

Microsoft Visual Studio. **NET** 7.0丛书

Visual Basic 7.0

起步进阶

武装 等编著



- ▲ 强大的Visual Studio.NET 7.0开发工具的重要成员
- ▲ 围绕.NET框架设计
- ▲ 面向对象的新特征
- ▲ 提供丰富的设计向导
- ▲ 内容深入，实例丰富
- ▲ 针对初学者，易于掌握

国防工业出版社



205120466

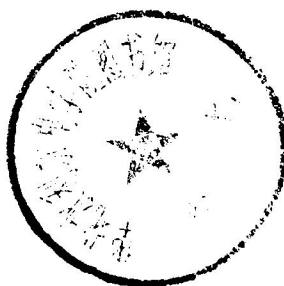
TP312

332

Microsoft Visual Studio.NET 7.0 丛书

Visual Basic 7.0 起步进阶

武装 等编著



国防工业出版社

· 北京 ·

512046

TP312
332

图书在版编目(CIP)数据

Visual Basic 7.0 起步进阶/武装等编著. —北京:
国防工业出版社, 2001.5

(Microsoft Visual Studio. NET 7.0 丛书)

ISBN 7-118-02517-8

I. V… II. 武… III. BASIC 语言 - 程序设计
IV. TP312

中国版本图书馆 CIP 数据核字(2001)第 10801 号

国防工业出版社 出版发行

(北京市海淀区紫竹院南路 23 号)

(邮政编码 100044)

三河市腾飞胶印厂印刷

新华书店经售

*

开本 787×1092 1/16 印张 16¼ 368 千字

2001 年 5 月第 1 版 2001 年 5 月北京第 1 次印刷

印数: 1—4000 册 定价: 22.00 元

(本书如有印装错误, 我社负责调换)

2001.5

丛 书 序

Visual Studio.NET 7.0 是微软推出的最新一代可视化开发工具，因版本延续及称呼方便之故，Visual Studio.NET 7.0 又被称为 Visual Studio 7.0。Visual Studio 7.0 主要用于开发企业规模的 Web 应用程序，以及高性能的桌面应用程序。所谓.NET 就是指.NET 框架(.NET Framework)，它是一种用于构建、配置和运行 Web 服务和应用程序的多语言环境，此框架提供了一种通用语言运行库(Common Language Runtime)和统一的编程类；ASP.NET(Active Server Pages.NET)正是使用这些组件来创建 Web 应用程序和 Web 服务的。实际上，正如微软在桌面操作系统中所取得的成就那样，它现在正致力于创建一种统一的开发工具。这种努力的确为开发者提供了极大的方便。

Visual Studio 7.0 中除了 C#、Visual Basic、Visual C++ 和 Visual FoxPro 等基于组件的开发工具外，还提供了一系列能够简化团队设计开发，提出解决方案的新技术。当然一直被视为开发宝典的 MSDN，也是 Visual Studio 的重要组成部分之一，有经验的程序员能够从中得到最好的帮助。在 Visual Studio 7.0 中，用户能够使用 Visual Basic、C#或 ATL 服务器，创建和包含 Web 服务。Visual Studio 7.0 还为 XML 提供了完全的支持，例如引入了 XML Designer 以简化 XML 编辑和创建等。

Visual Studio 7.0 为开发者提供了统一的集成开发环境。使用同一个 IDE 为开发者提供了很大的方便，这也符合微软统一开发语言的初衷。Visual Studio.NET 为程序员提供了灵活多样的模板支持，通过模板可以为应用程序提供结构化控制，并能降低软件设计开发的难度和成本。Visual Studio 7.0 的对窗口管理的改进，使得在屏幕上一次所能浏览的代码变得空前的多。在 Visual Studio 7.0 中对编辑工具作了进一步增强，整合了对象浏览器和代码注释支持。Visual Studio 7.0 中所有的语言都使用统一的代码编辑器，当然对于每种语言都有特定的功能。编辑器的另一个强大的功能是拥有剪贴板环(Clipboard Ring)。剪贴板环中能够存储系统剪贴板中被拷贝或剪切的最后 20 个条目。Visual Studio 7.0 中的所有语言使用同样的调试器，它同时具有旧版本的 Visual C++ 和 Visual Basic 调试器的功能，并在很多地方进行了改进。Visual Studio 7.0 中的安装模型，与以前版本中的传统安装模型之间有着显著的不同，例如支持 Microsoft Windows 安装技术，并简化了产品发布的开发。Visual Studio 7.0 的 IDE 拥有极其强大的自动化模型和宏支持，还为用户提供了多种快速检索和浏览文档的手段，例如动态帮助窗口和 MSDN 检索等。

