

C# 程序设计

北京科海 总策划

关玉英 主 编

- 从实际应用的角度出发，帮助读者以最快的速度进入C#的世界，提高程序开发技术水平
- 资深讲师精心编写，融入作者多年的教学经验
- 结构清晰、实例丰富，具有很强的操作性和实用性



1CD 大型多媒体教学系统



- 全程语音讲解的多媒体学习环境
- 超大容量，播放时间长达263分钟



吉林电子出版社

北京科海技术图书

www.khp.com.cn

高等院校规划教材·计算机系列

C#程序设计

关玉英 主 编

吉林电子出版社

北京科海培中技术有限责任公司

内 容 简 介

本书囊括了 C# 2.0 的基础知识、语法介绍、各类型应用程序的开发、数据库应用、XML 应用等内容。书中不仅包括 Windows 应用程序的开发，还包括 Windows 控制台应用程序和 ASP.NET 网络应用程序开发的介绍。

本书内容精炼、实例丰富，介绍了大量的知识点和程序实例，使读者从一开始就能把握应用程序开发的主线。本书以实践动手能力为主要出发点，对每一个实例都进行了讲解。

此外，为扩大读者的知识面，以及更好地掌握程序设计的方法和技巧，除书中讲解的内容之外，我们在光盘中补充了大量基于 Visual C++ 的拓展演示视频，以及书中全部实例的程序源代码，以期使读者能够在本书内容的基础上学到更多的知识，且提升应用技能。

本书既可作为高等院校相关课程教材，又可作为 C# 2.0 的初学者和各类院校的技术参考书。

编 者：关玉英

出 版 人：王保华

责任编辑：孟迎红 汤 洁 李晶璞

封面设计：林 陶

出版发行：吉林电子出版社

地 址：长春市人民大街 4646 号（邮编：130021）

印 刷：北京市艺辉印刷有限公司

开 本：787×1092 1/16

字 数：481 千字

印 张：19.75

版 次：2009 年 6 月第 1 版

印 次：2009 年 6 月第 1 次印刷

版 号：ISBN 978-7-89454-049-2

定 价：32.00 元（1 多媒体教学 CD+1 配套手册）

前言

C# 是微软公司 .NET 框架中的一种高级编程语言，该语言不仅支持控制台应用程序开发，还支持 Windows 窗体应用程序以及 ASP.NET 应用程序的开发。由于 C# 基于微软公司的 .NET 框架支持，因而拥有良好的灵活性、可扩展性，非常利于各种应用程序的开发。

与其他同类图书相比，本书具有如下优点：

(1) 实例丰富。本书介绍了 C# 中大量的知识点，几乎每个知识点都配有一个难易适中、长短适宜的实例。每个实例的最后都给出了运行结果示意图和代码分析，便于读者的理解。

(2) 结构合理。本书不仅为广大的初学者提供了翔实知识介绍，而且在结构的编排上力求合理。C# 的知识点非常丰富，类库也异常庞大，如何选取内容进行介绍以及如何进行内容的组织则更为重要。本书力求在结构上满足读者的需要。

(3) 内容精炼。本书在全面介绍 C# 主要知识点的基础上注意控制内容的深度，以初学者的需求为出发点，介绍了他们最需要掌握的知识。以实际的程序开发为主要目标，摒弃了一些对初学者无益的内容。

本书所介绍的知识点紧扣实用性的主题，考虑到读者学习的难度，主要介绍了 C#2.0 中读者最需要学习的知识，并合理地安排了本书的结构，便于读者上手。

本书从如何创建项目出发，逐渐介绍了各种复杂的应用，并在书中合适的位置介绍了一些作者常用的技巧、编程的建议以及部分常用的工具软件，使读者能够得到全面的锻炼，对于初学者而言是非常有益的。

