

return

郭文夷 戴芳胜 编著

Visual C#.NET 可视化程序设计

2C
-2



华东理工大学出版社

EAST CHINA UNIVERSITY OF SCIENCE AND TECHNOLOGY PRESS

Visual C# .NET 可视化程序设计

郭文夷 戴芳胜 编著

 华东理工大学出版社

图书在版编目(CIP)数据

Visual C#.NET 可视化程序设计/郭文夷,戴芳胜编
著. —上海:华东理工大学出版社,2005.2
ISBN 7-5628-1653-0

I. V... II. ①郭... ②戴... 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(2005)第 001897 号

Visual C#.NET 可视化程序设计

郭文夷 戴芳胜 编著

出版	华东理工大学出版社	开本	787×1092 1/16
社址	上海市梅陇路 130 号	印张	14.5
邮政	邮编 200237 电话(021)64250306	字数	351 千字
网址	www.hdlgpress.com.cn	版次	2005 年 2 月第 1 版
发行	新华书店上海发行所	印次	2005 年 2 月第 1 次
印刷	常熟市华顺印刷有限公司	印数	1-3050 册

ISBN 7-5628-1653-0/TP·131

定价:23.50 元



前言

PREFACE

Microsoft 的 .NET 平台是一个全新的开发框架,它具有与 Windows 服务和 API 整合的全新编程接口,集成了多种核心的技术和服务。

Microsoft C# (C Sharp)语言是一种新的编程语言,它是专门为生成运行在 .NET 框架上的企业级应用程序而设计的。C#吸取了 C、C++、Java、Delphi 等语言的一些优点,是一种简单、现代、类型安全和面向对象的语言。在 .NET 平台上,C#代码被作为托管代码编译,这意味着它能够从公共语言运行库(CLR)的服务中受益。这些服务包括:语言互操作性、垃圾回收、增强的安全性以及改进的版本支持等。

虽然 .NET 平台具有与编程语言无关的特征,但除非已习惯使用某种特定的语言(如 VB)或需要特定兼容性的场合,C#必然是 .NET 平台上首选的编程语言。当前国内理工科专业学生一般都学过 C 或 C++,在此基础上再学 C#是十分明智的选择。

Microsoft Visual Studio .NET 不仅是 Visual Studio 的升级版本,而且是第一个基于 Microsoft 的 .NET 框架的可视化开发环境。Visual Studio .NET 开发环境的易学程度与 VB、Delphi 等传统开发工具差不多,却可以开发出强大、高效和统一的企业级应用程序。

目前有关 .NET 框架和 C#的书籍中,大部分把重点放在 C#的语法或 .NET 的完整框架上,因此不能适合初学者。尤其缺少结合 Visual Studio .NET 开发环境介绍 C#可视化设计的入门教材。由此造成许多人想学习 C#和 .NET 编程却不知从何处下手。我们希望本书的出版能弥补这个缺憾。

本书作者长期从事 C、C++和 C#的开发和教学实践,对该语言和相关开发环境有较全面的了解和独到的见识,并积累了丰富的教学经验。本书兼顾易学性和实用性,语言生动简明。采用循序渐进、逐步深入的教学方法引导初学者掌握 C#的基本语法、面向对象编程的基本概念和在 Visual Studio .NET 下使用 .NET 各种常用组件的能力。书中侧重于用例子来解释概念和示范操作,并且每一章都提供了若干难度适中的习题。

本书正文包含 7 章的内容:第 1 章介绍 Visual Studio .NET 开发环境;第 2 章介绍 C#的基本语法;第 3 章介绍 C#类和对象、接口、委托和事件等概念和语法;第 4 章介绍数组、列表、队列等 C#类型和文件操作;第 5 章介绍窗体、控件的使用,对话框等 C#可视化开发技术;第 6 章介绍数据库程序设计,包括数据库的连接与操作、数据缓存、数据绑定等技术;第 7 章介绍图形、图像和多媒体程序设计,包括视频和音频的播放、Flash 动画播放等技术。

本书可作为大专院校计算机类专业本、专科相关课程的教材,特别适合高职高专有关专业作为“可视化程序设计”课程的教科书。此外,本书也可作为对 .NET 编程框架和 C#语言感兴趣的理工科学生和一般科技工作者的自学读本。