本套丛书中主要介绍 Visual Studio 7.0 中的编程语言，明确定位于初、中级用户。无论您是否使用过 Visual Studio，这套书对您都非常适合。本套丛书包括：

- Visual C++ 7.0 起步进阶
- Visual Basic 7.0 起步进阶
- Visual C# 起步进阶
- Visual FoxPro 7.0 起步进阶

书名中的“起步”是指，对于每个编程语言的讲解都从必备的基础操作和基本语法讲起，使得新用户能够轻松入门；老用户亦可从中快速了解新版本的新特色和新功能。至于“进阶”则意味着本书将在读者快速入门后，以明晰的操作步骤和典型的应用实例，教会读者更丰富全面的程序设计技术和应用技巧，使得读者真正对软件融会于心，熟练在手。

前 言

Visual Basic 的诞生使编程技术向前迈进了一大步，开可视化编程的一代先河。在它的带动下，许多优秀的可视化开发工具相继问世。这些开发工具各有千秋，但它们都或多或少地从 Visual Basic 中汲取了营养。随着组件对象技术的不断进步，以及 Internet 应用的不断普及，Visual Basic 也在不断进步。Visual Basic.NET 7.0 是 Microsoft 公司最新推出的 Visual Studio.NET 7.0 中的组成部分，是围绕 .NET 框架设计更新的。因版本延续与称呼方便之故，它们分别又被称为 Visual Basic 7.0 和 Visual Studio 7.0。

Visual Basic 7.0 有许多改进的语言特征，例如继承、接口和重载。这些特征使得 Visual Basic 7.0 成为一种强大的面向对象的编程语言。Visual Basic 7.0 的其他特征包括结构化异常处理、定制标志和通用语言规范兼容性。Visual Basic 7.0 中添加了一些新特性以利用 CLS 的优势。使得任何与 CLS 兼容的语言都可以使用由 Visual Basic 7.0 创建的类、组件和对象。同样，Visual Basic 7.0 也能访问由其他 CLS 兼容语言创建的类、组件和对象。

Visual Basic 7.0 支持许多新型面向对象语言的特性，例如继承、重载、重载关键字、接口、共享成员和构造函数。此外还引入了一些新的数据类型、结构化异常处理和 delegate。

本书易于理解，对 Visual Basic 7.0 的几乎所有方面都有介绍，并且本书假定读者以前从未用 Visual Basic 编过程序。对于那些以前使用过 Visual Basic 的早期版本或其他结构化语言编过程序的读者来说，本书会更容易掌握。如果从头开始学习，那么很快就能熟练掌握 Visual Basic 7.0，并能充分利用其强有力的事件驱动特性去编写精彩、完美的 Windows 应用程序。

本书面向初中级读者，侧重对基本概念、基本技能和简单的应用程序设计技术的介绍。由于时间所限，书中错误和疏漏之处在所难免，敬请指正。

本书除封面署名作者外，王江辉、廖晓筠、沈冰、汤春明、林茵茵、黄剑波、胡建明、龚雪梅、黄君玲、朱石、周一兵、李义、于佳音、王芳、沈鹏、刘树声、季洪飞、薛文涛、司马小凡、许颖、赵立峰、李国梁、吴强等都为本书的出版付出了不同程度的劳动，在此一并表示感谢。