本书全面介绍了 C#2.0 的基本语法、控制台应用程序的开发、Windows 应用程序的开发以及 Web 应用程序的开发。书中介绍了大部分常用的 .NET Framework 库，使读者对 C#2.0 应用程序的开发有一个全面的了解。在 Windows 窗体应用程序开发和 Web 应用程序开发中，向读者介绍了几乎所有的常用控件，便于读者快速掌握 C#2.0 应用程序开发的技巧。

本书以 C#2.0 为依托，介绍了目前常见的正则表达式技术、XML 技术以及数据库开发技术。这些技术在多数的应用程序中都有应用，熟练掌握这部分技术对于读者编程水平的提升有很大帮助。本书的后半部分介绍了数据库技术和 Windows 窗体应用程序的结合以及数据库技术和 Web 应用程序的结合使用，并给出了实例代码，还介绍了应用程序打包部署方面的知识，全面讲解了应用程序开发的全过程。

由于时间仓促与编者水平有限，不足与欠妥之处在所难免，恳请广大读者不吝指正。

编者

2009 年 6 月

目 录

第1章 C#与.NET框架简介	1
1.1 C#的概念	1
1.1.1 C#与.NET框架	1
1.1.2 .NET框架与其他语言的比较	2
1.1.3 C#的功能	3
1.1.4 安装Visual Studio 2005	4
1.2 第一个C#程序: Hello World	6
1.2.1 配置和认识Visual Studio 2005开发环境	6
1.2.2 第一个Windows控制台应用程序	8
1.2.3 第一个Windows窗体应用程序	11
1.2.4 第一个ASP.NET应用程序	14
1.3 小结与提高	15
1.4 思考与练习	15
第2章 C#程序设计基础	16
2.1 数据类型	16
2.1.1 值类型	16
2.1.2 引用类型	17
2.2 变量和常量	17
2.2.1 命名变量	17
2.2.2 声明和使用变量	19
2.2.3 类型转换	22
2.2.4 声明和使用常量	29
2.3 运算符	31
2.3.1 算术运算符	31
2.3.2 字符串运算符	33
2.3.3 赋值运算符	34
2.3.4 逻辑运算符	36
2.3.5 比较运算符	38
2.3.6 运算符的优先级	40
2.4 流程控制	40
2.4.1 使用if语句实现分支	41
2.4.2 使用switch语句实现分支	43
2.4.3 使用while语句实现循环	45

2.4.4	使用for语句实现循环	47
2.4.5	使用break/continue控制循环	48
2.5	程序的结构	50
2.5.1	用类搭建程序	50
2.5.2	使用命名空间	51
2.5.3	寻找程序的入口	51
2.6	异常处理	53
2.7	小结与提高	54
2.8	思考与练习	55
第3章	字符串操作	56
3.1	静态字符串String	56
3.1.1	比较字符串	56
3.1.2	搜索字符和子串	58
3.1.3	格式化字符串	59
3.1.4	连接多个字符串	60
3.1.5	分割字符串	61
3.1.6	插入字符串	62
3.1.7	删除字符串	63
3.1.8	复制字符串	64
3.1.9	替换字符串	65
3.1.10	大小写转换	66
3.2	动态字符串StringBuilder	66
3.2.1	声明StringBuilder变量	67
3.2.2	StringBuilder的容量	67
3.2.3	追加字符串	68
3.2.4	插入操作	69
3.2.5	删除部分字符	70
3.2.6	替换部分字符	71
3.2.7	StringBuilder与String的比较	72
3.3	使用正则表达式	73
3.3.1	正则表达式简介	73
3.3.2	使用Regex类	75
3.3.3	常用正则表达式	76
3.3.4	验证电子邮箱地址	76
3.4	小结与提高	78
3.5	思考与练习	78
第4章	数组	79
4.1	C#数组简介	79

