

高等学校计算机语言应用教程

Visual C#.NET

应用教程

• 童爱红 编著 • 张琦 主审

1011001001100
100110010010010
1001101001001100
0100110010010010
0110101100101101
0110010010010010
001001101001100100
0011001001001000
110100100110100110
110110100100110100
0010101010101100
1011010010101101
100101010101101
1110101010101101

本书配光盘



清华大学出版社 • 北京交通大学出版社



高等学校计算机语言应用教程

Visual C# .NET 应用教程

童爱红 编著

张琦 主审

清华大学出版社

北京交通大学出版社

• 北京 •

内 容 简 介

Visual C# .NET 是一门完全面向对象的程序设计语言, 具有简单、现代、类型安全、性能优良等特点, 是面向对象程序设计教学的主干语言之一。本书共分 15 章, 全面讲解了面向对象的程序设计概念、Visual C# 的数据类型和表达式、Visual C# 的程序设计语句、数组、面向对象的程序设计方法与原理、控件窗体与界面设计技术、文件程序设计、数据库程序设计和 Web 程序设计等内容。

本书从教学实践的角度出发, 立足于提高学生的程序设计应用能力, 全书理论分析透彻严谨, 实例丰富生动, 内容由浅入深, 能快速引导学生进入 Visual C# 编程世界。本书可作为高等院校的程序设计课程教材, 也可作为广大希望掌握 Visual C# 编程的程序设计人员的参考用书。

版权所有, 翻印必究。

本书封面贴有清华大学出版社激光防伪标签, 无标签者不得销售。

图书在版编目 (CIP) 数据

Visual C# .NET 应用教程 / 童爱红编著. —北京: 清华大学出版社; 北京交通大学出版社, 2004.11

(高等学校计算机语言应用教程)

ISBN 7-81082-391-4

I. V… II. 童… III. C 语言-程序设计-高等学校-教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2004) 第 085215 号

责任编辑: 谭文芳

出版者: 清华大学出版社 邮编: 100084 电话: 010-62776969

北京交通大学出版社 邮编: 100044 电话: 010-51686045, 62237564

印刷者: 北京东光印刷厂

发行者: 新华书店总店北京发行所

开 本: 185×260 印张: 21.25 字数: 540 千字 附光盘 1 张

版 次: 2004 年 11 月第 1 版 2004 年 11 月第 1 次印刷

书 号: ISBN 7-81082-391-4/TP·144

印 数: 1~5 000 册 定价: 34.00 元 (含光盘)

前 言

“程序设计”是高等院校理工科各专业学生的一门重要基础课程，目前通常使用的面向对象的程序设计语言主要有 Visual C++，Visual Basic，Delphi，Java，Visual C#等。其中 Visual C# .NET 是在 C 和 C++基础上发展起来的（有人称它为可视的 Java），具有简单、现代和类型安全的特点。由于 Visual C# .NET 语言版本较新，市场上可作为教材的书籍并不多，仅有的一些教材也在很多方面存在着不足。在此背景下，我们组织编写了这本《Visual C# .NET 应用教程》，目的是为了引导学生快速高效地进入 Visual C# .NET 编程世界。

本书的作者有多年从事程序设计教学的一线教学经验，对程序设计的教学把握较为独到，能够预料到学生在学习可能遇到的困难并加以解决。同时，本书的作者有教材编写的经验，具有很强的敬业精神，编写的教材有助于提高学生的学习效率。

本书的总体编写思路如下。

1. 全书分 15 章，全面讲解了 Visual C# .NET 程序设计语言的各个部分，特别加强了对数组、方法、高级事件、界面设计、面向对象的程序设计概念、多媒体和数据库等程序设计的重点、难点和具有较强实用价值的内容的讲解、引导和剖析。

2. 每章均分 4 个部分进行编写：“理论知识”部分简明扼要地讲解本章的主要理论，并通过小的实例进行深化理解；“典型实例”部分通过分析一些精心挑选和编制的典型实例，强化学生的编程能力；“上机练习”部分挑选一些具有实用价值的上机练习题，加以剖析并给出部分程序代码，引导学生在上机练习中提高应用能力；“课后考场”部分设计了一套试题，方便学生进行自我测试。