建议本书的教学时数为 50~60,选学第 1、2、5 章的全部内容和第 3、4、6、7 章的部分内容。如果学习全部内容,大约需要 80 学时。



本书主编是郭文夷、戴芳胜。仓晓磊也参加了部分章节的编写工作。由于作者的水平有限,加上时间仓促,书中若有不妥之处,恳请行家和读者批评指正。

为了方便教师备课,凡订购本书作为教材使用的,我们将提供免费的多媒体课件。联系本书作者的网址为:<http://202.121.241.111:3000>,该网址即将开通,欢迎各位访问并参与交流。

编 者

2004 年 12 月



目 录

CONTENTS

第 1 章 集成开发环境的使用	(1)
1.1 .NET 平台简介	(1)
1.2 C# 语言的特点	(1)
1.3 VS. NET 2003 集成开发环境的介绍	(2)
1.3.1 集成开发环境的组成	(3)
1.3.2 集成开发环境的菜单功能	(5)
1.3.3 集成开发环境的选项设置.....	(12)
1.3.4 使用 VS. NET 2003 开发 C# 工程项目	(13)
1.3.5 使用 VS. NET 2003 创建控制台模式的 Hello World Windows 应用程序	(18)
1.3.6 VS. NET 2003 的安装	(20)
1.4 Microsoft MSDN. NET 帮助系统	(22)
1.4.1 MSDN 联机帮助的使用	(22)
1.4.2 动态帮助的使用.....	(23)
1.4.3 VS. NET Framework SDK 帮助的使用	(24)
习题 1	(25)
第 2 章 C# 语言基础	(26)
2.1 C# 程序的基本结构	(26)
2.2 C# 中的变量和常量	(27)
2.2.1 变量.....	(27)
2.2.2 常量.....	(27)
2.3 C# 中的基本数据类型	(28)
2.3.1 整数类型.....	(28)
2.3.2 布尔类型.....	(29)
2.3.3 实数类型.....	(29)
2.3.4 字符类型.....	(29)
2.3.5 枚举类型.....	(30)
2.3.6 数据类型转换.....	(31)
2.4 值类型和引用类型.....	(32)
2.4.1 值类型.....	(32)
2.4.2 引用类型.....	(33)

2.4.3 堆栈和堆.....	(33)
2.5 C#中的运算符与表达式	(33)
2.5.1 C#中的运算符类型	(33)
2.5.2 关系运算符.....	(33)
2.5.3 逻辑运算符.....	(34)
2.5.4 算术运算符.....	(35)
2.5.5 位运算符.....	(35)
2.5.6 赋值运算符.....	(36)
2.5.7 条件运算符.....	(37)
2.5.8 运算符的优先级.....	(37)
2.6 控制语句.....	(38)
2.6.1 条件语句.....	(38)
2.6.2 循环语句.....	(40)
2.6.3 跳转语句.....	(43)
2.7 异常处理.....	(44)
2.7.1 使用异常处理.....	(44)
2.7.2 抛出异常.....	(46)
2.8 其他特点.....	(47)
2.8.1 C#中的注释	(47)
2.8.2 命令行参数.....	(49)
习题 2	(49)
第3章 C#面向对象基础	(50)
3.1 类和对象.....	(50)
3.1.1 类的定义和对象的创建.....	(50)
3.1.2 类的可访问性和作用域.....	(50)
3.1.3 类属性的定义.....	(51)
3.1.4 类方法的定义.....	(53)
3.1.5 类的构造函数.....	(56)
3.1.6 类的静态成员.....	(57)
3.2 类的继承性和多态性.....	(58)
3.2.1 类的继承性.....	(58)
3.2.2 Protected 访问修饰符	(59)
3.2.3 密封类.....	(60)
3.2.4 调用基类构造函数.....	(61)
3.2.5 类的多态性.....	(62)
3.2.6 抽象方法和抽象类.....	(64)
3.3 接口.....	(66)
3.3.1 接口的定义.....	(66)



