



普通高等教育“十一五”国家级规划教材

Visual Basic .NET

程 序 设 计

王李进	薛岚燕	潘晓文	主 编
		赵艳萍	副主编
		宁正元	主 审



中国水利水电出版社

www.waterpub.com.cn

普通高等教育“十一五”国家级规划教材

Visual Basic .NET 程序设计

潘晓文 主 编

王李进 薛岚燕 赵艳萍 副主编

宁正元 主 审

中国水利水电出版社

内 容 提 要

本书是普通高等教育“十一五”国家级规划教材,系统介绍 Visual Basic .NET 应用程序设计技术,内容包括: Visual Basic .NET 集成环境、Visual Basic .NET 编程基础、程序控制结构、复合数据类型、过程、常用算法及其程序实现、面向对象的程序设计、窗体与控件、文件、ADO .NET 对数据库的访问、ASP.NET 技术等。读者在完成本书的学习和程序设计实践后,可以初步掌握 Visual Basic .NET 的应用程序开发技能。

本书可作为高等院校计算机基础课的教材,要求的起点不高,适合能进行简单计算机操作的初学者,也适合工程技术人员参考。

本书电子教案、在线和单机版的考试系统、在线辅导系统等资料可以从中国水利水电出版社网站免费下载,网址为 <http://www.waterpub.com.cn/softdown/>。

图书在版编目(CIP)数据

Visual Basic .NET 程序设计 / 潘晓文主编. —北京: 中国水利水电出版社, 2008

普通高等教育“十一五”国家级规划教材

ISBN 978-7-5084-5494-8

I. V… II. 潘… III. BASIC 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 (2008) 第 048761 号

书 名	Visual Basic .NET 程序设计
作 者	潘晓文 主 编 王李进 薛岚燕 赵艳萍 副主编 宁正元 主 审
出版 发行	中国水利水电出版社 (北京市三里河路 6 号 100044) 网址: www.waterpub.com.cn E-mail: mchannel@263.net (万水) sales@waterpub.com.cn 电话: (010) 63202266 (总机)、68331835 (营销中心)、82562819 (万水)
经 售	全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	北京市天竺颖华印刷厂
规 格	787mm×1092mm 16 开本 14.5 印张 350 千字
版 次	2008 年 4 月第 1 版 2008 年 4 月第 1 次印刷
印 数	0001—4000 册
定 价	26.00 元

凡购买我社图书,如有缺页、倒页、脱页的,本社营销中心负责调换

版权所有·侵权必究

前 言

Visual Basic .NET 是微软公司近年推出的一种核心编程语言,能快捷、方便地开发 Windows 应用程序和 Web 应用程序。本书以实例为中心,介绍使用 Visual Basic .NET 应用程序设计的方法和技术。

全书共分 11 章。第 1 章介绍 Visual Basic .NET 的入门知识,主要包括 .NET 框架、Visual Studio 集成开发环境及基本的 Windows 窗体控件,使读者能掌握基本的 Visual Basic .NET 程序设计步骤;第 2~5 章介绍 Visual Basic .NET 语言程序设计基础、程序流程控制,以及数组、结构与过程等基础知识;第 6 章介绍常用算法及其程序实现,通过该章的学习能提高学生的编程能力;第 7 章介绍面向对象的程序设计;第 8 章介绍窗体与控件;第 9 章介绍文件;第 10 章介绍 ADO .NET 对数据库的访问;第 11 章介绍 ASP.NET 技术。