4.1.1	数组的概念	79
4.1.2	多维数组	80
4.1.3	创建C#数组	81
4.2	使用数组	83
4.2.1	Array类	83
4.2.2	访问数组元素	85
4.2.3	遍历数组	86
4.2.4	数组排序	90
4.2.5	查找元素	93
4.2.6	逆序数组	95
4.2.7	复制数组	96
4.2.8	动态创建数组	98
4.2.9	清除数组元素	99
4.2.10	数组应用实例	100
4.3	小结与提高	101
4.4	思考与练习	102
第5章	集合	103
5.1	System.Collections命名空间	103
5.2	ArrayList类	104
5.2.1	创建ArrayList	104
5.2.2	添加元素	106
5.2.3	插入元素	107
5.2.4	删除元素	108
5.2.5	简单排序	112
5.2.6	查找元素	113
5.2.7	遍历所有元素	114
5.2.8	转换为数组	116
5.3	Queue类	117
5.3.1	创建队列	117
5.3.2	元素入队	118
5.3.3	元素出队	119
5.4	Stack类	120
5.4.1	创建堆栈	120
5.4.2	元素入栈	121
5.4.3	元素出栈	122
5.5	Hashtable类和SortedList类	123
5.5.1	构造普通哈希表	123
5.5.2	哈希表操作	124

5.5.3 排序哈希表SortedList	126
5.5.4 排序哈希表操作	126
5.6 小结与提高	128
5.7 思考与练习	128
第6章 C#面向对象	129
6.1 定义类	129
6.1.1 声明类	129
6.1.2 声明类的静态特征	129
6.1.3 声明类的动态行为	130
6.1.4 创建类实例	131
6.1.5 继承类	132
6.2 Visual Studio 2005中的OOP工具	134
6.2.1 使用类视图面板	134
6.2.2 使用对象浏览器	134
6.3 字段和属性	135
6.3.1 字段	135
6.3.2 定义属性	137
6.3.3 只读属性	140
6.3.4 只写属性	141
6.4 方法	143
6.4.1 方法的参数	143
6.4.2 使用构造函数	144
6.4.3 使用析构函数	146
6.4.4 使用静态方法	147
6.4.5 重载方法	149
6.4.6 访问父类方法	150
6.5 抽象类	151
6.5.1 抽象类简介	152
6.5.2 声明抽象类	152
6.6 接口	153
6.6.1 什么是接口	153
6.6.2 声明接口	154
6.6.3 实现接口	155
6.7 小结与提高	157
6.8 思考与练习	157
第7章 Windows窗体应用程序	158
7.1 按钮控件	158
7.2 标签控件	160

7.3	文本框控件.....	161
7.4	单选按钮控件.....	162
7.5	复选框控件.....	164
7.6	列表框.....	166
7.7	可选列表框控件.....	168
7.8	菜单.....	171
7.9	编写多文档界面应用程序.....	173
7.10	小结与提高.....	175
7.11	思考与练习.....	176
第8章	GDI+图形.....	177
8.1	第一个绘图程序.....	177
8.2	创建Graphics对象.....	179
8.3	绘制线条.....	183
8.4	绘制形状.....	188
8.5	用GDI+呈现图像.....	191
8.6	画点.....	192
8.7	坐标曲线.....	193
8.8	小结与提高.....	195
8.9	思考与练习.....	195
第9章	文件操作.....	196
9.1	文件.....	196
9.1.1	System.IO命名空间.....	196
9.1.2	文件类File.....	198
9.1.3	文件夹类Directory.....	199
9.1.4	文件信息类FileInfo.....	200
9.1.5	文件夹信息类DirectoryInfo.....	201
9.1.6	文件信息类与文件夹信息类的用法.....	202
9.2	流.....	203
9.2.1	流操作类介绍.....	203
9.2.2	文件流类FileStream.....	203
9.2.3	流写入类StreamWriter.....	205
9.2.4	流读取类StreamReader.....	207
9.2.5	二进制流写入类BinaryWriter.....	208
9.2.6	二进制流读取类BinaryReader.....	209
9.3	小结与提高.....	210
9.4	思考与练习.....	210
第10章	数据库操作.....	211
10.1	ADO.NET概述.....	211

