

Visual Basic .NET

学习范本

通过本书的学习将掌握下列基本技能

- 面向对象程序设计的技巧
- 事件的处理与运用
- 控制项的灵活搭配运用
- 程序图形化界面的设计
- 视窗环境下的电脑绘图技巧
- 子程序与模块的灵活运用
- 文档的存取与管理
- 数据库的设计与管理
- 音效的处理技巧
- 多媒体程序设计技巧
- 自行设计一个网页浏览
- Web Service设计技巧

Microsoft Certified
Professional

曹祖圣 等编著

Database Administrator
Systems Engineer
Solution Developer
Trainer



科学出版社

www.sciencep.com

Visual Basic .NET 学习范本

曹祖圣 等 编著

科学出版社

北 京

图字: 01-2003-6991 号

内 容 简 介

Visual Studio .NET 是微软公司于 2002 年推出的新平台, 它使编程更加快捷、方便。其中的 Visual Basic .NET 不仅保留了以前版本易学易用的特性, 而且提供了面向对象语言的全部特性。本书共 16 章, 分别介绍了集成环境的基本使用方法、程序的基本流程、函数的调用、语句结构、多媒体及数据库应用等方面的知识。全书结合实例, 一步一步深入介绍 Visual Basic .NET, 使读者能够通过本书的学习掌握 Visual Basic .NET 的编程。

本书可作为大学本科教材, 也可供初学者或有经验的程序员阅读参考。

本书繁体字版名为《Visual Basic .NET 学习范本》。由文魁资讯股份有限公司出版, 版权属曹祖圣、吴明哲、黄世阳所有。本书简体字中文版由文魁资讯股份有限公司授权科学出版社独家出版。未经本书原版出版者和本书出版者书面许可, 任何单位和个人不得以任何形式或任何手段复制或传播本书的部分或全部内容。

图书在版编目(CIP)数据

Visual Basic .NET 学习范本 / 曹祖圣等编. —北京: 科学出版社, 2003

ISBN 7-03-012318-2

I. V... II. 曹... III. BASIC 语言—程序设计—教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2003) 第 091180 号

策划编辑: 李佩乾 / 责任编辑: 朱凤成
责任印制: 吕春珉 / 封面设计: 一克米工作室

科 学 出 版 社 出 版

北京东黄城根北街16号

邮政编码: 100717

<http://www.sciencep.com>

双 青 印 刷 厂 印 刷

科学出版社发行 各地新华书店经销

2003 年 10 月第 一 版 * 开本: 787×1092 1/16

2003 年 10 月第一次印刷 印张: 26 1/4

印数: 1—4 000 字数: 610 000

定价: 43.00 元 (含光盘)

(如有印装质量问题, 我社负责调换〈路通〉)

前 言

在微软的软件开发工具中，Visual Studio 一直是 Windows 平台上集成能力最强、功能最齐的程序开发工具。新版的 Visual Studio .NET 针对多项软件开发技术加以强化。其中，Visual Basic .NET 不但继承了以往易学易用的特性，让初学者得以轻松进入程序设计的殿堂，而且支持 Visual Basic 以前版本所欠缺的面向对象机制。这样，使用 Visual Basic .NET 来开发大型的软件系统时，就可以降低软件设计与维护的复杂度。

在软件开发架构上，.NET 框架将成为未来软件开发的通用平台。.NET 框架很容易移植到各个不同的平台上(Windows、UNIX、PDA、手机等)。Visual Basic .NET 程序员可以直接运用.NET 框架中完整且丰富的类库，设计出跨平台的软件系统。另外通过公共语言运行库(Common Language Runtime)的接口，在.NET 框架上所开发的软件，不论使用哪一种程序语言，都可以直接互相引用。在多人共同开发软件系统时，就可以使用程序员自己所专长的程序语言来参与软件开发，直接解决了跨语言集成的问题。

微软提出的.NET 战略所要建立的信息应用环境，就是让用户可以在任何时刻(any time)、任何地点(any place)、使用任何设备(any device)来存取所需要的信息。而.NET 框架与 Visual Studio .NET 正是用来构建.NET 战备的基础平台与开发工具。通过开发 XML Web 服务与相关的机制，不但能够集成各种平台上的开发成果，降低企业 e 化成本与缩短导入时间，而且提高了分布式系统的安全性及多功能性。

本书的理念是，使用浅显易懂的语法与丰富的范例，将基本的编程技术传授给读者。

由于篇幅的限制，本书无法将 Visual Basic .NET 所有的功能介绍完整，但基本上可以满足教学的需要。笔者会陆续整理相关资料，以求再版时予以修订。敬请广大师生对书中存在的问题不吝指教。

