

计算机基础课程系列教材

Visual Basic.NET

程序设计 基础教程

苏国彬 主编

- 为完全不懂编程的人员设计
- 洞悉Visual Basic.NET编程思想
- 控件使用与代码设计的完美结合
- 学习.NET平台程序设计的最佳切入点



机械工业出版社
China Machine Press

计算机基础课程系列教材

Visual Basic.NET

程序设计基础教程

苏国彬 主编



机械工业出版社
China Machine Press

本书详细介绍了Visual Basic .NET中文版的编程环境、程序设计基础知识、程序控制结构、数组、过程、图形绘制、文件操作、常用内部控件的用法、菜单、工具栏、对话框、多媒体、数据库以及网络等程序设计技术。读者可以掌握Visual Basic .NET程序设计的基本知识以及软件界面设计、多媒体程序设计、数据库程序设计、网络编程的方法。本书通过实例将控件使用与代码设计有机结合起来,并做到概念清晰,逻辑性强、层次分明、例题丰富。每章均附有一定习题。

本书既可作为大学、高职、高专及各类中等学校、社会培训班的教材,也可供具有Windows 操作系统初步知识的大中专院校学生甚至是初学编程的读者自学使用。

版权所有,侵权必究。

图书在版编目(CIP)数据

Visual Basic.NET程序设计基础教程 /苏国彬主编. -北京:机械工业出版社, 2002.9
(计算机基础课程系列教材)

ISBN 7-111-10818-3

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

中国版本图书馆CIP数据核字(2002)第062696号

机械工业出版社(北京市西城区百万庄大街22号 邮政编码 100037)

责任编辑:张鸿斌

北京第二外国语学院印刷厂印刷·新华书店北京发行所发行

2002年9月第1版第1次印刷

787mm×1092 mm 1/16·18.25印张

印数:0 001-4 000册

定价:29.00元(附光盘)

凡购本书,如有倒页、脱页、缺页,由本社发行部调换

前 言

在微软公司提出的.NET构想中,网络已经成为构建新一代操作系统的基础。因此,微软公司在以.NET作为发展的核心策略指导下,重新编写了Visual Studio.NET系列产品。Visual Basic .NET(简称VB.NET)是其中一个重要的组成部分,它支持许多新的面向对象语言的特性,并增添了一些新的数据类型,以及结构化错误控制等,与它的以前版本相比,可以说是已经脱胎换骨。

同时,VB.NET为我们提供了操控微软.NET平台的最简单的切入点。使用VB.NET,编程人员可以轻松开发基于Windows平台的完全富有新特色的应用程序,它必将对程序开发带来深远的影响。

本书共分为19章,详细地介绍了VB.NET中文版的编程环境、程序设计基础知识、程序控制结构、数组、过程、图形绘制、文件操作、常用内部控件的用法、菜单、工具栏、对话框、多媒体、数据库以及网络等程序设计技术。读者可以掌握Visual Basic程序设计的基本知识以及软件界面设计、多媒体程序设计、数据库程序设计、网络编程的方法。最后综合讲述了程序的调试、出错处理以及程序的打包发行等。

本书既可作为大学、高职、高专及各类中等学校、社会培训班的教材,也可供具有Windows操作系统初步知识的大中专院校学生甚至是初学编程的读者自学使用。

本书特色

为完全不懂编程的人员设计

本教程为学习VB.NET程序设计提供一个最简单易学的切入点。本书面向全无编程经验的读者,通过实例将控件使用与代码设计有机结合起来,并做到概念清晰、逻辑性强、层次分明、例题丰富。每章均附有一定习题。

控件使用与代码设计完美结合

书店中能够见到的Visual Basic程序设计教程类图书大多偏重讲述控件,轻视代码设计。这对于有程序语言学习经验的读者可能比较适用,但不适于第一次学习程序设计且选择了Visual Basic编程工具的人员,那样的书学完后感觉只是学到了一个花架子,并不适合学校选作教材,也不适合读者自学使用。

附赠上机实验指导多媒体教学光盘一张

本书附赠真正的上机实验指导多媒体教学光盘一张,教程全部采用Flash制作。共安排了15个上机实验。盘中安排的上机实验,以及多媒体教学光盘的演示教学,弥补了图书的缺陷,使读者可以在互动的多媒体教学中尽快掌握VB.NET程序设计的方法。最后将书中实例的源代码也整理在光盘中。

本书由苏国彬主编，孙威、陈峰、高海茹、赵海越等参加编写，张政勋、杨柏松、李祥军、张海波分别审校了全书，提出许多宝贵意见。参加本书上机实验指导多媒体教学光盘制作工作的有吴欣、辛庆祥、张桂欣、孙含、丁胜昔、苏国建、高海卿、李学军、孟磊等。由于本书成书仓促，难免有错误之处，欢迎广大读者与同行专家批评指正。我们的电子邮件地址是longtu_tech@sohu.com。