3. 实例引导。本书的每一章都有着丰富的实例，有的实例具有较强的趣味性，易引起学生的兴趣，激发学生对程序设计的喜好。

与现有的教材相比，本教材具有以下特色。

1. 重点难点突出。本书没有罗列大量的语言成分，不介绍较琐碎或不太常用的属性、指令和方法，而是针对 Visual C# .NET 软件的特点较详细地介绍了 Visual C# .NET 的主要语言成分，重点讲述 Visual C# .NET 程序设计的概念和方法。

2. 不为写 Visual C# .NET 而写教材。始终贯彻为写程序设计教材而写教材的思路，Visual C# .NET 只是选择的一门工具语言。因此，本书将重点放在程序设计的基础上和程序设计教材的共性上，而不是仅着重于 Visual C# .NET 的强大功能和使用技巧。本书力争达到这样的目标：通过本书的学习使学生能够掌握程序设计的概貌，进入程序设计的大门。

3. 在编写风格上注重学生动手编程能力的培养。针对学生普遍认为程序设计语言难学的特点，本书不再讲解高深难懂的理论，而是强调通过实例学编程。通过精选有趣的实例，讲解实例的实现过程，激发学生的编程兴趣，引导学生一步一步地步入程序设计的大门。

本书由童爱红具体编写，博士生导师张琦教授对全书进行了审阅并最终定稿。

本书的配套光盘包含全书的所有例题、习题源代码和可执行文件，所有的程序都在

Windows XP 平台和 Visual Studio .NET 环境下调试通过并经过严格测试。另外，本书配套的电子教案可以在北京交通大学出版社网站 <http://press.bjtu.edu.cn> 上下载。

在本书的编写过程中，得到了解放军理工大学计算机与指挥自动化学院黄松副教授的指导和帮助，得到了解放军理工大学工程兵工程学院计算机应用教研室全体老师的指导与帮助，在此表示衷心的感谢。同时对参加本书资料收集、程序测试和文稿校对的胡光兵、侯太平、汪刚等同志表示衷心的感谢。编者参阅了大量文献资料及网站资料，在此对这些资料的所有者也一并表示感谢。

虽然我们力求完美，力创精品，但由于水平有限，书中难免有疏漏和错误等不尽人意之处，还请广大读者不吝赐教。

编 者
2004 年 9 月

目 录

第 1 章 Visual C# .NET 编程简介	1
1.1 理论知识	1
1.1.1 .NET 简介	1
1.1.2 Visual C# .NET 简介	2
1.1.3 Visual Studio .NET 集成开发环境简介	3
1.1.4 创建第一个 C# Windows 程序	7
1.1.5 创建第一个 C# 控制台（或称命令行）程序	9
1.1.6 C# 程序结构介绍	10
1.2 典型实例	11
1.3 上机练习	12
课后考场	13
第 2 章 基本数据类型与表达式	15
2.1 理论知识	15
2.1.1 Visual C# .NET 的标识符	15
2.1.2 Visual C# .NET 的数据类型	15
2.1.3 简单数据类型	16
2.1.4 常量与变量	18
2.1.5 运算符与表达式	22
2.1.6 常用方法	30
2.2 典型实例	31
2.3 上机练习	32
课后考场	33
第 3 章 结构化程序设计语句	35
3.1 理论知识	35
3.1.1 if 语句	35
3.1.2 switch 语句	38
3.1.3 while 循环语句	40
3.1.4 do...while 循环语句	42
3.1.5 for 循环语句	43
3.1.6 break 和 continue 语句	45
3.1.7 随机数的产生方法	46
3.2 典型实例	48
3.2.1 典型实例一	48

