以指定一个安装路径, 之后可以单击“下一步”按钮, 出现如图 1.6 所示的界面。

(5) 根据需要选择安装项目, 如果想运行本书中的实例, 则必须安装“Visual C#”和“Visual Web Developer”及“Microsoft SQL Server 2005 Express Edition”, 选择完毕后, 单击“安装”按钮, 出现如图 1.7 所示的界面后, 安装程序正式开始安装 Visual Studio 2008。

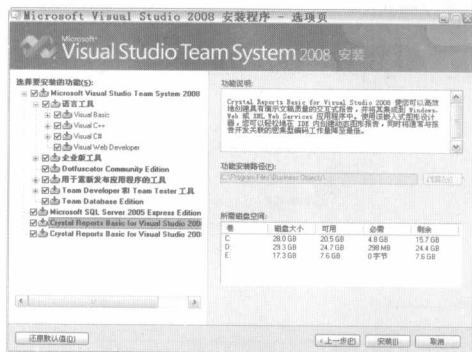


图 1.6 详细选择安装项目

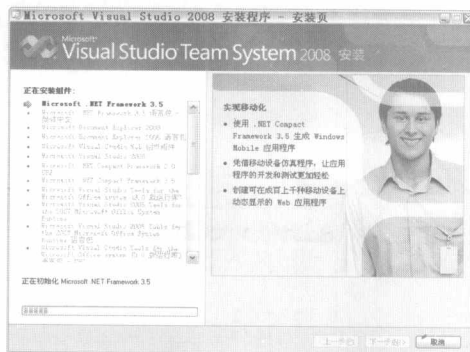


图 1.7 开始安装 Visual Studio 2008

(6) 安装全部项目后, 出现“完成页”, 单击“完成”按钮结束安装。

1.3.2 配置 Visual Studio 2008

第一次运行 Visual Studio 2008 时, 出现如图 1.8 所示的选择开发环境的对话框, 选择“Visual C# 开发设置”后, 单击“启动 Visual Studio”按钮。

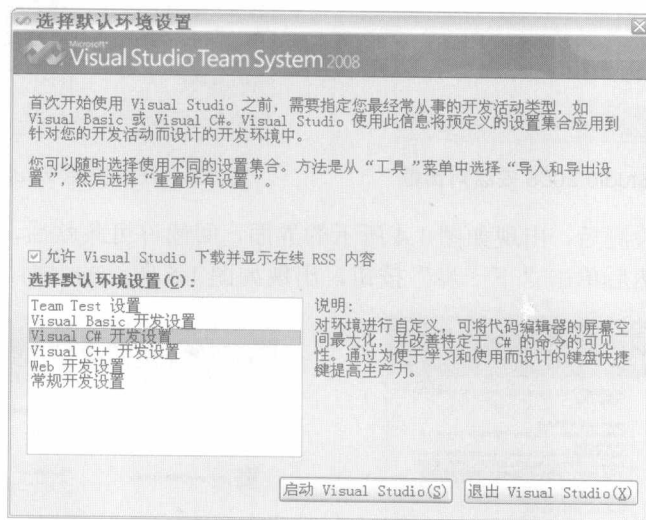


图 1.8 选择开发环境

成功启动 Visual Studio 2008 之后, 出现类似于图 1.9 的界面, 从菜单栏中选择“工具”→“选项”命令, 出现如图 1.10 所示的设置开发环境对话框。