目 录

前言

第1章 程序设计概述	1
1.1 程序设计与常用程序设计工具概述	1
1.1.1 程序与程序语言	1
1.1.2 常用程序设计工具	2
1.2 什么是VB.NET	4
1.2.1 Visual Basic发展简介	4
1.2.2 VB.NET的特点	4
1.3 程序设计的一般方法	5
第2章 认识VB.NET的可视化编程环境	6
2.1 新建一个编程项目	6
2.2 认识.NET的集成开发环境	7
2.3 控件工具箱	8
2.4 解决方案资源管理器	9
2.5 属性窗口	10
2.6 动态帮助窗口	11
2.7 其他浮动窗口	11
习题	12
第3章 VB.NET可视化编程思想	13
3.1 .NET 框架概述	13
3.1.1 公共语言运行环境	14
3.1.2 .NET 框架基础类库	15
3.1.3 面向Windows程序的.NET编程	15
3.1.4 面向网络应用的.NET编程	15
3.2 认识对象	16
3.2.1 VB.NET的对象	16
3.2.2 用对象的事件来驱动程序	17
3.2.3 使用属性来操作对象	17
3.2.4 使用方法来操作对象	19
3.3 认识窗体对象	19
3.3.1 各种控件通用的属性	20
3.3.2 窗体的属性	20
3.4 向窗体中加入控件	20
3.4.1 控件的加入	21

3.4.2 控件的缩放和移动	21
3.4.3 控件的复制和删除	21
3.4.4 设置焦点在窗体各个控件之间传递的顺序	21
3.4.5 控件的布局	22
3.5 VB.NET的编程步骤	23
3.5.1 新建项目	23
3.5.2 添加控件	23
3.5.3 修改属性	23
3.5.4 编写代码	24
3.5.5 运行程序	24
3.5.6 进一步完善程序	25
3.5.7 生成可执行文件	25
习题	26
第4章 常用的Windows窗体控件	27
4.1 文本控件	27
4.1.1 使用TextBox控件设计文本框	27
4.1.2 使用RichTextBox控件设计高级文本框	31
4.1.3 设置热键	34
4.1.4 使用标签控件Label来显示文本	34
4.1.5 给标签中的文字设计超链接	35
4.2 列表选择控件	38
4.2.1 使用ListBox控件设计普通的列表框	38
4.2.2 使用CheckedListBox控件设计带复选框的列表框	41
4.2.3 使用ComboBox控件设计下拉列表框	42
4.2.4 使用DomainUpDown控件设计滚动列表框	43
4.2.5 使用NumericUpDown控件设计滚动数字列表框	43
4.2.6 使用ListView控件设计列表视图	45
4.2.7 使用TreeView控件设计树状列表	48
4.3 图形显示控件	50

4.3.1 使用PictureBox控件显示图片	50	第7章 VB.NET程序设计基础	87
4.3.2 使用ImageList控件设计图片列表	50	7.1 认识与理解数据类型	87
4.4 按钮和选择框控件	52	7.2 常量和变量	89
4.4.1 使用Button控件设计按钮	52	7.2.1 常量	89
4.4.2 使用CheckBox控件设计复选框	53	7.2.2 变量	91
4.4.3 使用RadioButton控件设计单选钮	55	7.2.3 数据类型的转换	93
4.5 时间日期控件	56	7.3 程序的各种运算	94
4.5.1 使用Timer控件定时	56	7.3.1 算术运算	94
4.5.2 使用MonthCalendar控件显示月历	57	7.3.2 字符串运算	94
4.5.3 使用DateTimePicker控件设计日期 时间选择器	58	7.3.3 关系运算	95
4.6 分组控件	58	7.3.4 逻辑运算与位运算	97
4.6.1 使用Panel控件设计分组面板	58	7.3.5 运算的优先级	99
4.6.2 使用GroupBox控件设计分组框	59	7.4 程序语句	100
4.6.3 使用TabControl控件设计选项卡	59	7.4.1 程序中的语句	100
习题	59	7.4.2 赋值语句、注释语句和End语句	100
第5章 设计菜单、工具栏和状态栏	60	7.5 程序的编写规则	102
5.1 设计菜单	60	7.5.1 VB.NET的保留字	102
5.1.1 设计普通菜单	60	7.5.2 变量规则	103
5.1.2 设计右键快捷菜单	65	习题	106
5.2 设计工具栏	65	第8章 程序控制结构	108
5.3 设计状态栏	68	8.1 选择结构	108
习题	69	8.1.1 If...Then语句	108
第6章 设计对话框	71	8.1.2 Select...Case语句	111
6.1 使用对话框控件设计对话框	71	8.1.3 选择结构的嵌套	113
6.1.1 使用ColorDialog控件设计颜色 对话框	71	8.2 循环结构	115
6.1.2 使用FontDialog控件设计字体 对话框	73	8.2.1 While语句	115
6.1.3 使用OpenFileDialog控件设计打开 文件对话框	74	8.2.2 Do...Loop循环结构	116
6.1.4 使用SaveFileDialog控件设计另存 文件对话框	76	8.2.3 For...Next语句	118
6.1.5 使用PrintDialog控件设计打印对话框	78	8.2.4 循环的嵌套	120
6.2 模式对话框	79	8.3 跳转结构	121
6.3 使用内部对话框设计对话框	79	习题	122
6.3.1 用MsgBox控件显示信息对话框	80	第9章 数组、枚举与结构	124
6.3.2 用InputBox控件来获得用户输入	83	9.1 数组	124
6.4 自定义对话框	84	9.1.1 一维数组	124
习题	86	9.1.2 多维数组	126
		9.1.3 改变数组大小	128
		9.1.4 数组的数组	128
		9.1.5 For Each...Next 语句	129
		9.2 枚举	130
		9.3 结构	131