3.2.2 典型实例二	49
3.3 上机练习	50
3.3.1 上机练习一	50
3.3.2 上机练习二	51
课后考场	52
第4章 数组	55
4.1 理论知识	55
4.1.1 数组的概念	55
4.1.2 一维数组的定义、分配与使用	55
4.1.3 多维数组的定义、分配与使用	59
4.1.4 foreach 语句	63
4.2 典型实例	64
4.2.1 典型实例一	64
4.2.2 典型实例二	66
4.3 上机练习	67
4.3.1 上机练习一	67
4.3.2 上机练习二	69
课后考场	70
第5章 方法	72
5.1 理论知识	72
5.1.1 C#中的程序模块——方法	72
5.1.2 方法的定义	73
5.1.3 方法的调用	74
5.1.4 参数传递中的类型转换	76
5.1.5 参数的传递：值传递、引用传递和输出参数	77
5.1.6 变量的生命期和作用域	79
5.1.7 方法的嵌套与递归调用	81
5.2 典型实例	84
5.2.1 典型实例一	84
5.2.2 典型实例二	85
5.3 上机练习	87
5.3.1 上机练习一	87
5.3.2 上机练习二	88
课后考场	89
第6章 其他常用数据类型	91
6.1 理论知识	91
6.1.1 字符串类型	91
6.1.2 结构类型	96
6.1.3 枚举类型	98
6.1.4 委托类型	100

6.1.5	装箱与拆箱	101
6.2	典型实例	102
6.3	上机练习	103
	课后考场	104
第 7 章	面向对象的程序设计	107
7.1	理论知识	107
7.1.1	面向对象程序设计概述	107
7.1.2	类和对象的声明	109
7.1.3	类的构造函数和析构函数	113
7.1.4	类的方法及方法的重载	114
7.1.5	运算符重载	118
7.1.6	域与属性	120
7.1.7	this 关键字	122
7.1.8	类的继承	122
7.1.9	多态性	123
7.1.10	密封类与抽象类	125
7.2	典型实例	125
7.2.1	典型实例一	125
7.2.2	典型实例二	127
7.3	上机练习	129
7.3.1	上机练习一	129
7.3.2	上机练习二	131
	课后考场	133
第 8 章	异常处理	135
8.1	理论知识	135
8.1.1	异常处理概念	135
8.1.2	捕获处理异常	136
8.1.3	C#的异常类	139
8.1.4	抛出和重发异常	140
8.1.5	checked 和 unchecked 运算符的用法	143
8.2	典型实例	145
8.3	上机练习	146
	课后考场	147
第 9 章	Windows 应用程序设计	149
9.1	理论知识	149
9.1.1	Visual C#开发 Windows 应用程序的方法	149
9.1.2	窗体	151
9.1.3	文本类控件的使用	155
9.1.4	按钮类控件的使用	165
9.1.5	列表类控件的使用	169

9.1.6 PictureBox 控件的使用	174
9.1.7 Timer 控件的使用	175
9.1.8 ProgressBar 控件和 TrackBar 控件的使用	177
9.1.9 HScrollBar 控件和 VScrollBar 控件的使用	180
9.2 典型实例	180
9.2.1 典型实例一	180
9.2.2 典型实例二	182
9.3 上机练习	183
9.3.1 上机练习一	183
9.3.2 上机练习二	185
课后考场	186
第 10 章 Windows 高级界面设计	188
10.1 理论知识	188
10.1.1 对话框控件的应用	188
10.1.2 菜单控件	194
10.1.3 多窗体程序设计	197
10.1.4 MDI 应用程序设计	198
10.1.5 工具栏设计	203
10.1.6 状态栏设计	205
10.1.7 键盘事件处理	207
10.1.8 鼠标事件处理	208
10.2 典型实例	209
10.3 上机练习	211
课后考场	212
第 11 章 GDI+与图形编程	214
11.1 理论知识	214
11.1.1 GDI+与绘图命名空间	214
11.1.2 Graphics 对象	215
11.1.3 Pen 对象	216
11.1.4 Font 对象	217
11.1.5 Brush 对象	218
11.1.6 常用图形的绘制方法	220
11.2 典型实例	228
11.2.1 典型实例一	228
11.2.2 典型实例二	229
11.3 上机练习	231
课后考场	232
第 12 章 多媒体应用程序开发	234
12.1 理论知识	234
12.1.1 多媒体的概念	234