10.1.1	微软数据访问技术	211
10.1.2	System.Data命名空间	212
10.2	连接数据库	212
10.2.1	创建数据库	212
10.2.2	SqlConnection类	220
10.2.3	创建SQL Server连接	220
10.2.4	断开SQL Server连接	221
10.2.5	管理其他数据源连接	223
10.3	与数据库交互	224
10.3.1	使用SqlCommand提交添加数据命令	224
10.3.2	使用SqlCommand提交修改命令	226
10.3.3	使用SqlCommand提交删除命令	228
10.3.4	使用SqlCommand获取查询命令	230
10.4	小结与提高	233
10.5	思考与练习	233
第11章	XML编程	234
11.1	.NET框架中的XML	234
11.1.1	理解XML	234
11.1.2	XML文档的特点	235
11.1.3	System.Xml命名空间	236
11.2	读取XML文档	236
11.2.1	使用XmlReader类	236
11.2.2	读取元素值	240
11.2.3	读取元素的属性	244
11.3	写入XML文档	246
11.3.1	使用XMLWriter类	247
11.3.2	写入属性	248
11.3.3	写入元素	251
11.4	小结与提高	255
11.5	思考与练习	255
第12章	ASP.NET网络开发	256
12.1	什么是ASP.NET	256
12.1.1	ASP.NET的诞生	256
12.1.2	IIS管理ASPX页面	257
12.1.3	ASP.NET的优点	258
12.2	ASP.NET语法	259
12.2.1	剖析ASPX页面	259
12.2.2	使用<% %>嵌入代码	261

12.2.3	使用<Script>...</Script>嵌入代码	262
12.2.4	使用Server控件	264
12.2.5	使用<%--注释--%>	264
12.2.6	用<%@ Page...%>设置页面属性	265
12.2.7	使用<%@ Import %>引入类库	265
12.3	常用控件	265
12.3.1	标签控件Label	265
12.3.2	文本框控件TextBox	266
12.3.3	按钮控件Button	266
12.3.4	下拉框控件DropDownList	268
12.3.5	列表框控件ListBox	269
12.3.6	复选框控件CheckBox	271
12.4	小结与提高	273
12.5	思考与练习	273
第13章	数据绑定技术	274
13.1	数据绑定概述	274
13.2	列表控件	275
13.2.1	将数据绑定到下拉列表控件	275
13.2.2	将数据绑定到列表框	279
13.2.3	将数据绑定到可选列表框	280
13.3	数据网格: DataGridView	282
13.3.1	用DataGridView自动显示数据	282
13.3.2	用DataGridView导航数据	283
13.4	小结与提高	286
13.5	思考与练习	286
第14章	应用程序部署	287
14.1	部署工程的建立	287
14.1.1	Windows Installer介绍	287
14.1.2	Visual Studio 2005向导	289
14.2	规划安装程序	293
14.2.1	安装目录	293
14.2.2	快捷方式	293
14.2.3	依赖项	294
14.2.4	文件类型	294
14.3	制作安装程序	294
14.3.1	用户界面编辑器	294
14.3.2	欢迎界面	295
14.3.3	安装目录	296

14.3.4	确认安装	297
14.3.5	安装进度	297
14.3.6	完成界面	298
14.3.7	程序的删除	298
14.3.8	添加快捷方式	299
14.3.9	添加文件	299
14.3.10	注册表编辑器	300
14.3.11	文件类型编辑器	301
14.3.12	自定义操作编辑器	301
14.3.13	启动条件编辑器	302
14.4	小结与提高	302
14.5	思考与练习	302
主要参考文献		303

第 1 章

C#与.NET 框架简介

本章将讨论进行C#编程的基础知识。在本章里，读者可以对C#的概念有一个全面的了解，对C#及其相关技术有一定的认识，并在此基础上认识进行C#编程开发的主要环境Visual Studio 2005以及进行简单的编程和应用。