9.3.1 结构的定义	132	11.2.6 应用Rectangle类	174
9.3.2 结构和其他编程元素	133	11.2.7 用Inherits语句实现继承	177
9.3.3 用结构定义自己的数据类型	134	11.2.8 共享类成员	181
习题	134	11.2.9 用Interface语句定义接口	183
第10章 过程	135	11.2.10 用Implements关键字关联类和 接口	184
10.1 Sub过程	135	11.3 结构和类	185
10.1.1 Sub过程的定义	135	习题	186
10.1.2 简单文本编辑器中的Sub过程	136	第12章 文件操作	188
10.2 Function过程	140	12.1 流的基本概念	188
10.3 参数传递与Function过程返回值的 深入讨论	142	12.2 读写文本文件	189
10.3.1 深入参数列表	142	12.3 读写二进制文件	192
10.3.2 参数的传递	143	12.4 文件夹的相关操作	194
10.3.3 Function过程的返回值	146	12.4.1 取得和显示某文件夹中的所有子 文件夹	196
10.4 过程的嵌套与递归调用	147	12.4.2 移动文件夹	197
10.4.1 过程的嵌套	148	12.4.3 新建文件夹	198
10.4.2 过程的递归调用	148	12.4.4 删除文件夹	198
10.5 变量的作用域与生命周期	149	12.5 文件的相关操作	198
10.5.1 作用域	149	12.5.1 取得和显示文件的名称、属性、 修改时间和大小	200
10.5.2 生命周期	150	12.5.2 复制文件	201
10.6 常用内部函数	150	12.5.3 移动文件	201
10.6.1 输入输出函数	150	12.5.4 新建文件	201
10.6.2 类型转换函数	152	12.5.5 删除文件	202
10.6.3 字符串操作函数	154	12.6 “我的资源管理器”实例	202
10.6.4 数学函数	155	习题	208
10.6.5 Rnd随机函数	158	第13章 图形绘制与使用图像	209
10.6.6 日期函数	158	13.1 GDI+绘图基础知识	209
10.6.7 其他常用函数	161	13.2 认识坐标系统	211
习题	162	13.3 设置所要绘制图形的属性	211
第11章 面向对象	163	13.3.1 设置画线的宽度和样式	211
11.1 面向对象基本概念	163	13.3.2 设置填充属性	212
11.1.1 对象	163	13.4 设置绘图的颜色和文字属性	214
11.1.2 类	163	13.4.1 设置绘制图形的颜色	214
11.2 面向对象编程方法的实现	164	13.4.2 设置文字的字体、大小和样式	214
11.2.1 用Class语句创建类	164	13.5 绘制图形	215
11.2.2 用Property语句创建属性	166	13.5.1 绘制直线	215
11.2.3 用Sub和Function创建方法	168	13.5.2 绘制矩形方法和填充矩形	215
11.2.4 用Event语句声明事件	169	13.5.3 画弧	216
11.2.5 对象的生命周期——构造器与析 构器	170	13.5.4 绘制椭圆	217

13.5.5 绘制文字	217	15.8.2 浏览全部数据	251
13.5.6 使用图像	218	15.9 高级查询	251
13.6 虚拟时钟的实例	218	15.9.1 使用DataGrid控件显示查询结果	252
习题	221	15.9.2 使用DataView的筛选功能来实现 查询	253
第14章 多媒体程序设计	222	15.9.3 实现多条件高级查询	253
14.1 多媒体控制命令	222	习题	256
14.2 使用多媒体控件	223	第16章 设计网络应用程序	257
习题	226	16.1 网络基础知识	257
第15章 访问数据库	228	16.2 设计发送电子邮件程序	257
15.1 创建数据库介绍	228	16.2.1 定义邮件内容	258
15.1.1 什么是数据库	228	16.2.2 发送邮件	259
15.1.2 如何创建数据库	228	16.2.3 发送邮件附件	260
15.2 SQL	230	习题	261
15.2.1 SELECT语句	230	第17章 程序的调试与异常处理	262
15.2.2 INSERT语句	232	17.1 程序中的错误类型	262
15.2.3 UPDATE语句	232	17.1.1 语法错误	262
15.2.4 DELETE语句	233	17.1.2 运行错误	262
15.3 ADO.NET概述	233	17.1.3 逻辑错误	262
15.4 创建数据库应用程序项目	234	17.2 程序的调试	263
15.4.1 创建数据库	234	17.2.1 认识Visual Studio.NET的工作模式	263
15.4.2 设计所要实现的功能	235	17.2.2 使用调试工具栏调试程序	263
15.4.3 设计窗体	235	17.2.3 设置和删除断点	264
15.5 连接数据库	237	17.2.4 使用调试窗口	265
15.5.1 使用SqlConnection对象连接SQL Server数据库	238	17.3 异常处理	266
15.5.2 使用OleDbConnection对象连接 其他数据库	239	17.3.1 非结构化异常处理语句	267
15.5.3 使用.NET环境建立到通讯录的 连接	239	17.3.2 结构化异常处理语句	269
15.6 创建数据适配器访问数据库	240	17.3.3 结构化异常处理与非结构化异常 处理的比较	272
15.6.1 创建OleDbAdapter对象	240	习题	273
15.6.2 使用.NET环境创建数据适配器并 生成数据集	241	第18章 部署VB应用程序	274
15.6.3 数据绑定与浏览	241	18.1 .NET概述	274
15.7 添加、修改与删除记录	242	18.2 向解决方案中添加部署项目	274
15.7.1 添加新记录	243	18.3 添加文件、输出或其他项	275
15.7.2 修改记录	246	18.4 文件安装管理	277
15.7.3 删除记录	248	18.5 注册表设置管理	277
15.8 简单查询	249	18.6 用户界面管理	279
15.8.1 按姓名查询	250	18.7 自定义操作管理	279
		18.8 启动条件管理	280
		18.9 生成安装程序	281
		习题	281