12.1.2	图像的基本处理	235
12.1.3	Media Player 控件及其使用	240
12.1.4	MMControl 控件及其使用	242
12.1.5	ShockwaveFlash 控件及其使用	247
12.2	典型实例	249
12.3	上机练习	251
	课后考场	252
第 13 章	文件	254
13.1	理论知识	254
13.1.1	文件的相关概念	254
13.1.2	文件和流	256
13.1.3	目录管理	256
13.1.4	文件管理	258
13.1.5	文本文件的读写	266
13.1.6	二进制文件的读写	268
13.2	典型实例	271
13.3	上机练习	274
	课后考场	275
第 14 章	简单数据库编程	277
14.1	理论知识	277
14.1.1	数据库的基本概念	277
14.1.2	ADO.NET 概述	279
14.1.3	SQL 语言	281
14.1.4	ADO.NET 对象及其编程	283
14.1.5	利用 ADO.NET 控件编写数据库应用程序	292
14.2	典型实例	299
14.3	上机练习	303
	课后考场	305
第 15 章	Web 应用程序开发	307
15.1	理论知识	307
15.1.1	Web 应用程序介绍	307
15.1.2	Web 窗体设计	310
15.1.3	Web 服务的创建和使用	313
15.2	典型实例	321
15.3	上机练习	322
	课后考场	324
附录 A	C#关键字一览表	326
参考文献		328

第 1 章 Visual C# .NET 编程简介

本章要点

- ☒ .NET 概念及其组成
 - ☒ Visual C# .NET 的产生和特点
 - ☒ Visual Studio .NET 集成开发环境
 - ☒ 创建 Visual C# .NET 程序的一般方法
 - ☒ C#程序结构
-

1.1 理论知识

1.1.1 .NET 简介

1. .NET 的定义

.NET 技术是微软公司推出的一个全新概念，它代表了一个集合、一个环境和一个可以作为平台支持下一代 Internet 的可编程结构。.NET 的目的就是将互联网作为新一代操作系统的基础，对互联网的设计思想进行扩展，用户在任何地方、任何时间，以及利用任何设备都能访问所需的信息、文件和程序。用户不需要知道这些文件放在什么地方，只需要发出请求，然后就可以接受处理的结果，后台的复杂处理过程对用户而言是透明的。

2. .NET 开发平台

.NET 开发平台如图 1-1 所示。

.NET 开发平台包括.NET 框架和.NET 开发工具等组成部分，.NET 框架（Framework）是整个开发平台的基础，包括公共语言运行库和框架类库，.NET 开发工具包括 Visual Studio .NET 集成开发环境和.NET 编程语言。其中，Visual Studio .NET 集成开发环境用来开发和测试应用程序。.NET 编程语言包括 Visual Basic、Visual C++和新的 Visual C#等用来创建运行在公共语言运行库（Common Language Run，CLR）上的应用程序。

3. .NET 框架

.NET 框架的组成如图 1-2 所示。

.NET 框架包括公共语言运行库和.NET 类库。公共语言运行库是.NET 的基础，用户可以将公共语言运行库看做一个在执行时管理代码的代理，它提供核心服务（如内存管理、线程管理和远程处理），而且还强制实施严格的类型安全检查，以确保代码运行的安全性和可靠性。事实上，代码管理的概念是运行库的基本原则。以运行库为目标的代码称为托管代码，而不以运行库为目标的代码称为非托管代码。.NET 框架的另一个主要部件是类库，它是一

个综合性的面向对象的可重用类型集合，用户使用它不仅可以开发传统的命令行或图形用户界面（GUI）应用程序，而且还可以开发 ASP .NET 所提供的创新的应用程序（如 Web 窗体和 XML Web 服务）。