为方便教学，我们通过网站提供了本书所有范例代码，下载地址是 www.abook.cn。

曹祖圣 吴明哲 黄世阳 何嘉益 张志成

目 录

第 1 章 Visual Basic .NET 集成环境介绍	1	3.2.2 变量的种类	40
1.1 程序语言的发展	1	3.2.3 常数 (Constant)	45
1.2 BASIC 的发展	2	3.3 文字数据表示法	46
1.3 VB .NET 功能介绍	2	3.4 数字系统	47
1.4 对象和事件的基本概念	4	3.5 数字系统的转换	48
1.5 软硬件需求	6	3.6 习题	51
1.6 激活集成开发环境	6	第 4 章 窗体与基本语句	52
1.7 VB .NET 集成开发环境介绍	8	4.1 窗体	52
1.7.1 主窗口	9	4.1.1 窗体的建立	52
1.7.2 工具栏	10	4.1.2 设置窗体的属性	52
1.7.3 工具箱	11	4.2 标签与超链接标签	54
1.7.4 解决方案资源管理器	11	4.2.1 标签的建立	55
1.7.5 【属性】窗口	12	4.2.2 标签的属性	55
1.7.6 代码页	13	4.2.3 LinkLabel 控件的创建	56
1.8 动手写第一个 VB .NET 程序	14	4.2.4 LinkLabel 控件的属性	56
1.9 执行程序	16	4.3 按钮	56
1.10 打开旧文件	16	4.3.1 按钮的创建	56
1.11 离开 VB .NET 集成环境	17	4.3.2 设置 Button 控件常用的属性	57
1.12 习题	17	4.4 VB .NET 程序设计	57
第 2 章 程序设计基本流程	19	4.5 VB .NET 基本语句	67
2.1 如何开发应用程序	19	4.6 表达式与运算符	71
2.2 如何编辑 VB .NET 程序	21	4.7 习题	73
2.3 程序如何执行	29	第 5 章 输入工具与函数	75
2.4 程序如何存盘	30	5.1 文本框	75
2.5 如何读取程序	31	5.1.1 文本框的建立	75
2.6 如何打印程序代码	32	5.1.2 文本框的属性	75
2.7 如何在 Windows 下执行程序	35	5.1.3 数据多行显示	76
2.8 习题	36	5.2 数值与字符串间的转换函数	76
第 3 章 基本数据类型	38	5.3 InputBox 函数	82
3.1 存储器存储单位	38	5.4 MsgBox 函数与 MsgBox 语句	85
3.2 变量	39	5.5 月历与日期挑选	87
3.2.1 变量的命名规则	39	5.5.1 月历的建立	88
		5.5.2 月历的属性	88
		5.5.3 日期挑选的建立	88

5.5.4 日期挑选工具的属性.....	89	6.12 习题.....	135
5.5.5 范例.....	89	第7章 重复结构	138
5.6 格式化输出.....	92	7.1 计数循环.....	138
5.6.1 Space 函数.....	92	7.2 PictureBox 控件.....	140
5.6.2 vbTab 常数.....	92	7.2.1 PictureBox 控件的建立.....	141
5.6.3 Format 函数.....	92	7.2.2 PictureBox 控件的应用.....	142
5.7 实例.....	95	7.3 条件循环.....	145
5.8 习题.....	97	7.3.1 Do While...Loop 前测试循环.....	145
第6章 选择结构	100	7.3.2 Do Until...Loop 前测试循环.....	146
6.1 关系表达式与逻辑表达式.....	100	7.3.3 Do...Loop While 后测试循环.....	146
6.2 算法.....	101	7.3.4 Do...Loop Until 后测试循环.....	147
6.3 结构化程序设计.....	103	7.3.5 Do...Loop 无穷循环.....	147
6.3.1 循序结构.....	103	7.4 多重循环 Nested Loop.....	148
6.3.2 选择结构.....	103	7.5 局域变量与全局变量.....	149
6.3.3 重复结构.....	105	7.6 定时器 Timer.....	152
6.4 选择语句.....	106	7.6.1 定时器的建立.....	152
6.4.1 单一选择的 If...Then 语句.....	106	7.6.2 定时器的应用.....	152
6.4.2 双向选择的 If...Then...Else 语句.....	108	7.7 图像列表 ImageList.....	157
6.4.3 If 函数.....	109	7.7.1 图像列表控件的建立.....	158
6.5 单选按钮 RadioButton.....	110	7.7.2 图像列表控件的应用.....	159
6.5.1 单选按钮的建立.....	110	7.8 滚动条 ScrollBar.....	160
6.5.2 单选按钮的应用.....	111	7.8.1 滚动条的建立.....	161
6.6 多重选择语句.....	112	7.8.2 滚动条控件的应用.....	162
6.7 复选框 CheckBox.....	114	7.9 实例.....	164
6.7.1 复选框的建立.....	114	7.10 习题.....	167
6.7.2 复选框的应用.....	115	第8章 键盘与鼠标菜单	171
6.8 多向选择.....	118	8.1 键盘事件.....	171
6.8.1 Select Case 语句.....	118	8.1.1 KeyPress 事件.....	171
6.8.2 Choose 函数.....	121	8.1.2 KeyDown 和 KeyUp 事件.....	173
6.8.3 Switch 函数.....	121	8.2 鼠标事件.....	177
6.9 组合框 GroupBox.....	122	8.2.1 Click 和 DblClick 事件.....	177
6.9.1 组合框的建立.....	123	8.2.2MouseDown、MouseUp 与 MouseClick 事件.....	178
6.9.2 组合框的应用.....	124	8.3 菜单 Menu.....	184
6.10 面板 Panel 与数字按钮 NumericUpDown.....	125	8.3.1 菜单的介绍.....	184
6.10.1 面板与数字按钮的建立.....	125	8.3.2 菜单栏的制作.....	185
6.10.2 面板与数字按钮的应用.....	126	8.3.3 菜单栏的应用.....	189
6.11 实例.....	128	8.4 工具栏.....	192