第1章 程序设计概述

本章简单讲述了程序设计的一般问题，介绍了什么是程序和程序语言，以及常用的程序开发工具，特别介绍了几种典型的编程工具，最后还提出了编写程序需要注意的几点问题。

1.1 程序设计与常用程序设计工具概述

1.1.1 程序与程序语言

通常我们所说的语言是指人们在日常生活中用来进行交流的自然语言，比如汉语、日语、英语等，它是人与人之间的交流工具；而程序设计语言则完全不同，它是一种完全人造的语言，是与计算机打交道的一个媒介和工具。程序设计语言也常常被称为“编程语言”，在不引起混淆的时候简称为“语言”。

计算机是人类发明的一种自动运算机器，它能够完成的工作就是计算；程序设计语言可以看做是人们用来向计算机发送指令的工具。程序语言的突出特点就在于：不但人能够读懂它，掌握它，用它来描述自己所需要的计算过程；而且计算机也可以“懂得”它，按照人们用程序语言给出的关于计算过程的描述去运行，完成人们所需要的计算工作。所以程序语言可以看成是人与计算机“交流”的媒介：人们用这种方式指挥计算机完成各种各样的特定工作，或者说，完成各种各样的计算。

程序是一系列的计算机指令，命令计算机去完成某一个工作，从而来实现一定的结果。编写这些计算机指令的工作就称为编程，而这些指令集合就是程序，所用指令的描述方式或者描述规则就是程序语言。

进行程序设计需要有确定的描述方式，说得更准确一点，需要的是一种计算机能够处理的、意义清晰的，而且人们使用起来也比较方便的描述计算方式，也就是需要一种好的语言。

在计算机诞生之初，人们使用二进制形式的机器语言编写程序。二进制形式的机器语言不但书写不方便，冗长，而且正确性难以保证。后来人们发展了语言符号形式的、使用起来相对容易一些的汇编语言。用汇编语言书写的程序，计算机不能直接执行，需要有专门的软件（汇编系统）进行加工，把这种程序翻译成机器能够理解的二进制的指令形式，然后才能在计算机上使用。汇编语言由于其自身的特点，再加之编译速度也相当快，因此大量应用起来，但是其可读性仍然不是很好，且不太容易编写非常大型的程序。1954年诞生的第一个高级程序设计语言——FORTRAN，宣告了程序设计的一个新时代的开始。

高级程序语言更接近人们的习惯，更容易被人们接受，这使得更多的人参与到程序设计中来。人们用高级语言书写程序，工作效率更高，这样就大大推动了计算机行业的发展，并开发出了更多的应用系统。从FORTRAN的诞生到现在，人们已经提出了上千种各类语言，而且绝大部分都是高级语言。现在通常所指的编程语言都是高级语言。

目前全世界使用比较广泛的语言有Basic、FORTRAN、C、C++、PASCAL、COBOL、Java等等。这些语言有很多类似的地方，当然也有各自的特点。

1.1.2 常用程序设计工具

有了一种语言，一般都需要一种编写程序的软件，用来编译和连接程序。一种语言，特别是高级语言，并不能写出计算机可直接执行的代码，所以需要程序设计工具来完成转换任务，让计算机能够读懂程序员所编写的程序。随着程序设计的发展，使用程序设计工具不但能够完成编译，也就是转换成计算机可识别的代码，而且还可以检查错误，以及实现可视化编程。关于可视化编程，用户可以参看第3章。

比如大家经常听说的Turbo C、Visual C++、Visual Basic、Delphi等，都是程序设计工具，为程序设计提供了一个良好的编程环境。通常，一种程序设计工具都是针对一种特定语言的。比如Microsoft Visual Basic所使用的语言就是Basic语言；Microsoft Visual C++、Borland C++ Builder使用的是C++语言，同时兼容C语言；Delphi使用的则是PASCAL语言等等。