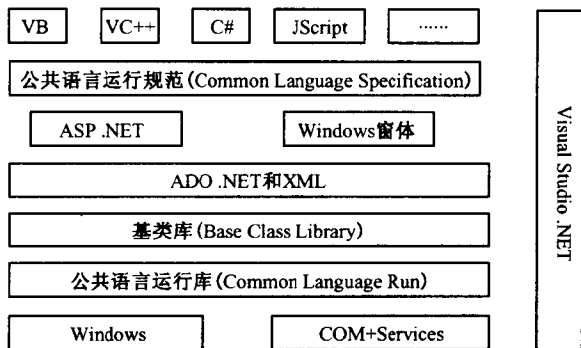


图 1-1 .NET 开发平台

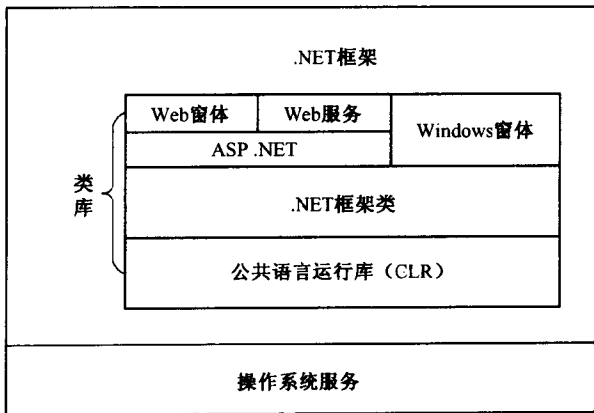


图 1-2 .NET 框架

1.1.2 Visual C# .NET 简介

1. Visual C# .NET 语言的产生背景

1995 年，Sun 公司正式推出面向对象的开发语言 Java，并提出跨平台、跨语言的概念，之后，Java 就逐渐成为企业级应用系统开发的首选工具。

为支持 Java 语言，微软开发出基于 Java 语言的编译器 Visual J++，并在很短的时间里由 1.1 版本升级到 6.0 版本，其中 Visual J++ 6.0 版集成在 Visual Studio 6.0 中。该语言开发的程序不但在 Java 虚拟机上（Java Virtual Machine, JVM）的运行速度大大加快，而且增加了许多新特性，同时还支持调用 Windows API，这些特性使得 Visual J++成为强有力的 Windows 应用开发平台，并成为业界公认的优秀 Java 编译器。

由于 Visual J++主要应用在 Windows 平台的系统开发中，因此 Sun 公司认为 Visual J++违反了 Java 的许可协议，违反了 Java 开发平台的中立性，并对微软提出了诉讼，这使得微

软处于极为被动的局面。为了改变这种局面，微软决定推出其进军互联网领域的 .NET 计划，C# 就是该计划中所要开发的一门重要的开发语言。

.NET 计划将是微软今后几年发展的战略核心，其内容非常庞大。它的技术开发平台是 Visual Studio .NET，而 C# 作为 Visual J++ 的替代语言也集成在该平台中。

C# 语言的开发成功，对微软未来的发展有着非常重要的意义。微软在 2000 年 6 月份举行的“职业开发人员技术大会”上正式发布了 C# 语言。微软公司对 C# 的定义是：“C# 是一种类型安全的、现代的、简单的、由 C 和 C++ 衍生出来的面向对象的编程语言，它是牢牢根植于 C 和 C++ 语言之上的，并可立即被 C 和 C++ 开发人员所熟悉。C# 的目的就是综合 Visual Basic 的高生产率和 C++ 的行动力。”

2. Visual C# .NET 的特点

与 C 和 C++ 相比，C# 具有以下特点：

- (1) 语法更简单；
- (2) 保留了 C++ 的强大功能；
- (3) 快速应用开发功能；
- (4) 语言的自由性；
- (5) 强大的 Web 服务器控件；
- (6) 支持跨平台；
- (7) 与 XML 相融合。

1.1.3 Visual Studio .NET 集成开发环境简介

.NET 开发语言都采用了统一的集成开发环境（Integrated Development Environment，IDE），使用同一个 IDE 为开发者提供了极大的方便。下面全面介绍 Visual Studio .NET 的开发环境。

1. Visual Studio 起始页

启动 Visual Studio .NET 后，首先看到的是一个如图 1-3 所示的起始页。此起始页是集成开发环境中默认的 Web 浏览器主页。它是设置首选选项、读取产品新闻和访问别的在 Visual Studio .NET 环境里启动和运行的信息的集中地。