3.3.2 接口的使用.....	(66)
3.3.3 继承多个接口.....	(69)
3.3.4 接口与抽象类的比较.....	(72)
3.4 委托和事件.....	(72)
3.4.1 委托.....	(72)
3.4.2 事件.....	(76)
3.5 名字空间.....	(80)
习题 3	(82)
第 4 章 C# 中常用的集合类和文件操作类	(83)
4.1 常用的集合类.....	(83)
4.1.1 数组.....	(83)
4.1.2 列表.....	(89)
4.1.3 队列.....	(91)
4.1.4 堆栈.....	(94)
4.2 常用的文件操作类.....	(95)
4.2.1 文件流类的简介.....	(95)
4.2.2 操纵文本文件.....	(98)
4.2.3 操纵二进制文件	(102)
4.3 文件和文件夹操作	(103)
4.3.1 文件类的简介	(104)
4.3.2 文件夹类的简介	(105)
习题 4	(107)
第 5 章 Windows 应用程序开发	(108)
5.1 窗体和主菜单的设计	(108)
5.1.1 简单窗体的生成	(108)
5.1.2 窗体的常用属性	(109)
5.1.3 窗体的主要事件和方法	(111)
5.1.4 在窗口中加入主菜单	(112)
5.1.5 多文档窗体的设计	(117)
5.2 基本控件的使用	(117)
5.2.1 标签控件	(118)
5.2.2 按钮控件	(119)
5.2.3 文本框控件	(119)
5.2.4 列表框控件	(124)
5.2.5 组合框控件	(127)
5.2.6 单选与复选按钮控件	(128)
5.2.7 工具栏控件	(131)

5.2.8 状态栏控件	(134)
5.3 对话框	(137)
5.3.1 通用对话框的使用	(137)
5.3.2 自定义对话框的设计	(141)
习题 5	(141)
第 6 章 数据库程序设计	(144)
6.1 Microsoft 数据访问策略的发展	(144)
6.1.1 从 ODBC、OLE DB 到 ADO .NET	(144)
6.1.2 使用数据向导	(145)
6.2 数据库的连接与操作	(148)
6.2.1 Connection 组件	(148)
6.2.2 设置 ConnectionString	(149)
6.2.3 SQL 语言与 Command 组件	(151)
6.2.4 调用 ExecuteNonQuery 方法	(155)
6.2.5 调用存储过程	(157)
6.2.6 操作 OleDb 数据库	(159)
6.3 数据缓存与数据集	(161)
6.3.1 DataSet 和 DataTable 组件	(161)
6.3.2 数据集的应用	(165)
6.4 数据绑定技术	(170)
6.4.1 .NET 控件的数据绑定	(170)
6.4.2 DataGrid 控件	(172)
6.4.3 设计数据表单	(175)
习题 6	(179)
第 7 章 多媒体程序设计	(181)
7.1 图形与图像	(181)
7.1.1 Graphics 类的使用	(181)
7.1.2 Image 类和 Bitmap 类的使用	(192)
7.1.3 PictureBox 控件的使用	(200)
7.2 播放视频音频和动画	(202)
7.2.1 播放视频和音频	(202)
7.2.2 播放 Flash 动画	(208)
习题 7	(211)
附录 1 缩略语及常用词汇表	(214)
附录 2 关键字一览表	(220)
附录 3 ASCII 码表	(221)



第 1 章

集成开发环境的使用

Visual Studio .NET 2003 集成开发环境中包含了大量新奇元素,提供了对多语言混合开发、Web Services 开发、嵌入式设备应用开发等的支持。该集成开发环境还继承了大量人性化的功能和实用的工具,为开发人员提供了一个便捷的开发环境。

1.1 .NET 平台简介

.NET 平台是一个全新的开发框架,它具有与 Windows 服务和 API 合作的全新编程接口,集成了多种核心的技术和服务。

1. 开发工具

Microsoft Visual Studio .NET 和 .NET 框架为开发人员提供了一套完整的解决方案。Visual Studio .NET 提供了一个能够基于 .NET 框架来创建应用程序的开发环境,该开发环境提供了一些工具和向导用以简化应用程序创建和进行可视化应用程序开发。

2. 专用服务器

.NET Enterprise Server 系列通过提供对 Web Service 的支持,提高了系统、应用程序以及其他合作伙伴的集成速度。Microsoft Host Integration Server 提供了对大型机的简单访问,Microsoft BizTalk Server 则提供了现有数据格式和 XML 数据格式之间的自动转换功能。另外,Enterprise Server 还包括 Microsoft Exchange Server、Microsoft SQL Server、Microsoft Commerce Server、Microsoft Content Management Server 等。