下面是几种常用编程工具的简单介绍。

1. 数据库编程——Visual FoxPro和PowerBuilder

这两个软件主要都是用来进行数据库编程的，但是它们又不完全相同。

Visual FoxPro（简称VFP）是微软公司的产品，作为dBase、Sybase、FoxBase系列数据库管理系统，是在FoxPro 2.6基础上推出的Windows多任务多环境下的数据库编程工具。它的出现为MIS(管理信息系统)、TIM(技术信息管理)、PDM(产品数据管理)、财会业务、Internet数据传递等提供了强有力的开发平台。它主要是用来管理数据库的，比如建立数据库、编辑数据库内容、查询数据、制作报表等等。

作为桌面型数据库应用的开发工具，小巧玲珑的Visual FoxPro拥有相当大的用户群。很多程序员从大学时学习dBase、VFP等课程，并由此开始接受数据库基础知识教育。和Oracle、Sybase等大型数据库系统相比，FoxPro对软硬件系统和开发者的能力要求低，开发周期短，虽然难以胜任大型关键性系统，但在开发日常管理的应用方面却是得心应手，如小型账务管理、物品管理、试题系统等。

PowerBuilder（简称PB）是美国PowerSoft公司(已被Sybase收购)推出的Windows下专门设计开发客户/服务器数据库应用程序的功能强大的可视化开发工具。

PB开发数据库系统与常见的FoxPro有着明显的区别，它不像FoxPro那样既要负责创建数据库，又要设计界面、对数据进行操作。相反，PB认为，创建数据库是数据库管理系统的事，它仅仅负责设计界面，对数据进行操作和控制。从这种意义上讲，PB更像一个数据库前端开发工具。当然这并不代表PB对数据库没有一点设计能力。在PB的数据库画板中同样可以创建数据表（而不是数据库）。而且PB还附带了一个SQL Anywhere 数据库系统，通过ODBC，PB可以创建数据库与表。

由于PB强大的功能和鲜明的特点，因此日益成为大中型C/S（客户/服务器）数据库应用系统的首选开发工具。在最新的8.0版本中，更是（同时也极大）增强了对Internet/ Intranet数据库以及多种操作系统的支持，使其在新一轮竞争中继续保持优势。

值得一提的是，PowerBuilder是少见的一个能够支持跨平台客户端应用程序开发的工具。除Windows 外，它还支持Macintosh和Unix平台，而且PowerBuilder可以在它所支持的所有平台上执行同一个P-code代码的文件。PowerBuilder提供优秀的报表生成方法、完整的在线帮助以及丰富的Internet访问功能和组件特性。

2. 一般Windows应用程序编程——Visual C++和C++ Builder

Visual C++和C++ Builder都是以C++语言为基础的,同时兼容C语言,并且都有可视化编程界面,都是主要用来编写Windows程序的。当然,C++是一种通用语言,所以这两个工具几乎可以编写所有的应用程序,而且功能强大。利用这两个工具不但编译的时候速度快,而且编写出来的程序非常稳定。所以,编写正式的大型复杂应用程序时,很多程序员都选用这两个工具中的一个。

Visual C++是微软公司的产品,自然跟微软的很多产品结合紧密,而且在ActiveX方面性能优越。不过它使用起来比较复杂,一般初学编程的人使用起来可能有一些困难。

C++ Builder是Borland公司推出的一个基于C++语言的编程工具,跟Visual C++非常类似,也同样有着强大的功能。不过它没有Visual C++那么复杂,可视化操作更加简单快捷。

3. 应用程序和数据库编程的结合——Visual Basic和Delphi

Visual Basic和Delphi都可以编写一般Windows应用程序和客户/服务器数据库应用程序,而且编程过程都不是非常复杂。

Visual Basic是由Microsoft(微软)公司提供的一种可视化开发工具。“Basic”指的是BASIC(Beginners All-Purpose Symbolic Instruction Code,初学者通用符号指令代码)语言,一种在计算技术发展历史上应用得最为广泛的语言。Visual Basic在原有BASIC语言的基础上进一步发展,至今包含了数百条语句、函数及关键字,初学者只要掌握几个关键字就可以建立实用的应用程序。现在,微软公司推出了最新版本的Visual Basic 7.0,也就是通常所说的VB.NET,它与以前的版本相比有了很大的提高和变化,其编程语言跟Visual C++更加接近,网络功能更加强大,面向对象性更好。

尽管在众多的可视化编程工具中,Visual Basic在技术上并不领先,但对一个初学者和非专业开发人员来说,Visual Basic还是最好的。原因如下:

- 1) 简单易学,在美国10岁的小孩便能利用Visual Basic编写小的应用程序,它的易用性由此可见一斑。用Visual Basic编程不需要掌握太多的计算机知识,只要懂得用画笔和Windows的使用方法再加上一点Basic语言知识便可。

- 2) 整个软件(包括联机帮助)已全部汉化。使用中文版进行编程,大量可供参考的中文帮助至少可以提高数倍的学习效率。

- 3) 半智能化的编码器,能帮我们完成代码和即时查询函数用法,纵观现在所有的编程工具,可以说Visual Basic的编码器是最好的!