8.4.1 连接 ImageList (图像列表).....	192	10.7 ArrayList 类别.....	256
8.4.2 使用 ToolBar(工具栏)控件.....	193	10.8 习题.....	261
8.5 快捷菜单 ContextMenu.....	198	第 11 章 子程序	263
8.6 实例.....	200	11.1 子程序.....	263
8.7 习题.....	203	11.2 内建转换函数.....	264
第 9 章 绘图	206	11.3 内建字符串函数.....	265
9.1 颜色设置和坐标.....	206	11.4 内建的数值函数.....	268
9.1.1 颜色设置.....	206	11.4.1 整数函数.....	268
9.1.2 坐标.....	209	11.4.2 随机数函数.....	269
9.1.3 绘图对象.....	210	11.4.3 数学类别(math class).....	270
9.2 常用的绘图方法.....	212	11.5 内建日期函数.....	273
9.2.1 绘图方法一.....	212	11.6 一般程序.....	275
9.2.2 绘图方法二.....	214	11.6.1 Sub 程序.....	275
9.2.3 绘图方法三.....	216	11.6.2 Call 语句.....	277
9.2.4 绘图方法四.....	219	11.6.3 传址调用 Call by address.....	278
9.2.5 绘图函数.....	221	11.6.4 传值调用.....	281
9.3 图文件的存取.....	222	11.6.5 Function 程序.....	284
9.4 实例.....	226	11.7 如何在程序间传递 数组数据.....	287
9.5 习题.....	229	11.8 实例.....	290
第 10 章 数组	231	11.9 习题.....	294
10.1 数组.....	231	第 12 章 模块与项目	296
10.1.1 一般数组的声明方式.....	231	12.1 方案与项目.....	296
10.1.2 指定初值的数组声明方式.....	233	12.2 模块 Module.....	296
10.1.3 数组索引指针的上下界值.....	233	12.3 多个模块的程序设计.....	297
10.1.4 For Each...Next 数组循环.....	234	12.3.1 程序的声明.....	297
10.2 二维数组.....	236	12.3.2 变量、常数、数组的声明.....	298
10.3 排序与搜索.....	239	12.3.3 如何建立多个模块的项目.....	299
10.3.1 冒泡排序法 Bubble Sort.....	239	12.3.4 多模块项目常用的 叙述与方法.....	303
10.3.2 顺序搜索法.....	241	12.4 多个项目的操作.....	308
10.3.3 二分搜索法.....	243	12.5 Shell 函数——调用 应用程序.....	313
10.4 静态数组与动态数组.....	245	12.6 习题.....	315
10.4.1 声明动态数组.....	245	第 13 章 数据库	317
10.4.2 保留数组 ReDim 前的元素内容.....	246	13.1 数据库介绍.....	317
10.4.3 声明静态数组.....	246	13.1.1 数据的搜集.....	317
10.4.4 清除数组.....	249		
10.5 列表框.....	249		
10.6 下拉列表框.....	253		

13.1.2 数据表与数据库	318	14.8 颜色对话框	373
13.1.3 数据的关联	319	14.9 习题	375
13.2 数据库系统的建立步骤	319	第 15 章 多媒体应用	378
13.3 使用 Access 来建立数据库	319	15.1 录音机程序	378
13.4 连接数据库	324	15.1.1 录放音	379
13.4.1 数据库连接原理	324	15.1.2 存盘	380
13.4.2 使用数据工具	325	15.1.3 读文件放音	381
13.5 SQL	332	15.1.4 产生声音特效	382
13.6 数据库实例演练	333	15.2 媒体播放程序	383
13.7 数据库的关联查询	340	15.3 API 播放语音函数	384
13.8 习题	345	15.4 调用 Windows Media Player... ..	391
第 14 章 文件与常用对话框	346	15.5 习题	394
14.1 文件的种类	346	第 16 章 网络应用技巧	395
14.2 文件的存取	347	16.1 网页浏览器的设计技巧	395
14.2.1 数据文件的作业方式	347	16.1.1 如何引用 Web 浏览器组件	395
14.2.2 文件打开与关闭	347	16.1.2 常用的属性、方法和事件	397
14.2.3 常用的文件函数	349	16.1.3 实例	397
14.3 顺序文件	350	16.2 网络应用程序	399
14.3.1 顺序文件数据的写入	350	16.3 Web Service	406
14.3.2 顺序文件数据的读取	352	16.3.1 如何建立 Web Service	407
14.4 结构	355	16.3.2 如何直接测试 Web Service	408
14.5 随机文件	360	16.3.3 建立窗口程序来测试	
14.5.1 随机文件的开文件和关文件	360	Web Service	409
14.5.2 随机文件的存取	361	16.3.4 实例	410
14.6 文件对话框	367	16.4 习题	411
14.7 字体对话框	373		