编 者

目 录

第 1 章	Visual Basic 7.0 使用入门	1
1.1	Visual Basic 7.0 的新特性	1
1.2	Visual Basic 7.0 集成开发环境	4
1.2.1	Visual Studio 7.0 集成开发环境的新改进	4
1.2.2	Visual Studio 起始页	8
1.2.3	方案管理器窗口	9
1.2.4	Class View 窗口	11
1.2.5	属性窗口	11
1.2.6	任务清单窗口	13
1.2.7	工具箱窗口	13
1.2.8	编辑器窗口	16
1.2.9	服务器管理器窗口	16
1.2.10	对象浏览器窗口	18
1.2.11	宏管理器窗口	22
1.3	Visual Basic 7.0 的菜单体系	25
1.3.1	File 菜单	25
1.3.2	Edit 菜单	28
1.3.3	View 菜单	30
1.3.4	Project 菜单	31
1.3.5	Build 菜单	34
1.3.6	Debug 菜单	35
1.3.7	Tools 菜单	36
1.3.8	Window 菜单	38
1.3.9	Help 菜单	39
	本章小结	40
第 2 章	Visual Basic 7.0 语言基础	41
2.1	Visual Basic 7.0 与 .NET 框架	41
2.1.1	引用与 Import 语句	41
2.1.2	部件	42
2.1.3	名称空间	43
2.1.4	标志	44
2.2	Visual Basic 7.0 中的语言更新	44

2.2.1	数组上下限的修改	44
2.2.2	数组尺寸声明的修改	45
2.2.3	ReDim 语句的修改	45
2.2.4	字符串长度声明的修改	46
2.2.5	整数数据类型的修改	46
2.2.6	通用数据类型的修改	46
2.2.7	数据类型的修改	47
2.2.8	布尔操作符的修改	47
2.2.9	默认属性的修改	48
2.2.10	属性的修改	49
2.2.11	结构声明的修改	50
2.2.12	声明语法的修改	51
2.2.13	变量作用域的修改	51
2.2.14	对象创建的修改	52
2.2.15	过程声明的修改	53
2.2.16	过程调用顺序的修改	53
2.2.17	参数传递机制的修改	54
2.2.18	流程控制语句的修改	55
2.2.19	由方法代替的编程元素	55
2.2.20	被修改了的编程元素	56
2.2.21	类和接口的修改	57
2.3	数据类型	59
2.3.1	数字类型	59
2.3.2	二进制数据类型	60
2.3.3	字符数据类型	61
2.3.4	字符串数据类型	61
2.3.5	日期数据类型	62
2.3.6	Object 数据类型	62
2.3.7	枚举类型	64
2.3.8	引用类型与数值类型	66
2.4	类型转换	67
2.4.1	扩展转换与紧缩转换	67
2.4.2	隐式转换与显式转换	68
2.4.3	数组类型转换	70
2.5	变量和常数	70
2.5.1	变量声明	70
2.5.2	变量作用域	74
2.5.3	静态变量	76
2.5.4	常数	77

2.5.5	变量和常数的命名约定	78
2.6	数组	80
2.6.1	声明数组	81
2.6.2	数组赋值	82
2.6.3	返回数组的函数	83
2.6.4	数组的数组	84
2.6.5	数组与集合	85
2.7	结构	85
2.7.1	结构声明	86
2.7.2	结构变量	86
2.7.3	结构和数组	87
2.7.4	结构和过程	88
2.7.5	结构和对象	88
2.7.6	结构和类的异同	89
2.8	过程	89
2.8.1	子过程	90
2.8.2	函数过程	92
2.8.3	子过程和函数过程的使用	92
2.8.4	向过程传递参数	94
2.8.5	属性过程	97
2.9	流程控制	99
2.9.1	条件判定	99
2.9.2	循环	102
2.9.3	使用控制结构	105
	本章小结	107
第 3 章	Visual Basic 7.0 编程基础	108
3.1	Windows 应用程序的工作机制	108
3.1.1	事件驱动	108
3.1.2	事件处理函数	109
3.2	Visual Basic 应用程序的结构	110
3.3	Visual Basic 的编程对象	112
3.3.1	对象概述	112
3.3.2	操作对象	114
3.3.3	对象之间的关系	116
3.3.4	创建对象	119
3.4	管理应用程序设置	125
3.4.1	创建或保存应用程序设置值	126
3.4.2	检索应用程序设置值的方法	126
3.4.3	删除应用程序设置值	127