- 4) Microsoft公司的大力推广。靠Basic起家的Microsoft对Basic总是“一往情深”,在它的Office系列中把Basic作为一个内置的编程语言(即VBA),这就意味着,如果熟悉Visual Basic,便能在Word、Excel、Access等产品中进行二次开发。同时,VB Script也是Visual Basic的一个变种,主要用于编写广泛使用的ASP网页。随着Visual Basic和微软.NET的推广应用,相信Visual Basic一定会大显身手。

- 5) 世界上最多的技术支持,你会发现Visual Basic的问题是最容易被解答的。

Delphi是Borland公司推出的一种强有力的编程工具,它提供基于PASCAL语言的可视化编程环境。

Delphi是面向对象的可视化编程工具。即用Delphi的可视化编程环境,结合面向对象的PASCAL语言的编程技巧,开发功能强大的Windows应用程序和数据库应用程序。

目前计算机软件开发工具分为两个体系,其中一个体系是使用传统的设计语言,如PASCAL、C等,主要是开发通常的Windows应用程序。另一个体系则是通用的数据库管理软件开发领域,主要是用来开发数据库应用程序。如果使用数据库语言进行传统的数据运算编程,其实现过程可能会极为复杂;然而使用传统语言编写数据库管理软件,做起来也同样十分困难。Delphi 结合两大体系的优点,使我们根本不需要编写任何的PASCAL 代码便可以创建一个简单的数据库应用。

1.2 什么是VB.NET

1.2.1 Visual Basic发展简介

VB.NET全称是Visual Basic.NET,也称为Visual Basic 7.0,起源于BASIC语言。BASIC是国际上广泛使用的一种计算机高级语言。BASIC是一种易学易用的高级语言,非常适合初学者学习使用,目前仍是计算机入门的学习语言之一。

1991年,微软推出了Visual Basic 1.0版,这在当时引起了很大的轰动。许多专家把VB的出现当做是软件开发史上的一个具有划时代意义的事件。其实,以我们现在的目光来看,VB1.0的功能实在是太弱了。但在当时,它是第一个可视化的编程软件。这使得程序员喜出望外,都尝试在VB的平台上进行软件开发。微软也不失时机地在四年内接连推出VB2.0、VB3.0、VB4.0三个版本。并且从VB3.0开始,微软将Access的数据库驱动集成到了VB中,这使得VB的数据库编程能力大大提高。从VB4.0开始,VB也引入了面向对象的程序设计思想。VB功能强大,学习简单,而且,VB还引入了“控件”的概念,使得大量已经编好的VB程序可以被我们直接拿来使用。如今,VB已经发展到了最新的VB.NET,即VB7.0。

通过几年的发展,VB已成为一种真正专业化的开发语言和环境。用户认为可用Visual Basic快速创建Windows程序,在现在还可以编写企业水平的客户/服务器程序及强大的数据库应用程序,Visual Basic新版本中还有更多可用功能。

1.2.2 VB.NET的特点

网络已经给我们这个时代打下了深刻的烙印,它是一个巨大的里程碑,同时又是一个巨大的指针,标示着未来技术的发展方向。在微软公司提出的.NET战略构想中,网络本身已经成为构建新一代操作系统的基础。在.NET核心策略指导下,Visual Studio.NET系列产品在这样一个新的高度上被重新编写。

在开发VB.NET时,微软充分地调查了全世界范围内Visual Basic开发人员的需求。Visual Basic现在已经真正成为面向对象以及支持继承性的语言。窗体设计器支持可视化继承,并且包含了许多新的特性,比如自动改变窗体大小、资源本地化支持、数据类工具内在支持XML数据。VB.NET直接建立在.NET的框架结构上,因此开发人员可以充分利用所有.NET平台特性,也可以与其他的.NET语言(比如C++、C#)交互。另外,VB.NET为Windows应用程序提供了XCOPY部署,开发人员不再需要为DLL的版本问题担忧。对于VB.NET来说,“DLL噩梦”(DLL Hell)已经成为历史。与以前的版本相比,它可以说是已经脱胎换骨。

微软几乎将所有最先进的技术都包含在VB.NET中了,而且简单易学。可以说VB.NET为我们提供了操控微软.NET平台的最简单的切入点。选择VB.NET作为编程的入门语言,不但可以

学会编程方法,而且还可以掌握非常实用的编程语言和编程工具,并且可以在相当长的时期内拥有当前最先进的编程技术,并能利用它轻松开发基于Windows平台的完全富有新特色的应用程序。总之,它必将对程序开发带来深远影响。

1.3 程序设计的一般方法

熟悉了一种程序设计语言,应该如何编写程序呢?程序设计是一种智力劳动,编写一个程序就是解决一个问题。在学习程序设计时,程序都比较简单,但是实际应用中要复杂得多,不过基本的方法还是一样的。一般来说,解决问题的过程分为以下几个步骤:

1. 分析问题,根据其特点,选择一个合适的开发工具,然后设计一个解决问题的具体方案,包括拟定程序的流程图。

2. 设计界面,比如在VB.NET中,应根据程序的特点创建窗体,并且精心设计这些窗体,把各个功能尽量分开,模块化。最主要的就是界面要人性化,以方便人们使用这个程序。

3. 利用程序语言在界面中加入代码,这样程序就能够真正运行起来。界面如同一个没有任何软件的计算机(裸机),不能做任何事情,而程序代码犹如是计算机的软件,使计算机真正“跑”起来,做很多的事情。