第 1 章 Visual Basic.NET 集成环境介绍

自从微软公司推出 Windows 后,由于它提供了易学易用的图形化界面,取代了传统 DOS 环境下要强行记忆的那些命令,而且在 Windows 下所开发的应用程序,操作方式都统一,无形中提高了亲和力。因此吸引了全球无数个人计算机用户的目光,形成计算机编程的主流。同时微软公司为了让学习程序设计的初学者也能轻松地在 Windows 环境下编写程序,于 1991 年推出 Visual Basic。初学者在 Visual Basic 的集成开发环境(IDE)下,通过所提供的工具在短短的时间内,不用写程序便能构建出需要的输入输出界面,只要着重在程序主体上的流程即可(这是传统 BASIC 所无法做到的),从而缩短了程序设计的时间。2002 年推出的 Visual Basic .NET(简称 VB .NET)在数据库和网络方面提供更多的高级功能,让 Visual Basic(简称 VB)程序设计能力更上一层楼。所以,VB .NET 是初学者在 Windows 下学习程序设计的最佳选择;至于专业的程序设计师也可使用 VB .NET 辅以 Excel、Access 等应用软件开发出需要的数据库管理系统,甚至可以使用 ActiveX、XML Web Services 等技术,进行 Internet 程序设计。

1.1 程序语言的发展

计算机系统是由硬件与软件组合而成。硬件是由中央处理器、存储器与输入输出单元等装置所构成的实体设备。软件就是程序,而程序是用来指挥计算机解决问题的指令集,它是使用程序语言编写而成的。通过编写程序来命令硬件,解决较复杂的计算和处理庞大的数据。

早期的程序语言就是机器语言,即使用 0 与 1 来编写程序。由于计算机只能处理 0 与 1,因此能马上执行,但是机器语言不易看懂且编写上很不方便,一般人很难接受。因此,希望能发展出一套比较容易为一般人所接受的计算机语言,只要通过语言翻译器就可以自动产生计算机能识别的机器语言。所以经过计算机专家改良而发展出来了以助记符号方式来编写的程序,这种语言称为汇编语言或低级语言。由于这种语言的语法相对机器语言易懂且易编写,但是它和机器有关,也就是机器不相同,语言的写法也不尽相同,因此可移植性(不同平台之间程序移植)不高,编写程序时也要先了解计算机内部缓存器和内存的运作才能编写,对一般的计算机初学者还是很困难。因此,此种语言只适用于计算机专业人员来编写有关计算机系统或输入输出接口的驱动程序。

针对以上缺点,计算机专家又发展出目前为大多数人所能接受的高级语言,由于此种语言可移植性高而且语法更接近人类的自然语言与数学表达式,因而得到了广泛的应用。高级语言中的 BASIC 语言是最容易为初学者所接受也是最普及的一种语言。

1.2 BASIC 的发展

自 1964 年, Dartmouth 学院的 John Kemeny 和 Thomas 设计出 BASIC 语言以来, 由于它的语法很接近人类所使用的自然语言和数学表达式, 广受计算机初学者的喜爱。早期的 BASIC 语言为了能在个人计算机的 64 KB 主存储器上执行, 并不注重结构化与模块化设计, 常造成程序维护上的困难。因此, 不为专业程序设计员所重视。由于近年来科技的不断进步, 个人计算机硬件, 如 CPU 速度不断地提升、主存储器容量大幅提高以及外围设备日新月异, 使得个人计算机应付功能强大的高级语言不再是梦想。

微软公司在 1988 年推出的英文版 Quick BASIC 4.5 版, 将 BASIC 语言推到最高峰。在 1991 年, 微软推出的中文版 Quick BASIC, 除了拥有一般高级语言的优点外, 程序也具有结构化、模块化、可编译成执行文件等特点, 更提供中文提示操作说明。国内计算机初学者纷纷以此作为初学程序设计的基础, 也有不少商业软件如计算机辅助教学软件等都使用 Quick BASIC 语言来设计, 其效果并不逊于其他的高级语言。

自从 Windows 推出以后, 人机之间的使用界面改善了很多, 大部分人只需经过短时间的学习就能操作计算机, 不用去记忆 DOS 命令而只需拖动鼠标即可。因此微软公司在 1991 年推出 Visual Basic(简称 VB)。它沿袭传统 BASIC 的精神, 是将 DOS 版的 Quick BASIC 改良的程序语言。

BASIC 程序的特点就是简单易学, 因此成为初学者入门的最佳选择。也正因为不严谨的传统, BASIC 语法在程序庞大时, 调试非常困难。另外 VB 虽然号称是“面向对象”的程序, 但却没有对象中非常重要的“继承”特性。微软公司有鉴于此, 重新修改 VB, 因而 2002 年推出了 VB.NET。

1.3 VB.NET 功能介绍

VB 发展至今已至 7.0 版, 微软将它与 C#、ASP、C++等一起置于 Visual Studio.NET 集成开发环境中。并在全球使用中文的地区推出 VB.NET 中文版。Visual 的中文意思是视觉, 它是指开发图形用户界面的方法, 其方法就是在建立输入输出界面时, 不必编写程序来描述界面组件的外观和配置, 只要使用工具箱的工具, 在程序设计阶段便可达成, 是属于一种直觉式的设计观念。VB 就是以 BASIC 语言为基础, 再加入许多和 Windows GUI 有关的功能。VB.NET 除了具有传统 BASIC 的优点外, 还具有下列传统 BASIC 所没有的功能:

- 基于对象(Object-Based)与面向对象的程序设计

VB.NET 执行的过程是以对象为主, 当使用鼠标或键盘触动某个对象时就执行该对象所设定的动作。传统 BASIC 程序设计是采用面向程序设计, 也就是说程序执行的流程在编写程序的时候就决定了; VB.NET 程序执行的流程是在执行时由用户来决定其流程。在 VB.NET 中更支持面向对象的相关语法与运作机制(如类、继承、多态和界面等), 让程序设计者在处理复杂的软件系统时, 通过面向对象的程序设计方式, 简化软件设计的复杂度, 提高软件开发

的效率并增强软件的可维护性。

- 输入输出界面设计简单

通过 VB.NET 所提供的工具,只要设置该工具的属性,不用写程序便能做到实用的输入输出界面,至于传统 BASIC 就必须编写繁复的程序,不具备一定编程经验的人是很难做到的。

- 用来制作多媒体软件

传统 BASIC 必须编写各个声卡的驱动程序才能发音,兼容性低。由于 VB.NET 是在 Windows 环境下执行,多媒体接口规格统一,因此编写程序时不必考虑兼容性问题。

- 多任务与多线程处理

在传统 BASIC 语言下,多个程序是无法同时执行的。在 Windows 环境,VB.NET 拥有多任务处理的功能,因此 VB.NET 能同时打开多个窗口执行不同的工作。在 VB.NET 中更支持执行线程的使用,这让程序设计师得以在同一个程序中进行异步运作,也就是说可以同时在一个程序中执行多项运算工作。

- .NET 框架提供许多对象类

.NET 框架中提供了一个丰富的基础类链接库 (fundamental class library),VB.NET 可以通过 .NET 框架这些类链接库来进行相关的处理,而不再需要直接调用 Windows API 函数,此举不但简化了程序设计的过程,更由于这一组类链接库是语言中立的,因此不论是使用 Visual Basic、Visual C# 或是 Visual C++ (甚至是其他支持 .NET Framework 的程序语言),使用的方式都是统一的,在进行语言转换时十分方便。

- 提供友好的人机界面

VB.NET 提供鼠标操作环境及键盘反应处理。

- 互联网

VB.NET 提供与 Internet 有关的控件,可以很容易地设计出有关互联网的应用程序。

- COM (Component Object Model)与 ActiveX

在 VB.NET 中除了可以使用 COM 组件 (软件组件,ActiveX 控件就是其中一种 COM 组件)进行 N-Tiers 程序设计外,更可以使用 VB.NET 来设计 ActiveX 控件,ActiveX 控件可以说是新版的 OCX 控件,可以置于网络服务器,用户只要通过 Internet Explorer 下载,就可以直接在浏览器中执行 VB.NET 应用程序。

- 动态链接库(DLL)

VB.NET 可以设计动态链接库(Dynamic Link Library, DLL),让程序的执行更有效率,更可以设计成网站应用程序组件。在 Visual Studio .NET 的组件版本架构管理下,可以改善 DLL Hell 现象 (因为 DLL 版本问题,而造成新旧版软件无法运行)。

- 与数据库连接

VB.NET 支持各类数据库,包括 Access、dBASE 与 SQL 等,并提供方便的数

数据库与控件的数据域位连接功能, 用户只要简单地设定控件与数据库的数据链路, 就可以制作出功能强大的数据库管理系统。VB .NET 提供了最先进的 ADO .NET, 更适合用于开发网络数据库与存取不同架构的数据库, 配合自制的 ActiveX Data Control, 让网络数据库程序的发展更加快速简单。

- IIS 应用程序

在 Windows NT + IIS 服务器上, 原本用户就可以运用 ASP (Active Server Page) 来开发各类 WWW 网站应用程序, 藉以取代传统 CGI (Common Gateway Interface) 程序, 在 VB .NET 中, 可以利用熟悉的 VB .NET 程序开发方法, 使用 WebClass 对象和 ASP 语法来开发 IIS 应用程序, 或者直接使用 ASP .NET, 快速编制出各类型网页应用程序, 例如在线订票系统、留言板、聊天室和公司库存管理系统等。

- 跨平台的程序开发

以往的 VB 经过编译, 会产生机器码然后保存成可执行程序。在 .NET Framework 架构下, 程序是编译成 MSIL (Microsoft Intermediate Language, Microsoft 中间语言), 然后也以可执行程序的形式存盘。MSIL 是一种接近机器码的中间语言, 通过各种平台的 JIT 编译器, 就可以跨平台执行。也就是说程序编译成 MSIL 后, 通过 PowerPC JIT 就可以在 PowerPC 下执行; 通过 x86 JIT 就可以在 x86PC 下执行。

- 多种程序语言集成

在 VS .NET 集成环境的架构下, Visual Basic、Visual C++、C# 和 Visual FoxPro 等支持 .NET 框架的程序语言可以直接互相引用彼此的类别程序, 因此在进行大型项目的开发时, 更有利于团队的分工合作。

1.4 对象和事件的基本概念

面向对象(Object-oriented)就是把现实世界中的现象, 以更接近自然的形式在计算机世界中反映出来。每一个对象都有属于自己的属性(Property)和方法(Method), 对象与对象间则是靠着方法来传递对象的消息(Message)。