3.5	条件编译与编码基础	127
3.5.1	条件编译	128
3.5.2	编码基础	129
3.6	Hello, World!	131
3.6.1	创建界面	131
3.6.2	设置对象属性	131
3.6.3	编写代码	132
3.6.4	运行程序	133
	本章小结	133
第 4 章	使用标准控件	135
4.1	工具箱与控件	135
4.1.1	工具箱	135
4.1.2	生成和调整控件	136
4.2	标签控件	139
4.2.1	常用标签属性和方法	139
4.2.2	常用标签事件	140
4.2.3	使用标签	140
4.3	图片框控件	141
4.3.1	常用图片框属性	141
4.3.2	使用图片框	141
4.4	文本框控件	141
4.4.1	常用文本框属性和方法	141
4.4.2	常用文本框事件	143
4.4.3	使用文本框	143
4.5	列表框控件	145
4.5.1	常用列表框属性和方法	145
4.5.2	列表框常用事件	146
4.5.3	使用列表框	146
4.6	组合框控件	147
4.6.1	常用组合框属性和方法	147
4.6.2	常用组合框事件	147
4.6.3	使用组合框	147
4.7	按钮控件	148
4.8	复选框、单选按钮和编组框	149
4.8.1	常用复选框属性	149
4.8.2	常用单选按钮属性	149
4.8.3	常用编组框属性	150
4.9	滚动条控件	150
4.9.1	常用滚动条属性	151

4.9.2 常用的滚动条事件	151
4.10 计时器控件	151
4.11 控件焦点与命名约定	152
4.11.1 焦点	152
4.11.2 设置 tab 键顺序	153
4.11.3 控件命名约定	153
4.12 控件的综合应用	155
本章小结	158
第 5 章 界面设计技术	160
5.1 深入了解窗体	160
5.1.1 窗体概述	160
5.1.2 相对于 Visual Basic 6.0 的更新	161
5.1.3 默认窗体代码解析	162
5.1.4 模式和非模式	163
5.1.5 最顶层窗体	164
5.1.6 创建透明窗体	164
5.1.7 重设窗体尺寸	164
5.1.8 设置窗体的位置	165
5.1.9 窗体边框风格	166
5.1.10 设置启动窗体	168
5.1.11 闪屏效果	169
5.2 选择界面样式	169
5.3 设计 MDI 界面	171
5.3.1 创建 MDI 父窗体	172
5.3.2 创建 MDI 子窗体	172
5.3.3 确定活动子窗体	173
5.3.4 子窗体排列	174
5.3.5 维护子窗体的状态信息	174
5.4 设计对话框	175
5.4.1 创建对话框	175
5.4.2 显示对话框	176
5.4.3 对话框中的用户输入	176
5.4.4 使用属性获取对话框信息	177
5.4.5 使用对象获得对话框信息	177
5.4.6 从父窗体中获取信息	178
5.4.7 关闭对话框	179
5.4.8 显示消息框	179
5.5 设计菜单	180
5.5.1 添加菜单和菜单项	180

5.5.2	移动菜单项	183
5.5.3	拷贝菜单项	183
5.5.4	禁止菜单项	184
5.5.5	隐藏菜单项	184
5.5.6	删除菜单项	185
5.5.7	其他菜单特性	186
5.5.8	合并菜单	186
5.5.9	切换窗体菜单	187
5.6	使用快捷菜单	190
5.6.1	添加快捷菜单	191
5.6.2	动态快捷菜单	191
5.7	工具栏和状态栏	194
5.7.1	使用工具栏	194
5.7.2	使用状态栏	195
	本章小结	197
第 6 章	面向对象的程序设计	198
6.1	基础概念	198
6.1.1	对象和类	198
6.1.2	封装、继承和多态性	199
6.1.3	共享成员	199
6.2	管理对象组	201
6.2.1	创建对象数组	201
6.2.2	集合	203
6.2.3	创建对象集合	205
6.3	定制类	205
6.3.1	类是数据和过程的结合	205
6.3.2	类对象的创建和销毁	207
6.3.3	为类添加属性	208
6.3.4	为类添加方法	209
6.3.5	友元属性和方法	211
6.3.6	使用字符串名调用属性或方法	212
6.3.7	为类添加事件	213
6.3.8	定制事件处理实例	215
6.3.9	类属性和方法的多态性	219
6.3.10	类成员的命名规则	221
6.4	接口	221
6.4.1	.NET 框架中的接口	221
6.4.2	创建和实现接口	223
6.5	继承	224