本章主要内容

- C#的概念
- .NET框架的概念
- C#与.NET框架的关系
- C#的特点
- Visual Studio 2005的使用
- 简单的C#应用

1.1 C#的概念

C#与.NET 框架有着紧密的联系。本部分内容将涉及 C#的概念、.NET 框架的概念、C#与.NET 框架的关系以及 C#的特点等。C#是一门崭新的编程语言，可以满足大多数应用场景的需要，能编写出用户体验丰富、功能强大的应用程序。C#一经推出就受到了广大程序开发人员和用户的好评，成为.NET 框架中最重要的语言。

1.1.1 C#与.NET框架

.NET 框架是微软公司推出的一个全新的编程平台，目前的版本是 2.0。此平台支持本书中所要介绍的 C#。.NET 框架还支持许多其他的语言，如 VB.NET，Visual C++.NET 等。应用.NET 框架可以满足应用程序开发的大部分需要。由于其具有强大的功能特性和方便易用性，.NET 框架已成为越来越多公司、机构的开发工具。

C#是微软公司为.NET 框架全新设计的一门语言，包含很多丰富的特性，为程序开发带来了极大的方便。C#是微软公司在参考了之前比较流行的编程语言后，才专门提出的，C#几乎囊括了所有编程语言的优点。因此，C#有很广阔的应用前景。.NET 框架的基本结构如图 1-1 所示。

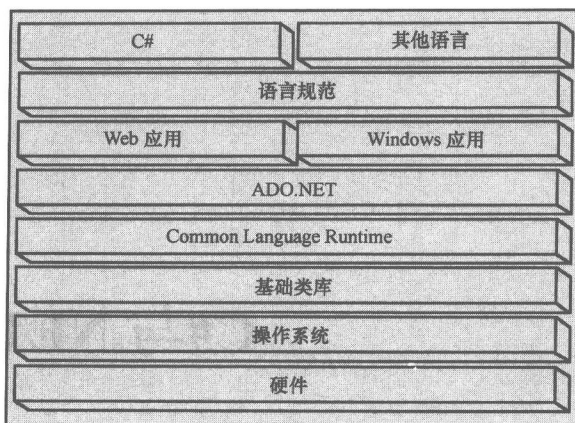


图 1-1 .NET 框架基本结构

在图 1-1 中，除去操作系统和硬件两部分，就是 .NET 框架的大部分结构。

1.1.2 .NET 框架与其他语言的比较

由于 .NET 框架的设计借鉴了许多其他编程语言的优点，因此在某些方面类似于 Java、Delphi 和 Visual Basic 等编程语言。

在运行机制上，.NET 框架非常类似于 Java 语言。与 Java 虚拟机相对应，.NET 框架中的通用语言运行时 (CLR) 就有类似的作用。.NET 框架中的编程与其他高级语言有所不同，在编译的时候并不是直接生成本机可直接运行的代码，而是生成了一种名为微软中间语言 (Microsoft Intermediate Language MSIL) 的代码。在运行时，再由 CLR 将其转换为与平台相关的代码。由于 .NET 框架这种特殊的机制，为其带来了一些先进的特性。

1. 平台无关

.NET 框架编写出来的代码与平台无关，只要平台上支持 .NET 框架的运行时，就可以直接在该平台上运行代码。

2. 性能提升

在 .NET 框架的运行机制中，尽管与 Java 语言类似，但是还有很大的不同。在 CLR 进行第二步编译时，.NET 框架采用了一种名为即时编译 (JIT) 的技术。与 Java 语言的解释不同，.NET 框架并不是在一开始就将全部的代码编译为本机代码，而是当某部分代码被调用时编译该部分代码。一旦编译成功，在下次被调用时也无需再次编译。由此就性能上而言，.NET 框架在采用类似虚拟机技术的语言中是有优势的。