3. Web 服务

通过使用 Web 服务可以更方便地进行企业级的系统集成,应用程序可以方便地调用其他应用程序所创建的 Web 服务而无需考虑被调方的操作系统类型、平台类型、开发语言等。

4. 设备

设备包括个人计算机、便携式计算机、工作站、手持计算机(PDA)、智能手机、Tablet PC 等。通过访问 Web Services 数据,这些智能化设备可以无需考虑正在使用设备的类型、种类、数量以及所处的位置。

1.2 C# 语言的特点

Microsoft C# (C Sharp)语言是一种新的编程语言,它是为生成运行在 .NET Framework 上的企业级应用程序而设计的。C# 从 Microsoft C 和 Microsoft C++ 演变而来,它是一种简单、现代、类型安全和面向对象的语言。与 C++ 代码不同,C# 代码被作为托管代码编译,这意味着它能够从公共语言运行库(CLR)的服务中受益。这些服务包括:语言互



操作性、垃圾回收、增强的安全性以及改进的版本支持。表 1-1 给出 C# 与 C++ 的最重要差异。

表 1-1 C# 与 C++ 的差异

	C#	C++
继承	只支持单继承,但类或接口可以实现多个接口	支持多继承,不支持接口
数组	声明数组时“[]”标记出现在数组类型的后面	声明数组时“[]”标记出现在数组名的后面
bool 类型	bool 类型和其他类型(特别是 int)之间没有转换	bool 类型和其他类型(特别是 int)之间有转换
long 类型	long 数据类型为 64 位	long 数据类型为 32 位
struct 类型	类和结构在语义上不同。struct 是值类型,而 class 是引用类型	类和结构在语义上相同。struct 和 class 都是引用类型
switch 语句	不支持从一个 case 标签贯穿到另一个 case 标签	支持从一个 case 标签贯穿到另一个 case 标签
delegate 类型	委托与 C++ 中的函数指针基本相似,但由于其属于托管代码,所以具有类型安全,是安全的	由于属于非托管代码,所以不具有类型安全
C# 运算符	C# 除支持 C++ 原有的大部分运算符外,还支持其他运算符,如 is 和 typeof。此外它还引入了某些逻辑运算符的不同功能	
方法参数	支持 ref 和 out 参数,这两个参数取代指针通过引用传递参数。方法参数不能有默认值,如果要获得同样的效果,请使用方法重载	使用指针传递参数。支持方法参数使用默认值
全局方法和全局变量	没有全局方法和全局变量,方法和变量必须包含在类型声明(如 class 或 struct)中	有全局方法和全局变量,方法和变量可以写在类外
局部变量	局部变量在初始化前不能使用	局部变量在初始化前可以使用
析构函数	不能控制析构函数的调用时间,原因是析构函数由垃圾回收器自动调用	
异常处理	支持 finally 语句	不支持 finally 语句

1.3 VS. NET 2003 集成开发环境的介绍

微软在 2002 年发布了基于 .NET Framework 1.0 的 Visual Studio .NET 2002,之后在吸取了大量用户反馈意见后,在 2003 年推出了趋于稳定的基于 .NET Framework 1.1 的 Visual Studio .NET 2003。Visual Studio .NET 2003 是一套完整的开发工具,用于生成桌



面应用程序、ASP.NET Web 应用程序、XML Web Services 和移动应用程序。与以往的开发工具不同,这次的 VS.NET 2003 中 Visual Basic.NET、Visual C++ .NET、Visual C#.NET 和 Visual J#.NET 全部都使用相同的集成开发环境 (IDE),该环境允许它们共享工具并有助于创建混合语言解决方案。另外,这些语言利用了 .NET Framework 的功能,此框架提供对简化 Web 应用程序和 XML Web services 开发的关键技术的支持。

1.3.1 集成开发环境的组成

Visual Studio.NET 2003 IDE(Integrated Development Environment 集成开发环境)如图 1-1 所示。从图中可见,此开发环境包含了大量可自定义显示的菜单、工具栏、可停靠窗口和面板,下面对这些窗口一一进行介绍。