6.5.1	继承基础	225
6.5.2	创建派生类	226
6.5.3	何时使用继承	228
6.5.4	继承、接口和多态性	228
	本章小结	231
第 7 章	异常处理	232
7.1	异常处理概述	232
7.1.1	Visual Basic 中的错误种类	232
7.1.2	Err 对象	233
7.2	非结构化异常处理	234
7.2.1	On Error GoTo Line	234
7.2.2	On Error Resume Next	234
7.2.3	On Error GoTo 0	235
7.2.4	On Error GoTo -1	235
7.3	结构化异常处理	236
	本章小结	239
第 8 章	自由线程	240
8.1	理解自由线程	240
8.1.1	创建和使用新线程	240
8.1.2	为新线程提供参数	240
8.1.3	获取线程信息	241
8.1.4	同步化	241
8.2	自由线程示例	242
	本章小结	247

第 1 章 Visual Basic 7.0 使用入门

Visual Basic 是当今最流行的软件开发工具之一，是程序员的首选编程利器。Visual Basic 7.0 并不只是一个程序代码的编辑器和编译器。要想掌握使用 Visual Basic 7.0 编程的方法，首先需要掌握软件的基本操作，这样可以提高使用软件的效率。本章将向读者介绍其用户界面和编程工具，使读者能够快速掌握它的基本使用。

本章要点：

- ❖ Visual Basic 7.0 的新特性
- ❖ Visual Basic 7.0 集成开发环境的使用
- ❖ Visual Basic 7.0 菜单的使用方法

1.1 Visual Basic 7.0 的新特性

Visual Basic.NET 7.0 是 Visual Studio.NET 7.0 家族的一个成员。因版本延续及称呼方便之故，Visual Basic.NET 7.0 又被称为 Visual Basic 7.0。

由 Visual Studio.NET 的名称就可以断定：它较之其以前版本，至少在网络开发方面有了显著的改进。这一点从微软对 Visual Studio 7.0 的定位得到了验证——它主要用于开发企业规模的 Web 应用程序，以及高性能的桌面应用程序。所谓 .NET 就是指 .NET 框架（.NET Framework），此框架由通用语言运行库（Common Language Runtime）、统一的编程类和 ASP.NET（Active Server Pages.NET）组成。通过这些组件能够方便地创建 Web 应用程序和 Web 服务。

Visual Studio 7.0 中除了 C#、Visual C++、Visual Basic 和 Visual FoxPro 等基于组件的开发工具外，还提供了一系列能够简化团队设计开发，提出解决方案的新技术。当然一直被视为开发宝典的 MSDN，也是 Visual Studio 的重要组成部分之一，有经验的程序员能够从中得到最好的帮助。

在 Visual Studio 7.0 中，用户能够使用 Visual Basic、C# 或 ATL 服务器，创建和包含 Web 服务。Visual Studio 7.0 为 XML 提供了完全的支持，例如引入了 XML Designer 以简化 XML 编辑和创建等。

由于本书的主题限制，这里主要向读者介绍 Visual Basic 7.0 的一些重要更新。

Visual Basic 7.0 有许多改进的语言特征，例如继承、接口和重载。这些特征使得 Visual Basic 7.0 成为一种强大的面向对象的编程语言。而且，Visual Basic 开发者现在还能够使用显式多线程处理创建多线程、可缩放的应用程序。Visual Basic 7.0 的其他特征包括