3. 语言互操作性

由于 MSIL 的存在，使得 .NET 框架下语言之间的互操作性成为可能。在 .NET 框架支持的语言中，各种语言在第一步编译时实际上都被编译成 MSIL 代码。而 MSIL 代码是不存在语言差别的，因此使得语言之间的互相调用得以实现。这点在某些方面有很重要的应

用，比如开发人员可以任意选用自己熟悉的编程语言，它们之间的代码共享和相互调用不会存在任何障碍。

.NET 框架中默认提供了以下几种常见语言的支持：Visual C# 2005、Visual Basic 2005、Visual C++ 2005 和 Visual J# 2005。

这些语言之间的互相调用都可以实现，同样，将来.NET 框架支持的语言之间也可以进行语言之间的互操作。

1.1.3 C#的功能

C#有着非常强大的功能，能满足大部分应用场景的需要。

1. Windows控制台应用程序

Windows 控制台应用程序通常以 DOS 窗体的形式出现。尽管其表现形式十分简单，但却是一种功能非常强大的应用程序类型。控制台应用程序以流的方式输入和输出数据，而不是像 GUI（图形用户接口）应用程序那样使用单击交互模型。控制台应用程序通常被称为命令行应用程序。

2. Windows窗体应用程序

Windows 窗体应用程序是 Windows 操作系统中最为常见的应用程序类型，其最大特点是以用户的操作为主要的响应对象，根据用户的操作进行不同的处理。此类操作主要体现为鼠标的单击和键盘的输入。Windows 窗体应用程序能够带来更好的用户体验和操作感受。

3. ASP.NET应用程序

相对于 Windows 窗体应用程序类型来说，ASP.NET 应用程序则是另外一种完全不同的类型。Windows 窗体应用程序类型的程序一般需要用户在本机安装，进行的是本机的操作。如果有服务端的程序，则称为客户端/服务器（C/S）程序，而 ASP.NET 应用程序全部部署在服务器端，用户只需要一个标准的浏览器即可使用，因此被称为浏览器/服务器（B/S）程序。

4. Windows控件

通常支持 Windows 窗体应用程序开发的各种类型的编程语言都会提供一些标准的控件。C#除了提供这些标准控件以外，还提供了创建自定义控件的可能，既可以对现有控件的功能进行扩充，也可以编写一个全新的控件。编写出的控件既可以在自己的应用程序中使用，也可以作为产品发布。

5. Windows服务

Windows 服务通常是运行在 Windows 操作系统后台的应用程序。此类程序一般没有用户的单击交互操作，没有 Windows 控制台应用程序和 Windows 窗体应用程序类似的界面，但从本质来讲，与 Windows 控制台应用程序和 Windows 窗体应用程序没有区别。

1.1.4 安装 Visual Studio 2005

Visual Studio 2005 中包含最新的 .NET 框架 2.0 版, 因此, 只需进行 Visual Studio 2005 的安装即可获得一个完整的 C# 开发环境。获得 Visual Studio 2005 后, 首先要将其安装到计算机中。本书以 Visual Studio 2005 为例介绍其安装过程。