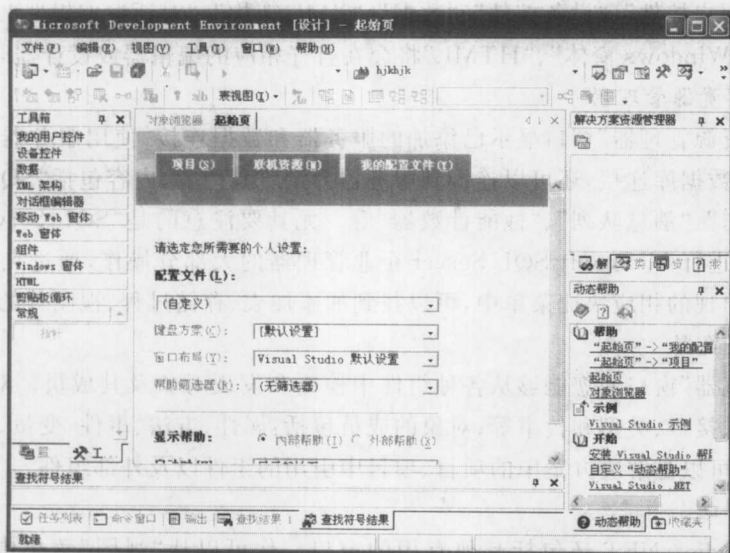


图 1-1 Visual Studio.NET 开发环境

1. 解决方案资源管理器

“解决方案资源管理器”窗口以树状结构显示当前打开解决方案的所有项目和项目所包含的所有文件,右击解决方案资源管理器的任意项时,根据项的类型不同,将会显示与该项类型相关的快捷菜单。例如:右击一个解决方案时,其快捷菜单将可以执行将一个已有项目加入到该解决方案的操作。右击一个项目时,相应的快捷菜单则可以将一个子项加入到该项目。

2. 类视图

“类视图”窗口显示了当前项目所包括的类和类型的层次信息。双击一个类或类型节点,将会打开源代码编辑器并跳转到相应位置。右击类视图中的一个类型时,显示的快捷菜单将可以执行添加方法、添加属性、添加字段、添加索引器等快捷操作。

3. 资源视图

“资源视图”窗口显示了当前项目所相关的资源文件,正常情况下,只有项目包括(.rc)文件时才会出现该视图,通过该视图可以访问项目所包括的菜单、字符串表、对话框等资源。



4. 属性

“属性”窗口用于设置解决方案中不同子项的属性,不同子项的类型对应显示的属性内容也不同,可以通过选择解决方案资源管理器、类视图、资源视图等窗口中的一个子项来显示属性内容,也可以通过选择一个设计界面元素来显示其对应的属性内容。通过该窗口,可以设置窗体和其他用户界面控件的属性。同样的,在处理 HTML 和 XML 文档时,也可以通过这个窗口来设置这些文档的对象模型属性。

5. 工具箱

“工具箱”窗口显示 Visual Studio 项目中使用的各个不同的项。根据当前正在使用的设计器或编辑器的不同,“工具箱”中可用的选项卡和项也会有相应的变化。默认状态下,“工具箱”窗口将始终显示两个选项卡:“常规”选项卡和“剪贴板循环”选项卡。其他的选项卡包括“我的用户控件”、“设备控件”、“数据”、“XML 架构”、“对话框编辑器”、“移动 Web 窗体”、“组件”、“Windows 窗体”、“HTML”将会在打开相应的编辑器或设计器时显示出来。

6. 服务器资源管理器

“服务器资源管理器”窗口显示已添加的服务器和数据连接,使用服务器资源管理器可以创建和操作数据库连接,还可以连接到服务器并浏览其内容,内容包括“SQL Server”、“服务”、“事件日志”、“消息队列”、“性能计数器”等。尤其要注意的是“SQL Server”项视图,在该视图中可以进行 Microsoft SQL Server 企业管理器的大部分操作,通过点击该视图中的各种元素,在出现的相应快捷菜单中,可以找到如添加表、存储过程、视图等的操作。

7. 对象浏览器

“对象浏览器”窗口使您能够从各种组件中检查和发现对象及其成员。对象包括:命名空间、类、结构、接口、类型和枚举等;对象的成员包括:属性、方法、事件、变量、常数和枚举项等。这些组件可以是解决方案中的项目、项目中引用的组件以及外部组件。

8. 其他窗口

Visual Studio .NET 还包括其他有用的窗口。你可以从“视图”菜单中选择“其他窗口”,然后选择想要显示的窗口。这些窗口包括:

(1) 宏资源管理器。“宏资源管理器”窗口列出了在当前解决方案中可用的所有宏。这些宏以树视图的形式显示,通过双击一个模块或右击模块名称并选择“编辑”,可以在打开的编辑器界面进行宏编辑。在这个窗口中通过简单地录制宏,然后重放它就可以大大地简化一些乏味的重复性操作。

(2) 文档大纲。“文档大纲”窗口显示了当前打开编辑的 HTML 页面的大纲和包括的脚本的大纲,通过该窗口,可以在复杂的页面中快速定位网页元素和查看以树视图组织的层次结构信息。

(3) 任务列表。“任务列表”窗口用于显示一个将要执行的任务列表,在该窗口中标记为 TODO、HACK 和 UNDONE 的项指示不影响项目编译,但会导致运行时错误的代码问题。应在运行项目之前纠正这些错误。默认情况下,这个窗口是隐藏的,从“视图”菜单中选择“其他窗口”,然后选择“任务列表”可以显示这个窗口,在“任务列表”窗口中,您可以使用以下的功能:

- 标识和定位在编辑或编译代码时自动检测到的问题。
- 双击任一智能感知消息、注释或快捷方式以打开适当的编辑器,并将插入点移动到



指定位置。

- 标识和定位在应用企业版模板策略时检测到的问题。
- 在“单击此处添加新任务”字段中输入与解决方案有关的用户说明。
- 筛选“任务列表”项,这些项使用若干预定义视图中的任意一个进行显示。
- 根据不同列对“任务列表”项进行排序,这些列可以是以“优先级”、“类别”、“已选中”、“说明”、“文件”或“行”为标题的列。

(4) 命令窗口。“命令窗口”窗口可以说是非常有用的一个窗口,该窗口有两种模式,在非调试状态时处于“命令”模式,在“命令”模式中可以在 Visual Studio .NET 环境中绕过菜单系统,直接执行命令或别名,或者用于执行不在任何菜单中出现的命令,该模式的命令提示符为“>”号。当处于调试状态时,命令窗口自动转变成“即时”模式,此模式用于调试目的,如计算表达式、执行语句、打印变量值等。它使您得以在调试期间输入表达式,由开发语言对其进行计算或执行。在某些情况下,可更改变量的值,并且“即时”模式还支持智能感知。该模式没有提示符,如果要在“即时”模式下执行“命令”模式下的命令,请以“>”号开头,或者执行>cmd 切换回“命令”模式,如果要从“命令”模式切换到“即时”模式,请在命令提示符后输入 immed。

(5) 输出。“输出”窗口显示编译项目时的各种生成错误和编译状态等信息,在调试状态时,也用来显示调试中的输出信息。

(6) 查找结果。“查找结果”窗口用以显示在“在文件中查找”对话框中执行的文本搜索匹配结果项和在“在文件中替换”对话框中执行的替换操作的替换结果项。

(7) 查找符号结果。“查找符号结果”窗口显示在“查找符号”对话框中执行的搜索的结果摘要。

(8) “收藏夹”窗口。“收藏夹”窗口中的项与 IE 中的收藏夹中的项相同,显示了以前收藏的网址,您可以把一个页面加入到收藏夹,以便下次直接从该窗口打开。

1.3.2 集成开发环境的菜单功能

1. 文件菜单

文件菜单(见图 1-2)中主要包括与文件相关的操作,其包含的主要命令如下:

(1) 新建项目:关闭当前项目,提示保存修改过的文件,从出现的新建项目对话框中选定一个项目类型,创建所选类型的一个新项目。

(2) 新建文件:新建一个指定类型的文件,从出现的新建文件对话框中选择一个文件类型,创建所选类型的一个新文件。

(3) 新建空白解决方案:新建一个空白的解决方案,然后您可以在这个空白的解决方案里加入新添的项目或现有项目。

(4) 打开项目:打开一个已有项目,从出现的打开项目对话框中选择一个已有的项目文件来打开一个已有项目。

(5) 转换:转换项目(将以前的 VB 项目转换成 VB .NET 项

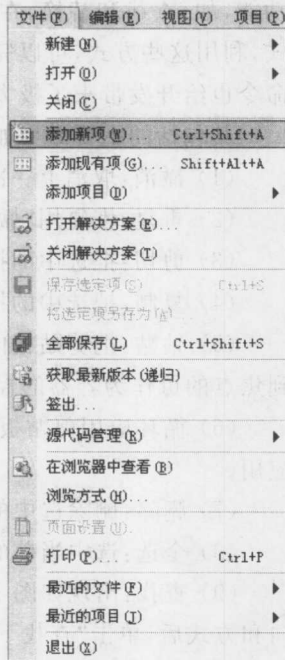


图 1-2 文件菜单



目和将 Java 语言项目转换成 C# 语言项目)。

(6) 关闭:关闭当前处于活动状态的编辑项。

(7) 添加新添项目:为当前的解决方案添加一个新的项目。

(8) 添加现有项目:为当前的解决方案添加一个现有的项目。

(9) 打开解决方案:打开一个扩展名为 .sln 的解决方案。

(10) 关闭解决方案:关闭当前打开的解决方案。

(11) 保存选定项:保存当前处于选定状态的文件项。

(12) 将选定项另存为:将当前处于选定状态的文件项保存到其他位置。

(13) 全部保存:保存当前解决方案的所有更改。

(14) 获取最新版本(递归):需要 VSS(Microsoft Visual SourceSafe)的支持,从源代码管理中获得当前打开项目的最新版本。

(15) 获取:不将文件或项目签出而直接获得源代码管理中的最新版本文件,并替换工程中的现有版本。

(16) 签出:将文件或项目从源代码管理中签出进行独占操作直到进行签入操作。

(17) 签入:将文件或项目重新签入到源代码管理中,使其可以被其他人签出编辑。

(18) 撤消签出:撤消签出操作,并回滚到文件签出时的版本。

(19) 打印:出现打印对话框,选择打印的范围、内容和质量后,单击“确定”按钮即可进行打印。

(20) 退出:退出 VS2003 集成开发环境。

2. 编辑菜单

编辑菜单(见图 1-4)中主要包括对代码编辑器的一些操作,其中包括了一些强大的新功能,如:查找和替换,在查找和替换窗口中,提供了以前不包括的正则表达式和通配符方式,利用这些方式,可以完成以前不能完成的替换任务。高级菜单里的编排文档格式等操作命令也给开发带来了极大便利。

其包括的主要命令如下:

(1) 撤消:取消上一次所作的操作。

(2) 重复:恢复用“撤消”命令所取消的操作。

(3) 剪切:把选中的控件或文本等内容移动到剪贴板。

(4) 复制:把选中的控件或文本等内容复制到剪贴板。

(5) 粘贴:把剪贴板内容复制到当前位置,文本被置于当前插入点,控件被置于当前得到焦点的可作为容器的控件中。

(6) 循环应用剪贴板中的复制项:将所有在 VS2003 中复制过到剪贴板中的项作循环应用。

(7) 删除:删除选中的控件或文本。

(8) 全选:选中当前的所有对象或文本。

(9) 查找:出现如图 1-3 所示的“查找”对话框,输入查找的文本,选择查找的范围、方向和方式后,单击“查找下一个”按钮即可进行查找。

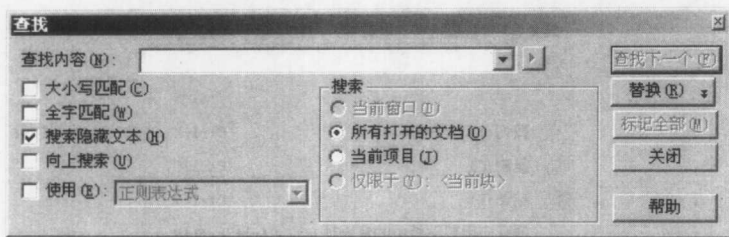


图 1-3 “查找”对话框

(10) 替换:出现与“查找”对话框相似的“替换”对话框,可将查找到的文本替换成指定文本。

(11) 查找符号:出现“查找符号”对话框,输入要查找的符号,可以在查找符号窗口显示查找结果。符号包括如类名、命名空间名、方法名、变量名等。

(12) 转到:转到指定行。

(13) 将文件作为文本插入:出现“插入文件”对话框,选择一个想要作为文本插入的文件后,该文件的内容将作为文本插入当前的插入点位置。

(14) 格式化选定内容:将选定内容的文本自动进行缩进等格式化操作。

(15) 转换为大写:将选定内容全部转换为大写。

(16) 转换为小写:将选定内容全部转换为小写。

(17) 自动换行:当一行的文本大于编辑器的可显示范围时,进行自动换行显示。

(18) 注释选定内容:注释当前的选定内容。

(19) 取消注释选定内容:取消注释当前的选定内容。

(20) 添加行缩进:添加选定行前面的缩进量。

(21) 减小行缩进:减少选定行前面的缩进量。

(22) 切换书签:使当前选中行在设置书签和取消设置书签之间切换。

(23) 下一书签:切换到下一个书签所在的位置。

(24) 上一书签:切换到上一个书签所在的位置。

(25) 清除书签:清除所有书签。

(26) 切换大纲形式展开:使当前选中区域在折叠和展开之间切换。

(27) 列出成员:出现一个下拉列表,显示当前对象可用的属性和方法。

(28) 参数信息:显示当前函数或过程对象的参数信息,并用粗体显示当前有待输入的参数。如果当前函数或过程有多个重载,则可以通过上下键进行切换显示。

3. 视图

视图菜单(见图 1-6)中主要包括了与视图显示相关的一些操作,除了在 1.3 节介绍的窗口的显示外,还包括如下主要命令:

(1) 代码:显示当前选中的窗体所对应的代码。

(2) 设计器:显示当前编辑代码所对应的设计窗体。

(3) 打开:打开选中的文件进行编辑。

(4) 打开方式:显示“打开方式”对话框,如图 1-5 所示,将当前选择的文件用其他编辑器打开后编辑。

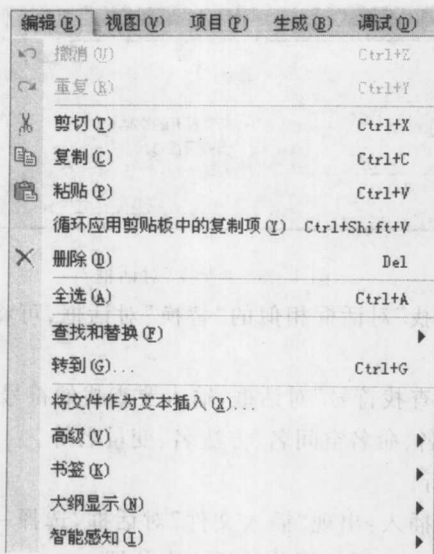


图 1-4 编辑菜单

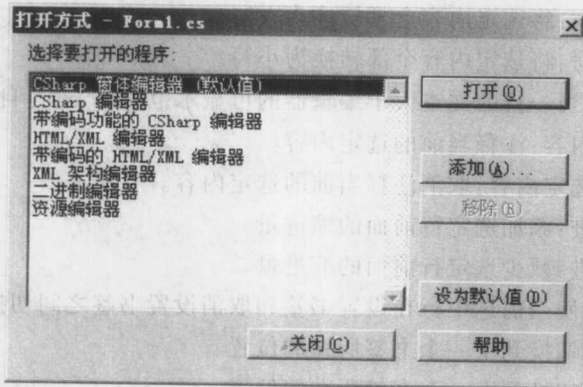


图 1-5 “打开方式”对话框

(5) 显示任务:显示任务菜单项中的各命令如显示全部,则显示注释等命令用来对任务列表窗口进行筛选操作。

(6) 工具栏:自定义工具栏的显示与隐藏,默认状态下许多工具栏都是不出现的。

(7) 全屏:可以将当前的编辑窗口在全屏和非全屏之间进行切换转换。

(8) 向前定位:类似上一页的操作,将当前的编辑焦点移向上一个跳转处。

(9) 向后定位:类似下一页的操作,将当前的编辑焦点移向下一个跳转处。

(10) 属性页:当选中一个解决方案或一个项目时,属性页命令变为可用,单击显示解决方案属性配置页或项目属性配置页。

4. 工具

工具菜单(见图 1-7)中提供了 Visual Studio .NET 包括的一些实用工具和外接程序。包括以下常用项:

(1) 连接到数据库:为“服务器资源管理器”窗口添加一个数据连接。

(2) 连接到服务器:为“服务器资源管理器”窗口添加一个服务器连接。