针对图 1.10, 选择“文本编辑器”节点中的“所有语言”项目, 在右侧的面板中选择“自动换行”复选框并选择“行号”复选框, 最后单击“确定”按钮完成配置。

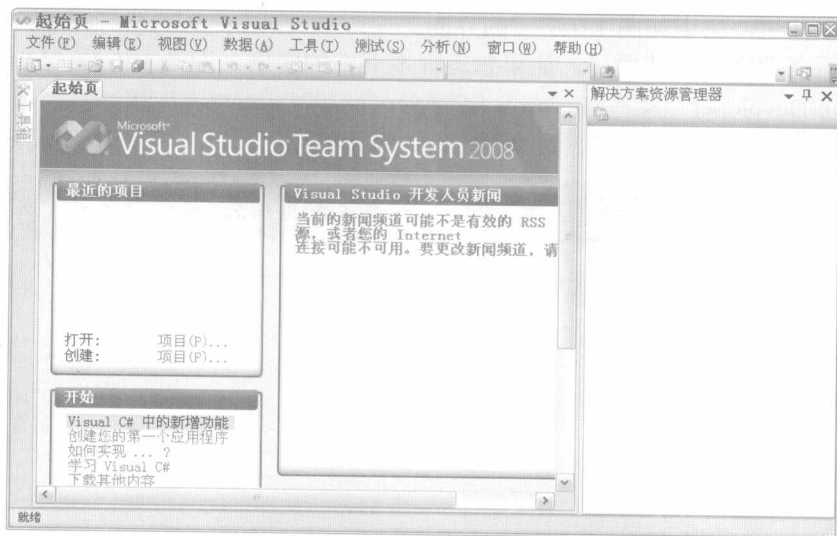


图 1.9 Visual Studio 2008 开发环境

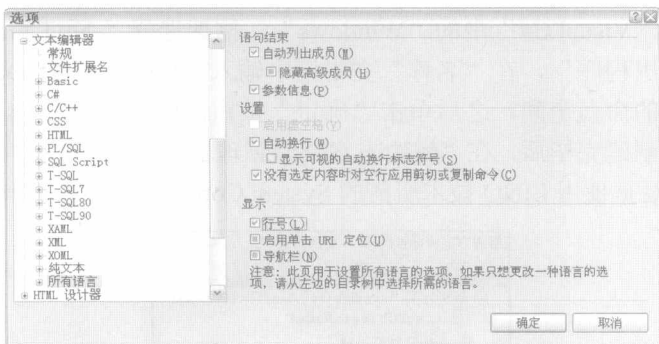


图 1.10 设置开发环境

1.4 初识 LINQ 技术

本节主要是快速地浏览一下 LINQ 的三大组件，增强对 LINQ 技术的感性认识，代码看不看得懂不重要，只要能体会到 LINQ 到底能干什么就可以了。

1.4.1 初识 LINQ to Objects

在编程中要经常接触和使用数组、集合等对象，LINQ to Objects 可以对实现 IEnumerable 或泛型 IEnumerable<T>接口的集合、数组对象进行查询。

在.NET 框架中，几乎所有的集合或数组对象都已经全部或部分实现了上述两个接口，可以方便地在程序中的任何位置使用 LINQ 来查询这些对象的元素。

在没有 LINQ 以前，要查询一个集合中的元素，一般使用 foreach 循环，用遍历集合元素的方法来查询，而 LINQ 提供了一个全新的方式。

实例 1：传统集合查询和 LINQ 集合查询对比

启动 Visual Studio 2008 后，从菜单栏中选择“文件”→“新建项目”命令，将会出现如图 1.11 所示的对话框。

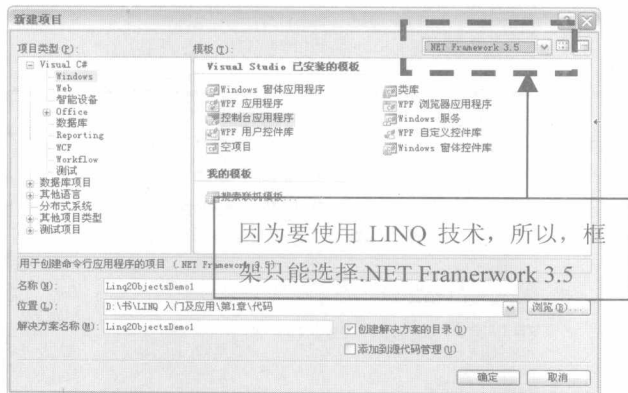


图 1.11 新建项目

项目类型选择“Visual C#”中的“Windows”，框架选择“.NET Framework 3.5”，模板选择“控制台应用程序”，在“名称”文本框中输入“Linq2ObjectsDemol”，这个名称也是程序代码默认的命名空间，之后单击“确定”按钮，即可创建项目。

项目按照模板创建完毕后，在“解决方案资源管理器”中打开“引用”节点，如图 1.12 所示，可以看到已经把使用 LINQ 技术所需的 System.Core 程序集引用进来了。

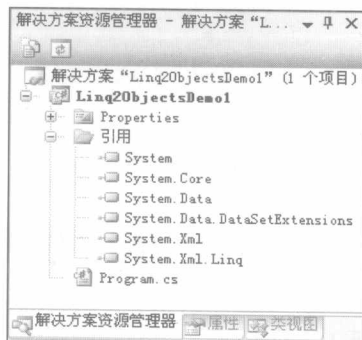


图 1.12 项目引用的程序集

在“解决方案资源管理器”中打开“Program.cs”文件，输入如下的代码：

```

1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Collections;

6 namespace Linq2ObjectsDemol
7 {
8     class Program
9     {
10         static void Main(string[] args)
11         {
12             //定义字符串数组
13             string[] friends = { "霓裳公主", "萧家大小姐", "出云公主", "苗族圣姑", "高丽公主" };
14             Console.WriteLine("按照传统的方法查找\n");

15             //初始化一个新的集合，用于存储后面查询到的元素
16             ArrayList values = new ArrayList();
17             //使用传统的方法查找字符串包含“公主”的词组。
18             //遍历数组把符合条件的元素放入 values 集合中
19             foreach (string word in friends)
20             {
21                 if (word.IndexOf("公主") > -1)
22                     values.Add(word);
23             }
24             //对结果进行排序
25             values.Sort();