结构化异常处理、定制标志和通用语言规范 (CLS, Common Language Specification) 兼容性。

Visual Basic 7.0 中添加了一些新特性以利用 CLS 的优势。CLS 是一套规则, 它标准化了数据类型和内部对象如何被外部使用的方式。任何与 CLS 兼容的语言都可以使用由 Visual Basic 7.0 创建的类、组件和对象。同样, Visual Basic 7.0 也能访问由其他 CLS 兼容语言创建的类、组件和对象。这种互用无需考虑语言规范之间的差别, 因此是无缝的。Visual Basic 7.0 使用的 CLS 特性包括部件、名称控件和标志。

Visual Basic 7.0 支持许多新型面向对象语言的特性, 例如继承、重载、重载关键字、接口、共享成员和构造函数。此外还引入了一些新的数据类型、结构化异常处理和 delegate。

- 继承

Visual Basic 7.0 允许为类定义基类。派生类继承并能扩展基类的方法和属性, 而且还能重载继承所得的方法以提供更新的实现。所有由 Visual Basic 7.0 创建的类都默认是可继承的。因为设计的窗体实际就是类, 因此可以使用继承定义基于现存窗体的新窗体。

继承的目标之一就是允许代码重用。代码重用有两种形式: 组合和继承。Visual Basic 一直都允许创建和重用 COM 组件, 并因此支持组合代码重用。而继承则是在 Visual Basic 7.0 中新添的特性。

继承和接口允许使用多态性 (Polymorphism), 这允许同名方法或属性可以定义于多个类中, 而客户代码可以在运行时区分使用。多态性对于面向对象编程是非常重要的, 因为它允许调用同名方法, 而无论当时使用的是何种类型的对象。

- 结构化异常处理

Visual Basic 支持使用 Try-Catch-Finally 的增强版本进行结构化异常处理。结构化异常处理中组合了先进的异常控制结构 (与 Select Case 或 While 语句类似)、保护型代码块和过滤器, 这使得程序更加稳固而不会轻易崩溃。

- 多态性

多态性允许定义同名但不同数据类型的属性、方法或过程。多态型过程允许以单个通用过程为不同数据的处理提供多种实现。

只要进行了一般形式的函数调用, 多态性就会保证把函数的实现细节留给接受函数调用的对象。一般情况下, 函数调用时用到的参数决定了函数的行为。利用传送的不同参数, 可以使同样的对象具有不同的性质、功能。如函数调用 Draw 时, 向其传递一个位图对象作为参数, 则将该位图绘制在屏幕上; 而当传递给它的参数为一个文本对象时, 其执行的功能是在屏幕上输出文字。多态性有时也指方法的重载。方法的重载是指同一个方法名在上下文中有不同的含义, 是让类以统一的方式处理不同数据类型的一种手段, 它是静态而不灵活的, 但在某种情况下非常必要。

- 重载方法

Overrides 关键字允许派生对象重载继承自父对象的特征。重载方法与基类方法的参数相同, 但是具体的实现不同。如果在方法名前指定了 MyBase 关键字, 则新的方法实现中可以调用其基方法的原始实现。

- 构造函数和析构函数

构造函数是控制类的新实例初始化的方法；而析构函数则是用于释放由类实例占用的系统资源的方法。Visual Basic 7.0 中通过 Sub New 和 Sub Destruct 支持构造函数和析构函数。

- 新的数据类型

Visual Basic 7.0 中引入了下列新数据类型：

Char——无符号 16 位整数，用于存放 Unicode 字符，等价于 .NET 框架的 System.Char 类型。

Short——有符号 16 位整数。在早期的 Visual Basic 版本中，此数据类型被称为 Integer。

Decimal——96 位十进制值。在早期的 Visual Basic 版本中，此数据类型只能在 Variant 中使用。

- 接口

接口描述了类的方法和属性，但不提供实现。在 Visual Basic 7.0 中，Interface 关键字允许声明接口，而 Implements 关键字允许为接口成员编写实现代码。

- delegate

每个 delegate 实例都将进一步调用特定对象的某个方法，而对象和方法的选择则是在构造 delegate 实例时决定的。因此，delegate 有时被描述为类型安全的、面向对象的函数指针。delegate 可以用于事件处理和自由线程中。

- 共享成员

共享成员就是能由所有类实例使用的属性、过程和字段。对于使用继承特性的 Visual Basic 7.0 应用程序来说，所有类实例共享同一个数据成员或函数实例是非常有用的。