4. 在计算机上调试这个程序,运行并查看是否能够真正解决问题。

如果第4步发现了错误,那么就需要仔细分析错误的原因,然后回到前面的步骤去纠正错误。如果发现程序写的有问题,那么就需要修改程序,然后重新运行检查;如果发现是问题求解方案有误,那么就要修改方案,并重新编写程序,这样一直下去直到程序最后正确完成所提出的问题为止。

对于一个初学者而言,需要学习的东西很多,包括知识的记忆和灵活掌握,解决问题的思维方法,具体处理问题的手段和技巧,及其他许多实际的工作、操作技能问题。在学习编程的过程中,需要注意以下几个方面:培养问题分析能力,特别是从计算过程和程序的角度分析问题的能力;掌握所使用的编程语言,熟悉语言中各种结构和有关规定,包括它们的形式和意义;熟悉所使用的工具和环境;编写一个好的程序;检查和排除程序错误的能力。

另外,人们还关注写出的程序是否易于阅读和理解,这就要求程序的可读性好,要充分使用程序的注释功能,为编写程序的程序添加注释。当程序所处理的问题中有些条件或要求(如操作平台或某些参数)发生变化时,程序是否容易修改以满足新的要求,这就涉及到程序的可移植性的问题。当进行团队开发程序时,由于每人负责一个部分,但是这些部分又不是独立的,因此程序的可读性、可扩展性、可移植性就显得尤为重要。

第2章 认识VB.NET的可视化编程环境

在上一章里，我们对程序设计的基本概念和方法做了大致的介绍，并介绍了常用的程序设计工具。本章，我们将介绍VB.NET的可视化编程环境，然后在下一章，讨论如何使用VB.NET的可视化编程环境来编程。

事实上，VC.NET、VB.NET和C#等.NET语言使用的是同一个可视化编程环境，即.NET集成开发环境（IDE，Integrated Development Environment）。这就给程序员提供了很大的便利，只要熟悉一种语言的开发环境，也就熟悉了其他几种语言的开发环境。这也体现了.NET的统一开发语言的思想。

2.1 新建一个编程项目

假设我们已经安装完成了Microsoft Visual Studio .NET中文正式版（安装的过程并不复杂，只要按照提示一步一步地做就可以了）。现在打开Microsoft Visual Studio .NET，单击桌面左下角的“开始”按钮，选择“程序”→“Microsoft Visual Studio .NET”→“Microsoft Visual Studio .NET”命令，打开如图2-1所示的显示了.NET开发环境起始页的窗口。

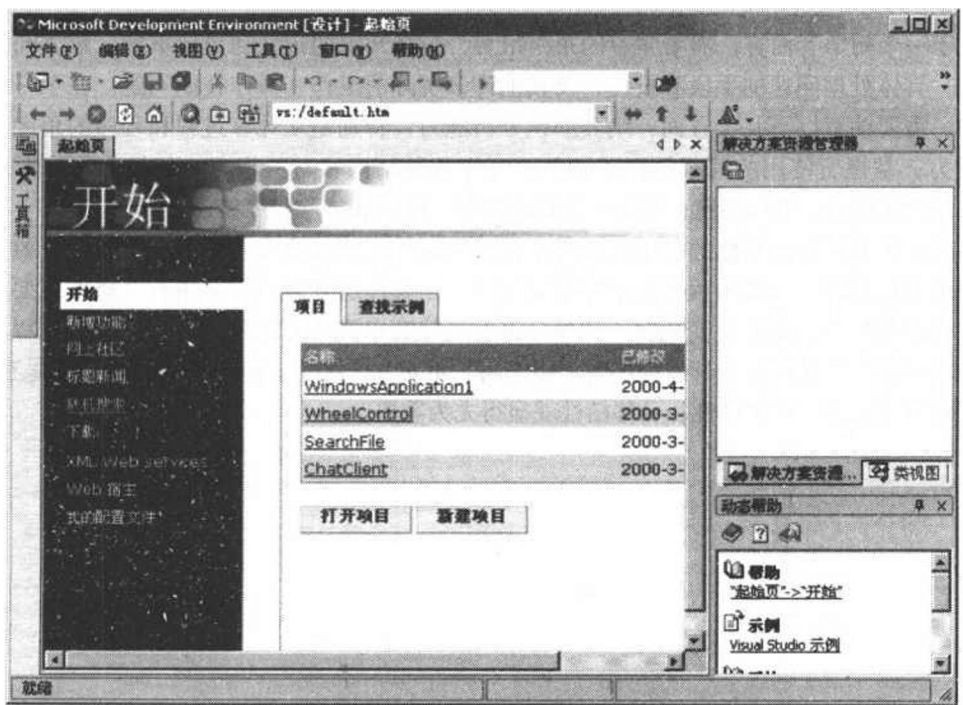


图2-1 显示了.NET开发环境起始页的窗口

因为我们要介绍的VB.NET的可视化编程环境是用VB.NET编程时的环境，所以要先新建

一个程序，在.NET开发环境中，所有的程序都是属于一个项目文件的。一个项目文件里通常包含了窗体文件和其他一些相关文件。所以，编程的第一步是新建一个项目或者打开原来已有的项目。单击起始页中列出的几个项目或者单击起始页中的“打开项目”按钮可以打开项目。假设这是我们第一次使用.NET开发环境，就要新建一个项目。新建项目的操作是：单击起始页中的“新建项目”按钮或者选择菜单命令“文件”→“新建”→“项目”，打开如图2-2所示的“新建项目”对话框。