- 对象(Object)

宇宙间任何具体或抽象的事物, 基本上都可视为“对象”。例如: 一个人可视为一个大对象; 身体中的头部、胸部、腹部以及四肢等的部位可视为小对象; 胸部又可再细分成心、肺等更小的对象。在 VB .NET 中的对象是一些程序代码和数据的组合, 对象可以被视为一个完整的单元。对象也可以是应用程序的一部分, 譬如一个控件或窗体, 整个应用程序也是一个对象。对象可以是:

- ◆ 窗体(Form)、打印机、屏幕、鼠标、键盘等。
- ◆ 利用工具箱的工具所产生的控件(Control), 如: 列表框、命令按钮、文本框、图像框等。
- ◆ 菜单、数据库管理员、报表设计等。

- 属性(Property)

在 VB 中的每个对象都有其特性, 利用这些特性来描述这个对象的外观, 例如: 窗体的特性包括窗体的大小、前景色、背景色和标题等, 将这些特性称为“属性”。每个对象的属性大都有其默认值, 对象在设计阶段或执行阶段(程序中), 都可以改变这些对象的属性值, 对象便会按照您的设定表现出来。例如: “球”是一个对象, 然而却能分辨出篮球和乒乓球, 因为它们的大小不同、材质不同、颜色不同、名称也不相同。所以, 不同类的球(对象)虽有共同的属性, 但属性内容不同因此可造出不同类的球。例如, 如图 1.1 所示, 不同标题名称的命令按钮, 即视为不同的对象。

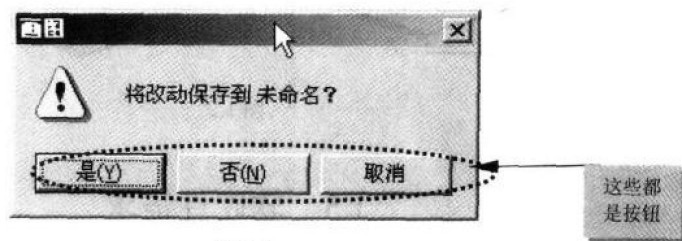


图 1.1

- 事件(Event)

在 VB .NET 中, 只要在某个有作用的对象上按下键盘、移动鼠标或单击鼠标, 都会产生一种反应, VB .NET 将此反应称为事件。当 VB .NET 收到属于某个对象的事件时, VB .NET 会根据此事件做适当的处理, 其处理的方式就是去执行该对象相对应的程序代码, 称为事件程序。而这些程序代码就是用 VB .NET 所提供的命令所设计出来的。

- 方法(Method)

所谓方法, 是指对象完成某件事或某项目标, 所采取的处理方式。所以在 VB .NET 中的每个对象都有一些方法用来改变该对象的外貌或内部行为, 有了对象的方法, 可以在不需要知道真正的数据结构(隐藏在对象内)以及其他相关的事项的情况下, 就可以满足需求。例如, 在窗体对象中提供了清图、打印、画点、画线等功能, 这些功能通称为方法。

- 事件驱动 (Event driven)

VB .NET 所提供的“事件驱动”就像消防队的工作一样。平时若没有状况发生时, 便在驻地做救火的准备或训练工作, 随时待命。当有状况发生, 知道火灾的地点、楼层的高度及该处是否有易燃物等。再携带必要的装备前往现场扑救。像这些因事件(火灾)发生, 才做适当处理的观念称为事件驱动。

同样, Windows 的应用程序也是如此, 当执行应用程序后, 该应用程序就进入待命状态, 当按下键盘或鼠标时就表示有事件要发生, Windows 将此事件的相关消息传给该应用程序, 该程序会根据送来的消息作适当的处理(图 1.2)。传统的程序设计, 程序设计者主导整个程序的流程, 用户只能按照既定的流程来操作, 是顺序的。VB .NET 的设计观念是将所有流程都交给用户来主控, 完全由用户对窗口内所做出的反应或系统事件来决定, 程序设计者不决定整

个程序的流程,是非顺序的。但是处理事件的方法和步骤到最基层时,仍然是顺序的,就好像消防队员进入火场灭火的基本操作方法是一定的。所以,善用 Visual Basic 的特点很容易设计出用户接口以及变化多端的事件驱动处理程序,可以大大地缩短开发应用程序的时间。

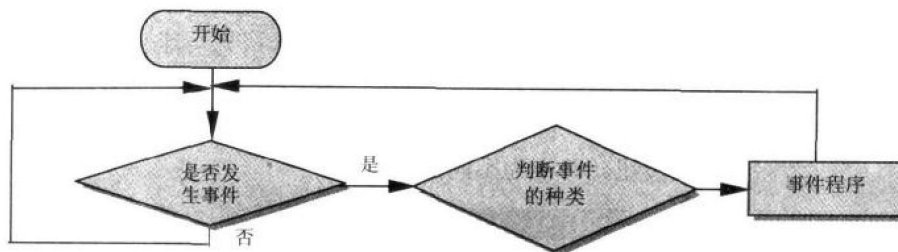


图 1.2

1.5 软硬件需求

在安装与激活 VB .NET 之前,先要了解 VB .NET 所需要的软硬件需求,以免执行时发生问题,以下是 VB.NET 在硬件及软件方面的基本需求:

- 硬件需求
 - ◆ CPU 至少 Pentium II 450 MHz 以上(建议用 Pentium III 733 MHz 以上)。
 - ◆ 彩色显示器: 800×600, 256 色(建议用真彩色 24 位)。
 - ◆ 硬盘: 至少 3GB。
 - ◆ 内存, 128MB RAM 以上(建议用 256MB RAM)。
 - ◆ 鼠标。
 - ◆ 只读光驱(CD-ROM)。
 - ◆ 16 位声卡、喇叭、麦克风, 多媒体配备。

- 软件需求

Microsoft Windows XP 或 Windows 2000 操作系统。

1.6 激活集成开发环境

集成开发环境(Integrated Develop Environment, 简称 IDE)是目前程序语言操作环境的潮流,早期有关程序的编写、编译、调试和运行等功能都互相独立,使用上很不方便;目前很多高级程序语言(像 VB .NET)都已经采用 IDE 方式,将编辑器(Editor)、编译器(Compiler)、链接器(Linker)、调试器(Debugger)、执行(Execution)文件存取等功能全部集成在一个窗口下,以利操作。由于 Visual Basic .NET 中文版是在 Windows 环境下使用的程序语言,因此必须先进入 Windows 操作系统才能顺利执行。

如果 Visual Studio .NET 已安装到 Windows 环境中,系统会将 Microsoft Visual Studio .NET 的快捷方式存放在【开始】|【程序】菜单下的 Microsoft Visual Studio .NET 子菜单中。

- (1) 单击【开始】按钮,选择【程序】菜单,再选择 Microsoft Visual Studio .NET 文件夹,最后执行 Microsoft Visual Studio .NET 命令,进入 IDE 集成开发环境的开始界面(图 1.3)。



注意: 本书中将上述连续操作简述为: 执行【开始】|【程序】| Microsoft Visual Studio .NET | Microsoft Visual Studio .NET 命令。

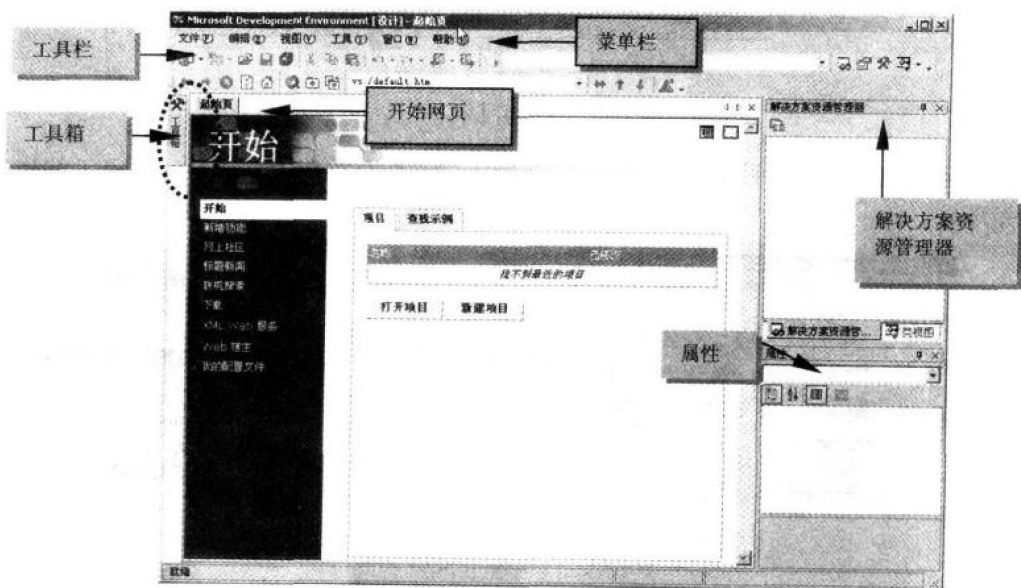


图 1.3 开始界面

- (2) 建立新的项目。

建立新项目有几个方法:

方法 1 单击【起始页】选项卡中的【新增项目】按钮。

方法 2 执行【文件】|【新建】|【项目】命令。

- (3) 此时会出现如图 1.3 所示的对话框。先在【项目类型】中选择【Visual Basic 项目】文件夹,然后在右边【模板】中选择【Windows 应用程序】(单击【Windows 应用程序】,表示要建立一个可以在窗口环境中执行的应用程序)。接着在【名称】文本框中输入项目名称(在本例中,项目名称为 Windows Application1),【位置】列表中可以单击 **浏览** 按钮选择项目保存的文件夹。设置完毕后,单击 **确定** 按钮就会建立新的项目,如图 1.4 所示。