- 引用

引用允许使用定义于其他部件中的对象。

- 名称空间

使用名称空间能够避免在类型库中将类、接口和方法按层次结构组织时出现的名称冲突。实际上，在 Visual Basic 7.0 中创建的工程也是一个名称空间，其名称即为工程名。

- 部件

部件替代了类型库，它能够描述特定组件或应用程序所需的所有文件。部件可以包含一个或多个名称空间。

- 标志

标志为程序元素提供了附加信息。例如，可以指定类方法的访问修饰符。

- 自由线程处理

在 Visual Basic 7.0 中，可以编写独立执行多个任务的应用程序。所谓自由线程处理，就是指在另一个独立线程中运行，可能阻塞其他任务执行的任务。由于可以使用与用户界面独立的线程运行复杂任务，因此自由线程处理使得应用程序对用户输入的响应更加灵敏。

这些新特性使 Visual Basic 7.0 更加适应现在计算机网络化、运行速度快以及加强数据传输的趋势，成为软件开发的首选工具。

1.2 Visual Basic 7.0 集成开发环境

Visual C++ 7.0、Visual Basic 7.0 和 Visual Foxpro 使用同样的用户界面，即 Visual Studio 7.0 的集成开发环境（IDE，Integrated Development Environment）。使用同一个 IDE 为开发者提供了很大的方便，这也符合微软统一开发语言的初衷。Visual Studio 具有包括源码创建、资源编辑、编译、链接和调试等在内的许多功能，其初始启动界面如图 1-1 所示。它的界面是智能化的，而且非常宽容，鼓励用户去实践和尝试。