```

引用 System.Core 程序集后，可以添加对命名空间的引用，方便 LINQ 子句在编程时的使用

传统的检索、排序方法

```

26      //遍历并输出查询结果
27      foreach (string w in values)
28          Console.WriteLine(w);

29      Console.WriteLine("\n-----\n");

30      Console.WriteLine("使用 LINQ 方法查询\n");

31      //使用 LINQ 技术查找包含“公主”的词组，并对结果进行排序
32      var value = from v in friends
                   where v.IndexOf("公主") > -1
                   orderby v
                   select v;

33      //遍历并输出查询结果
34      foreach (var w in value)
35          Console.WriteLine(w);

36      //为了方便查看运行结果，等待用户按键后结束程序
37      Console.ReadKey();

38  }
39  }
40  }

```

使用 LINQ 查询和排序。这是一条 LINQ 指令，为了方便查看，分成了 4 行显示

通过上述演示代码，可以看到传统的集合查询方法大概需要 4 到 5 个步骤的算法组合，才能完成对集合的查询和排序工作，如果还需要对集合进行分组等操作，算法会更加复杂，代码的出错概率会提高，可读性进一步变差。

使用 LINQ 技术后，查询、排序整合成一条指令，并且查询和排序只是 LINQ 中常用的两个子句，还有更多的子句可以现实分组、汇总、变造等操作。如果读者对 SQL 语句有所了解的话，会发现 LINQ 的查询语句有似曾相识的感觉。

在书写 LINQ 语句的时候，IDE 提供实时编译的语法检查，如果手误，出现写错变量名等问题，IDE 会温柔地画出一条波浪线，如图 1.13 所示，作为错误警告，这就有效地降低了出错概率。

```

//使用LINQ技术查找包含“公主”的词组,并对结果进行排序。
var value = from v in friends
             where error IndexOf("公主") > -1
             orderby v
             select v;

```

图 1.13 波浪线的出错提示

此外，如图 1.14 所示，如影相随的智能感应提示极大地提高了编程效率，并且避免了输入错误，使 LINQ 编程成为一种乐趣。

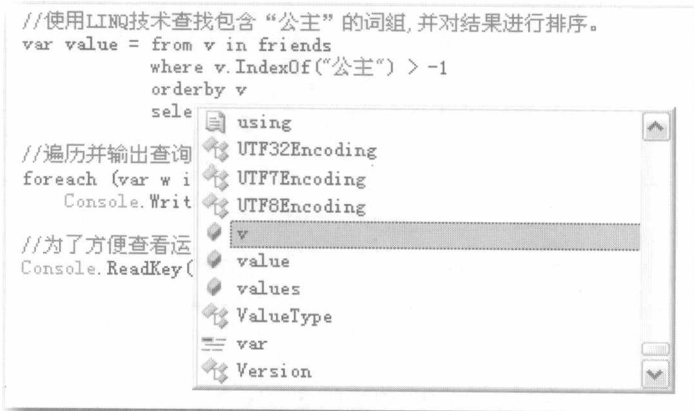


图 1.14 书写 LINQ 时的智能感应提示

从 Visual Studio 2008 的菜单栏中,选择“调试”→“启动调试”命令,运行这个实例,得到如图 1.15 所示的结果。

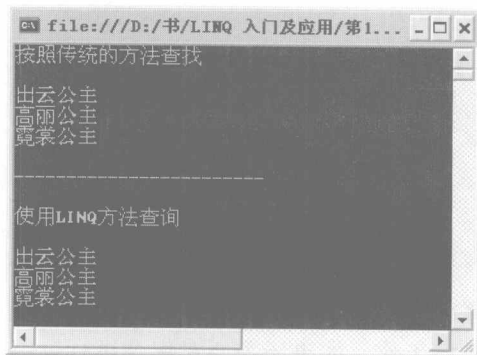


图 1.15 对比传统查询和 LINQ 查询

1.4.2 初识 LINQ to ADO.NET

用 LINQ 技术查询关系型数据库,需要使用 LINQ to ADO.NET,它还细分为 LINQ to SQL、LINQ to DataSet 以及在 Visual Studio 2008 发布后开发的 LINQ to Entities:

- LINQ to SQL 是通过使用开发者建立的一个映射 SQL Server 中的数据表,被称作实体类的类型,来完成对数据库的查询、修改、删除等操作。
- LINQ to DataSet 用来查询 DataSet 数据集或从数据库断开连接的数据。
- LINQ to Entities 的用途跟 LINQ to SQL 类似,但比 LINQ to SQL 功能更加复杂和强大,例如,在一个实体类中映射多个数据库表格和它们之间的关系,用 LINQ 查询 Microsoft SQL Server 系列产品以外的关系数据库等。

由于 LINQ to Entities 需要安装后续发布的 ServerPack 才能使用,所以将不在本书中介绍。

下面我们通过实例来讲解 LINQ to SQL。