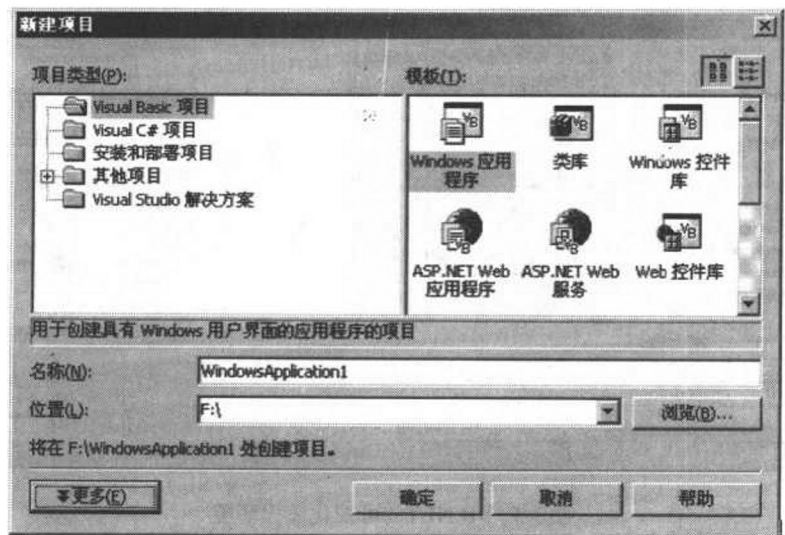


图2-2 “新建项目”对话框

在对话框的左边窗格中选择项目类型为“Visual Basic项目”，然后在右边窗格中选择“Windows应用程序”模板。在这个模板中有一些初始化代码，它自动添加一个窗体。如果选择“空项目”模板，则要自己编写初始化代码。如果编写网络应用的程序，可以选用“ASP.NET Web应用程序”模板或者“ASP.NET Web服务”模板。

接着，在“名称”文本框中填入项目名称，在“位置”文本框中填入项目保存的位置。然后单击“确定”按钮完成项目的新建。这时，.NET开发环境类似于图2-3，只是在窗口中间的工作区中是一个空白的窗体。

下面我们来详细介绍图2-3所示的.NET的集成开发环境。

2.2 认识.NET的集成开发环境

.NET的集成开发环境是如图2-3所示的窗体。和常见的Windows窗体一样，.NET的集成开发环境也有自己的菜单栏和工具栏按钮。窗体的中央是用来编辑程序的界面和代码的工作区，除此之外，窗体四周有很多浮动窗口。

我们不会逐一地介绍.NET集成开发环境的每个菜单项和每个工具栏按钮的含义，因为这样做没有必要，很多菜单命令和工具栏按钮的含义是一目了然的。而那些复杂的、令人费解的菜单命令和工具栏按钮会在用到它们的时候再介绍。在本节中我们要介绍的是.NET 集成开发环境中的浮动窗口，因为这些浮动窗口很重要，而且很快就要用到它们。因为刚刚开始学习VB.NET，对于VB.NET中很多重要的内容还不了解，所以我们不会很详尽地介绍编程环境

中的每个浮动窗口。而是只对其中一些比较重要的窗口详细地介绍；而对于一些不很重要的浮动窗口，只略提一下，在后面讲解编程的过程中再进行详细的说明。



图2-3 VB.NET的可视化编程环境

在.NET 的集成开发环境（IDE）中，可以显示众多辅助性的浮动窗口，为了方便使用者，通常几个浮动窗口重叠在IDE中的同一个位置，通过切换顶部或者底部的标签就可以在不同的浮动窗口之间切换。可以选择“视图”菜单下的相关命令显示和关闭这些浮动窗口。单击浮动窗口右上角的按钮可以把窗口固定在所在的位置，这时按钮变成，再次单击这个按钮可以使窗口重新浮动起来。单击窗口右上角的按钮可以关闭该窗口。

下面简要介绍一下显示在图2-3中的几个主要的浮动窗口。

集成开发环境中间的工作区通常用来显示窗体设计窗口和代码编辑窗口（当然你也可以把别的浮动窗口拖动到这个位置上），在这个例子里，单击“mycalculator.vb[设计]”标签可以切换到代码编辑窗口，单击“mycalculator.vb”标签可以切换到窗体设计窗口。其中mycalculator就是当前的程序项目的名称。

工作区左侧的两个重叠的浮动窗口是工具箱和服务器资源管理器。可以单击窗口下面的标签在两个窗口之间切换。工作区右侧上方的两个重叠的浮动窗口是解决方案资源管理器和类视图。工作区右侧下方两个重叠的浮动窗口是属性窗口和动态帮助。工作区下方的浮动窗口是输出窗口。

下面逐一介绍各个浮动窗口。

2.3 控件工具箱

控件工具箱可列出所有可用控件。所谓控件就是窗体中的相对独立的部分，比如按钮、标签、文本框等都是控件。当然这只是对控件的一个描述性的定义，控件的准确定义要在了