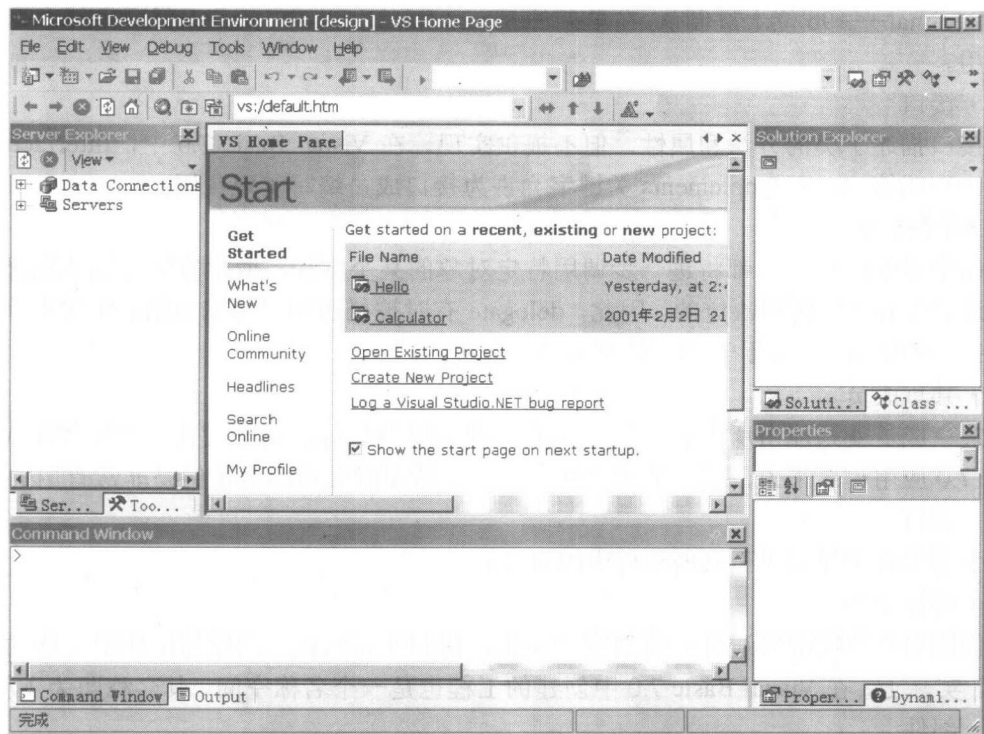


图 1-1 Visual Studio 7.0 的用户界面

1.2.1 Visual Studio 7.0 集成开发环境的新改进

本节将向读者简单介绍 Visual Studio 7.0 集成开发环境中的新加或改进之处。

1.2.1.1 通用集成开发环境工具

Visual Studio 7.0 中新增的工具为开发者提供了很大方便。

• Web 浏览器

Visual Studio 7.0 的 IDE 能够直接显示网页。要使 Web 浏览器在 IDE 中出现，只要选择 View/Web Browser 菜单命令即可。当首次打开 Web 浏览器窗口时，其中将显示默认的 Visual Studio 起始页。如果 Web 浏览器被打开，则同时也会出现 Web 工具栏，其功能与 IE 中的相同。

- Visual Studio 起始页

在默认情况下，每次 Visual Studio 启动时，Visual Studio 起始页都显示在其用户界面的 Web 浏览器窗口中，如图 1-1 所示。它提供了设置诸如 IDE 行为、键盘类型、窗口布局等用户参数，以及进行打开、新建工程等操作的快速途径。通过起始页还能查看最新的文章、事件和 MSDN 在线的主题。

- 命令窗口

命令窗口如图 1-2 所示，它是命令行和 Visual Basic 即时窗口（Immediate Window）的结合，因而也具有两个模式：命令模式和即时模式。在命令模式下，用户能够在角括号>后输入 IDE 命令名。为了方便地执行使用频率较高的命令，可以创建别名（Alias）或短名。自动完成（Auto-completion）功能对命令名、别名和文件名也同样有效。用户可以通过命令窗口执行代码行，对变量进行设置或赋值，或计算表达式的值。使用命令还有另一种途径，这就是直接在 Visual Studio 工具栏中的 Find/Command 框中输入命令。

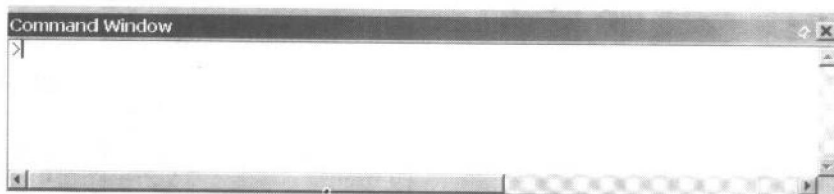


图 1-2 命令窗口

1.2.1.2 窗口管理

Visual Studio 7.0 的对窗口管理的改进，使得在屏幕上一次所能浏览的代码变得空前的多。图 1-3 所示为 Visual Basic 7.0 中的窗口布局，其中包括了所有主要窗口。

- 标签化文档

标签化文档功能将自动在 IDE 中为文档窗口设置标签。例如，默认情况下 Class View、Resource View、宏管理器和方案管理器使用同一（左上）窗口，而通过切换底部标签以在此窗口中查看不同视图中的内容。当用户在编辑器或设计器中编辑多个文档时，它们将全部显示于多文档界面（MDI）区域中。虽然一次只能显示一个文档中的内容，但是通过顶部/底部标签可以方便地浏览其他文档中的内容，参见图 1-3。这较之以前的版本中，必须通过 Window 菜单中的窗口列表选择，其提供的方便显而易见，尤其是在打开的文档数较多时。

- 自动隐藏

在窗口标题栏处单击右键，在弹出的快捷菜单中选择 Auto Hide 命令，即可实现自动隐藏。自动隐藏允许用户能够最小化工具窗口，例如方案管理器和工具框，并将其排列在 IDE 边缘，从而节省宝贵的屏幕空间，提高编辑器的可视面积。

- 前进/后退导航（Navigate Backward/Navigate Forward）

前进/后退导航允许用户在打开的窗口中，或同一文件中的被选文本和光标位置中进行切换，这与 Web 浏览器中的“前进”和“后退”功能极为类似。例如，如果用户在第 12 行中编辑代码后移动到第 102 行，这时就可以通过 Navigate Backward 按钮返回光标上次在 12 行中的位置。Navigate Backward 和 Navigate Forward 按钮都带有下拉列表，其