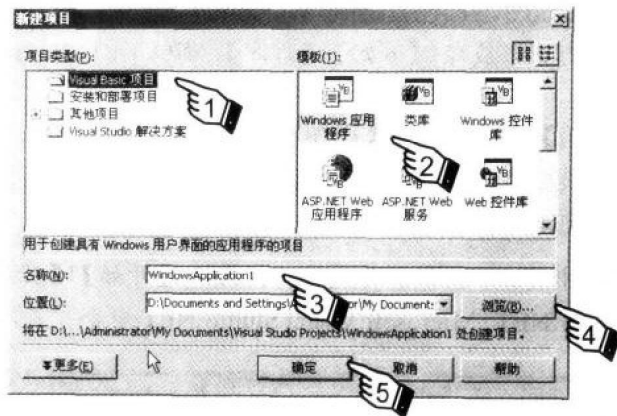


图 1.4

(4) 建立好新项目后, 出现如图 1.5 所示的界面, 称为 Visual Basic .NET 的集成开发环境。

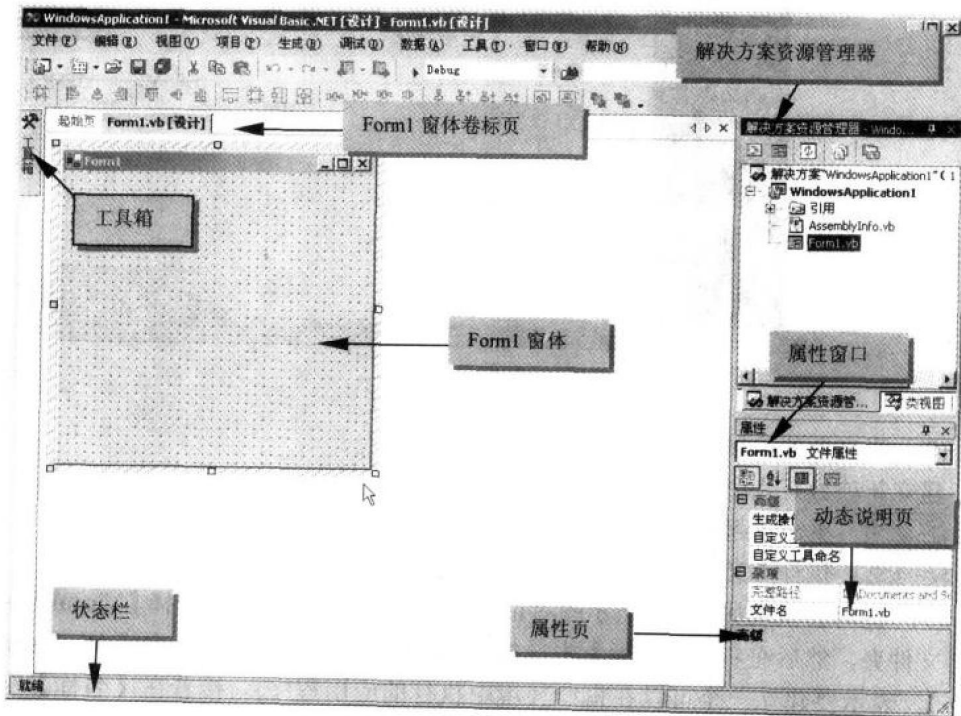


图 1.5

1.7 VB.NET 集成开发环境介绍

VB .NET 集成开发环境所提供的功能很多, 本书只针对初学者常用的部分做详细的介绍, 至于较复杂的功能请参考其他高级书籍。

1.7.1 主窗口

主窗口包含标题栏、菜单以及工具栏等如图 1.6 所示。



图 1.6

1. 标题栏

- 最小化按钮、 还原按钮、 最大化按钮、 关闭按钮操作说明：
 - ◆ 单击 关闭按钮，可关闭 VB.NET 集成开发环境。
 - ◆ 单击 最小化按钮，可将 VB.NET 集成环境缩至最小，并置于任务栏中。
 - ◆ VB.NET 集成环境窗口预设是最大化，若单击 还原按钮，主窗口可调大小，而 还原按钮变成 最大化按钮。
 - ◆ 若单击 最大化按钮，可将 VB.NET 集成环境的窗口放到最大，同时 最大化按钮变成 还原按钮。
- 版权商标“WindowsApplication1-Microsoft Visual Basic .NET[设计]”，其中 WindowsApplication1 为目前编辑的项目名称，而括号内的文字，用来指示 VB.NET 目前的编辑状态，其状态有下列 3 种：
 - ◆ [设计]：表示 VB.NET 目前处于设计程序的状态。
 - ◆ [执行]：表示 VB.NET 目前处于执行程序的状态。
 - ◆ [中断]：表示 VB.NET 目前处于暂停程序执行的状态。
- 在标题栏左侧是 VB.NET 窗口图标，主要功能是在鼠标不能正常操作时，可以按 + 选择菜单，再用键盘按键来控制常用的操作，譬如：关闭 VB.NET 窗口。

2. 菜单

菜单依操作性质的不同，共区分为 9 个菜单，如表 1.1 所示。

表 1.1 主菜单及其功能

菜 单	说 明
文件(F)	提供有关项目的保存、打印、编译可执行文件及结束 VB.NET 等命令
编辑(E)	提供剪切、复制、查找、替换、撤消等命令
视图(V)	提供打开 VB.NET 环境的各个窗口的命令
项目(P)	提供添加继承、窗体、组件、对象类别、用户控件等命令
生成(B)	提供生成解决方案、全部生成解决方案等命令
调试(D)	提供逐语句、逐过程、设断点、执行至光标处等执行调试的命令