2. 新建 Visual C# .NET 项目

在 Visual Studio .NET 集成开发环境中，通过执行【文件】→【新建】→【项目】菜单命令，将会弹出【新建项目】对话框，在该对话框中，可以选择不同的编程语言来创建各种项目，这些语言将共享 Visual Studio .NET 的集成开发环境，如图 1-4 所示。

要创建新的 Visual C# .NET 项目，需要在该对话框的【项目类型】窗口中选中【Visual C# 项目】，在【模板】窗口中选中【Windows 应用程序】。然后在【位置】文本框中输入项目保存的位置（路径），在【名称】文本框中输入项目的名称，如图 1-4 所示。然后单击【确定】按钮，将会出现如图 1-5 所示的 Visual Studio .NET 集成开发环境。该开发环境由许多窗口组成，下面将一一介绍这些窗口的功能。

3. 【解决方案资源管理器】窗口

首先应理解解决方案与项目的关系。项目可以视为编译后的一个可执行单元，可以是应用程序、动态链接库等，而企业级的解决方案往往需要多个可执行程序的合作，为便于管理多个项目，在 Visual Studio .NET 集成环境中引入了解决方案资源管理器，用来对企业级解决方案涉及的多个项目进行管理。【解决方案资源管理器】窗口如图 1-6 所示。如果集成环境中没有出现该窗口，可通过执行【视图】→【解决方案资源管理器】命令来显示该窗口。



图 1-6 【解决方案资源管理器】窗口

4. 【类视图】窗口

【类视图】窗口如图 1-7 所示。如果集成环境中没有出现该窗口，可通过执行【视图】→【类视图】命令来显示该窗口。

【类视图】窗口中以树形结构显示了当前项目中的所有类，并在每个类中列出了成员变量和成员函数，每一个类首先列出带有紫色图标的成员函数，然后是带有蓝绿色图标的成员变量。每个成员的图标左边都有一个标志，以表示成员类型和存取类别的信息，保护型成员图标旁边的标志为一把钥匙，私有成员的标志是一把锁，而公有成员图标旁边没有标志。



图 1-7 【类视图】窗口

在【类视图】窗口中双击类名，会在主工作区中打开这个类的头文件，显示出类的声明；而双击某个类的成员，则主工作区中会显示该成员的定义代码。

5. 【工具箱】窗口

【工具箱】窗口如图 1-8 所示。如果集成环境中没有出现该窗口，可通过执行【视图】→【工具箱】命令来显示该窗口。

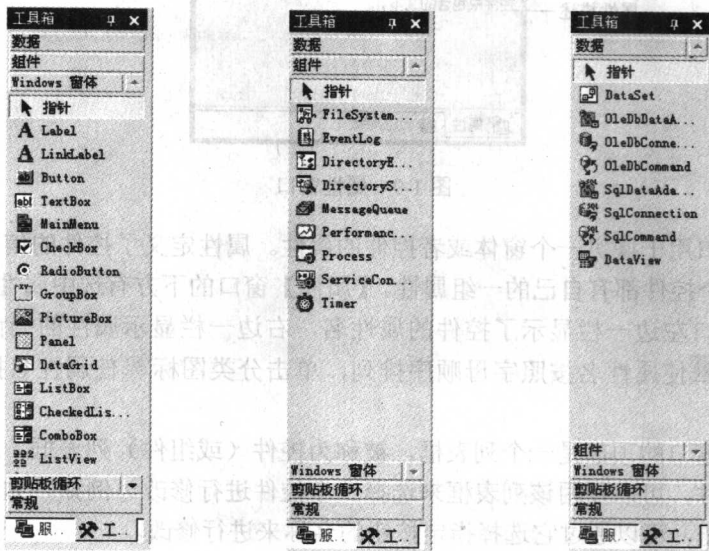


图 1-8 【工具箱】窗口

【工具箱】中包含了可重用的控件（或称组件），用于自定义应用程序。使用可视化的方法编程时，可在窗体中“拖放”控件，绘制出应用程序界面，而不用自己去写代码。

控件在【工具箱】中是以组的形式出现的，如【数据】组、【组件】组和【Windows 窗体】组。通过单击组名称能展开一个组。通过使用【工具箱】右下部的滚动箭头，能滚动浏览所有的控件。组的第一项不是控件，它是鼠标指针，单击它后可以取消对控件的选择，以便重新选择其他控件。