(1) 将 Visual Studio 2005 光盘放入光盘驱动器, 将会弹出如图 1-2 所示的窗口。



图 1-2 Visual Studio 2005 安装界面 1

(2) 单击“安装 Visual Studio 2005”链接, 进入下一步安装界面, 如图 1-3 所示。

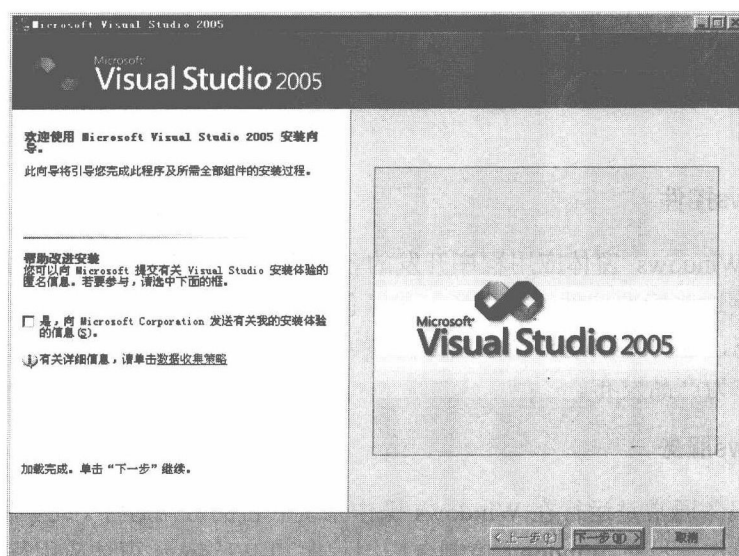


图 1-3 Visual Studio 2005 安装界面 2

(3) 单击“下一步”按钮，进入下一步安装，如图 1-4 所示。这个窗体中包含最终用户许可协议，读者需要同意其所有条款才能继续下一步安装。界面的右下方分别是产品密钥和名称的输入框，可输入相应信息。

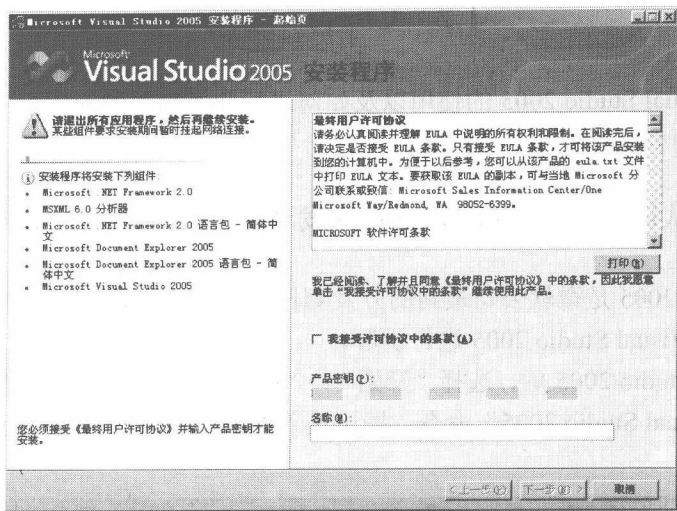


图 1-4 Visual Studio 2005 安装界面 3

(4) 单击“下一步”按钮，出现如图 1-5 所示的提示对话框。



图 1-5 Visual Studio 2005 安装界面 4

(5) 单击“确定”按钮，进入下一步安装界面，如图 1-6 所示。

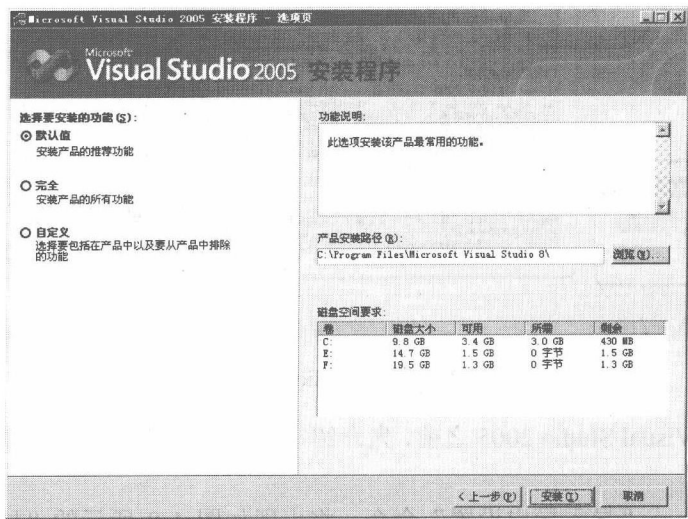


图 1-6 Visual Studio 2005 安装界面 5