本书各章节的内容是按照任务驱动教学法的要求进行组织的。任务驱动是建构主义理论中的一种教学模式,是将要学的新知识隐含在一个或多个任务之中,学生通过对所提出的任务进行分析、讨论,明确它大体涉及哪些知识,并找出哪些是旧知识,哪些是新知识,在老师的指导和帮助下找出解决问题的方法,最后通过任务的完成而实现对所学知识意义建构。建构既是对新知识意义的建构,同时又包含对原有经验的改造和重组,它强调以学生为中心,要求学生由被动的灌输对象转变为主动建构意义的帮助者、促进者。任务驱动并不是简单地给出任务就了事,重要的是要让学生学会学习。学生需要主动置身于学习情景中,通过自主发现问题、解决问题,从而构建知识经验,形成自己的见解,培养学生的创新能力与实践能力。本书的大多数章节都以一个综合性例子为主线。首先制定一个简单的任务,知识点随着功能的扩展而扩展,通过一个项目将整个章节所有的知识点串起来,通过带领学生完成一个项目来完成知识点的学习,改变了过去纯学习理论技能、纯做项目的做法。为了方便学生快速查阅书本中的知识点,本书仍然用知识点作为章节标题,而不用任务步骤作为章节标题。

本书提供的程序均已通过测试,另外还提供了教学幻灯片、在线和单机版的考试系统、在线辅导系统等资料,读者可到中国水利水电出版社网站上免费下载。

本书第 1、2、7、9、11 章由潘晓文编写,第 3、4、8 章由薛岚燕编写,第 5、6 章由赵艳萍编写,第 10 章由王李进编写,宁正元教授审阅了全书,并提出许多宝贵意见,在此表示感谢。另外参加部分章节编写工作的还有林灵芝、吴福生、李世阳、刘珠明、刘桂英、张素娟、魏志煌、郑志翔、林上富、陈元洪等。由于作者水平有限,书中错误和不当之处在所难免,恳请广大读者批评指正。

编 者

2008 年 1 月

目 录

前言

第1章 Visual Basic .NET 入门.....	1
本章学习目标	1
1.1 程序与程序语言	1
1.2 面向对象的基本概念	2
1.3 .NET 框架 (.NET Framework) 概述.....	3
1.3.1 公共语言运行库	4
1.3.2 .NET 框架基础类库.....	4
1.4 Visual Studio .NET	5
1.4.1 新建一个项目	5
1.4.2 Visual Basic .NET 集成开发环境.....	6
1.4.3 控件绘制	8
1.4.4 调整大小、移动和锁定控件.....	9
1.4.5 设置窗体和控件的属性	9
1.4.6 编写事件过程	11
1.4.7 其他浮动面板	14
第2章 Visual Basic .NET 编程基础	17
本章学习目标	17
2.1 Visual Basic .NET 数据类型.....	17
2.2 变量和常量	19
2.2.1 变量的命名规则	20
2.2.2 变量的声明	21
2.2.3 变量的作用域与生存期	22
2.2.4 静态变量	22
2.2.5 常量	22
2.2.6 数据类型的转换	24
2.3 运算符与表达式	24
2.3.1 赋值运算符	24
2.3.2 算术运算符	25
2.3.3 逻辑/按位运算符	26
2.3.4 移位运算符	28
2.3.5 比较运算符	30
2.3.6 串联运算符	31
2.3.7 运算符优先级	31

2.4 常用的方法	32
2.4.1 Math 类	32
2.4.2 Conversion 模块	34
2.4.3 MessageBox 类	34
第 3 章 程序控制结构	37
本章学习目标	37
3.1 顺序结构程序设计	37
3.2 选择结构程序设计	38
3.2.1 If 语句	41
3.2.2 Select Case 语句	45
3.2.3 On-GoTo 语句	47
3.3 循环结构程序设计	48
3.3.1 For...Next 语句	49
3.3.2 While...End while 循环	51
3.3.3 Do...loop 循环	52
3.3.4 循环的嵌套	55
3.4 其他控制语句	56
3.4.1 GoTo 语句	56
3.4.2 Exit 语句	56
3.4.3 End 语句	57
第 4 章 复合数据类型	58
本章学习目标	58
4.1 结构	58
4.1.1 声明结构	59
4.1.2 结构变量	59
4.2 数组	60
4.2.1 数组的定义	60
4.2.2 数组的基本操作	62
4.2.3 结构数组	66
4.2.4 Array 对象	67
4.3 枚举	69
4.3.1 用 Enum 语句定义枚举	69
4.3.2 枚举变量	70
4.3.3 枚举的应用	71
4.4 集合	71
4.4.1 Collection 对象	71
4.4.2 自定义集合对象	73
第 5 章 过程	74
本章学习目标	74