【工具箱】最初可能会隐藏起来，而仅仅是在 IDE 的边上显示出窗口名称。移动鼠标指针到窗口名字上能打开这个窗口，鼠标指针移出窗口时窗口会消失，这是自动隐藏特性。要关掉【工具箱】的自动隐藏功能，单击在窗口右上角的大头针图标，再次单击将又打开【工具箱】的自动隐藏功能。

注意：当打开自动隐藏功能时，大头针指到了旁边，这在其他窗口的隐藏中也是一样的。

6. 【属性】窗口

【属性】窗口如图 1-9 所示。如果集成环境中没有出现该窗口，可通过执行【视图】→【属性】命令来显示该窗口。



图 1-9 属性窗口

【属性】窗口用于操纵一个窗体或者控件的属性。属性定义了控件的信息，如大小、颜色和位置等。每个控件都有自己的一组属性。【属性】窗口的下方有选中属性的描述。

【属性】窗口左边一栏显示了控件的属性名，右边一栏显示属性的当前值。可以单击按字母排序图标使属性名按照字母顺序排列，单击分类图标使属性名按照分类顺序排列。

在【属性】窗口的顶部是一个列表框，被称为控件（或组件）列表框。此列表框显示当前正在修改的控件，可以使用该列表框来选择一个控件进行修改。例如，如果一个图形用户界面包含几个按钮，可以通过它选择指定按钮的名称来进行修改。

1.1.4 创建第一个 C# Windows 程序

【例 1-1】 编写一个 C# Windows 应用程序，程序运行时显示一句欢迎词“欢迎您进入 C#编程世界!”程序的运行界面如图 1-10 所示。

实现步骤如下：

(1) 启动 Visual Studio .NET。在 Windows XP 中的启动方法是：执行【开始】→【所有程序】→【Microsoft Visual Studio .NET】→【Microsoft Visual Studio .NET】菜单命令，如图 1-11 所示。

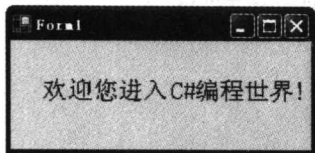


图 1-10 例 1-1 程序运行界面



图 1-11 Visual Studio .NET 启动菜单

(2) 执行【文件】→【新建】→【项目】命令，在弹出的【新建项目】对话框的【项目类型】列表中选择【Visual C#项目】，在【模板】列表框中选择【Windows 应用程序】，在【位置】文本框中输入项目保存位置“D:\C#App\01\A”，在【名称】文本框中输入“A_1_1”，然后单击【确定】按钮，出现 Visual Studio .NET 的集成开发环境。

(3) 在【工具箱】中单击【Windows 窗体】工具组，用鼠标单击 Label 控件工具 **A Label**，然后在窗体的适当位置按下鼠标左键并拖动，将会在窗体上生成一个标签对象，如图 1-12 所示。

(4) 在窗体上单击 Label1 控件，在【属性】窗口中选中文本属性，设置它的属性值为“欢迎您进入 C#编程世界!”。再选中 Font 属性，设置它的属性值为“宋体，14.25pt”。然后再选中“ForeColor”属性，设置它的属性值为“ActiveCaption”。设计完成的界面如图 1-13 所示。

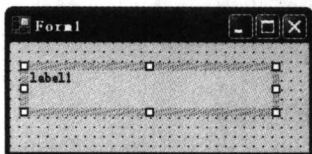


图 1-12 创建的 Label1 控件

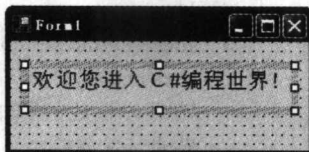


图 1-13 设计完成后的程序界面

(5) 在窗体上单击鼠标右键，在出现的快捷菜单中选择【查看代码】命令（或执行【视图】→【代码】命令），切换到代码视图，可以看到生成的 C#程序代码，如下所示。

```
using System;
using System.Drawing;
using System.Collections;
using System.ComponentModel;
using System.Windows.Forms;
using System.Data;
```