5.1 Sub 过程	74
5.1.1 通用过程的定义	74
5.1.2 通用过程的创建	75
5.1.3 Sub 子过程的调用	76
5.2 Function 过程	77
5.2.1 Function 过程的定义	77
5.2.2 Function 过程的调用	78
5.2.3 模块	80
5.3 参数的传递	81
5.3.1 形参与实参	82
5.3.2 数组参数	85
5.3.3 可选参数	88
5.3.4 可变参数	89
5.4 过程的嵌套和递归调用	90
5.4.1 过程的嵌套	90
5.4.2 递归调用	91
5.5 过程重载	92
第6章 常用算法及其程序实现	94
本章学习目标	94
6.1 算法和数据结构	94
6.1.1 算法的概念	94
6.1.2 算法的描述	95
6.1.3 数据结构的概念	95
6.2 查找和排序算法	97
6.2.1 查找算法及其程序实现	97
6.2.2 排序算法及其程序实现	99
6.2.3 数组查找和排序算法程序举例	102
6.3 基本数值算法	104
6.3.1 一元非线性方程实根的数值算法及其程序实现	104
6.3.2 一元函数定积分的数值算法及其程序实现	106
第7章 面向对象的程序设计	108
本章学习目标	108
7.1 面向对象的相关概念	108
7.1.1 面向对象程序设计的基本特性	108
7.1.2 字段、属性过程、方法和事件	109
7.2 定义类	109
7.2.1 用面向对象的思想对问题进行描述	109
7.2.2 定义类	110
7.3 对象	113

7.3.1	创建和使用对象	113
7.3.2	创建和使用对象数组	114
7.4	类的继承	117
7.4.1	Inherits 语句	118
7.4.2	继承修饰符	119
7.4.3	添加子类的属性和方法	119
7.4.4	重写派生类中的属性和方法	119
7.4.5	重载和重写	120
7.4.6	继承和作用域	120
7.4.7	New 和 Finalize 方法在类层次结构中如何工作	120
7.4.8	面向对象程序设计实例	121
7.5	Visual Basic 中的共享成员	122
7.6	事件	122
7.7	接口	125
第 8 章	窗体	127
	本章学习目标	127
8.1	Visual Basic 的窗体	127
8.2	窗体外观设计	129
8.3	窗体的事件	132
8.3.1	鼠标事件	132
8.3.2	键盘事件	133
8.3.3	窗体启动和关闭事件	133
8.4	在窗体上加入控件	134
8.4.1	标签	134
8.4.2	文本框	135
8.4.3	单选按钮	136
8.4.4	按钮	136
8.4.5	调整控件位置	137
8.4.6	分组框	138
8.4.7	组合框	139
8.4.8	复选框控件	140
8.4.9	图片框	140
8.4.10	列表框	141
8.4.11	菜单	142
8.4.12	计时器	145
8.4.13	滚动条	145
8.4.14	打印	146
8.5	多文档界面应用程序	147

第9章 文件	149
本章学习目标	149
9.1 文件和流的概念	149
9.1.1 文件的概念	149
9.1.2 流的概念	149
9.2 文本文件的读写	150
9.2.1 StreamReader 对象	150
9.2.2 StreamWriter 对象	151
9.3 二进制文件的读写	152
9.3.1 FileStream 对象	153
9.3.2 BinaryWriter 和 BinaryReader 对象	155
9.3.3 复杂数据类型的数据写入	156
9.4 目录和文件操作	157
9.4.1 目录操作	158
9.4.2 文件操作	159
第10章 ADO.NET 对数据库的访问	160
本章学习目标	160
10.1 数据库基础	160
10.1.1 数据库基本概念	160
10.1.2 用 Access 实现数据库	161
10.2 结构化查询语言 SQL	166
10.2.1 SELECT 语句	166
10.2.2 UPDATE 语句	167
10.2.3 INSERT 语句	168
10.2.4 DELETE 语句	168
10.3 ADO.NET 体系结构	168
10.4 Connection 对象	170
10.4.1 SqlConnection 对象	171
10.4.2 OleDbConnection 对象	172
10.5 Command 对象与 DataReader 对象	173
10.5.1 Command 对象	173
10.5.2 DataReader 对象	173
10.5.3 使用 DataReader 对象读取数据库	174
10.6 DataSet 对象与 DataAdapter 对象	179
10.6.1 DataSet 对象	179
10.6.2 DataAdapter 对象	179
10.7 DataView 对象	182
10.8 在 DataGrid 控件中显示数据	184
10.9 使用 DataSet 更新数据库	185

第 11 章 ASP.NET 初步	189
本章学习目标	189
11.1 Web 编程基础	189
11.1.1 网页浏览基本原理	189
11.1.2 网页地址	190
11.1.3 HTML 文件格式	190
11.1.4 静态网页和动态网页	192
11.2 Web 服务器与 Microsoft IIS	193
11.2.1 IIS 的安装	193
11.2.2 配置 IIS	193
11.3 ASP.NET 的工作原理	194
11.3.1 ASP.NET 简介	194
11.3.2 新建 ASP.NET 应用程序	195
11.3.3 ASP.NET 网页代码模型	198
11.3.4 Web 窗体运行机制	200
11.4 服务器端控件	200
11.4.1 控件的声明及属性、事件和方法	200
11.4.2 HTML Server 控件	201
11.4.3 Web 服务器控件	203
11.4.4 Web 窗体程序设计与 Windows 窗体程序设计的不同之处	209
11.5 数据验证控件	210
11.5.1 RequiredFieldValidator 控件	211
11.5.2 RangeValidator 控件	211
11.5.3 RegularExpressionValidator 控件	211
11.5.4 CompareValidator 控件	212
11.5.5 ValidationSummary 控件	212
11.6 内部对象	213
11.6.1 Request 对象	214
11.6.2 Response 对象	214
11.6.3 Session 对象	214
11.6.4 Application 对象	215
11.6.5 Cookie 对象	216
参考文献	219

第 1 章 Visual Basic .NET 入门

本章学习目标

本章作为学习程序设计的入门章节，首先介绍了程序以及程序设计语言的概念，以便读者初步了解程序设计的内容，接着介绍了各种程序设计语言的特点。本章通过一个简单的例子描述了 Visual Basic .NET 程序的开发步骤，并对 Visual Basic .NET 集成开发环境做了总体介绍。由于 Visual Basic .NET 是基于 Microsoft .NET 平台的开发工具，所以本章也对 .NET Framework 做了简单介绍。

1.1 程序与程序语言

计算机程序是一系列的计算机指令，命令计算机去完成某一项工作，从而实现一定的结果。编写这些计算机指令的工作就称为编程，这些指令的集合就是程序，所用指令的描述方式或描述规则就是程序语言。

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。机器语言是 CPU 的指令系统，用机器语言编写的程序可以被计算机直接理解和执行。但是它们不直观、难记、难理解、不易查错、开发周期长，要求用户必须具有较多的计算机方面的知识，熟悉机器的内部结构，并且不同指令集的 CPU 机器语言也不尽相同，使得程序的可移植性差。为了减轻程序员的负担，人们使用一些用于帮助记忆的符号来代替机器语言的指令，由这些助记符组成的指令系统称为汇编语言。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，可读性仍然不是很好，且不太容易编写非常大的程序。

为了克服低级语言的弱点，从 20 世纪 50 年代起，相继研制出了 FORTRAN、C、Pascal、BASIC 等高级语言。高级语言是一种类似自然语言和数学公式的程序设计语言，它们独立于计算机，所以程序的可移植性好。另外，高级语言的语句功能强，通常一个语句相当于许多条机器指令，使编程效率大为提高。使用高级语言，程序员可以不关心机器的内部结构，把主要精力放在解决问题的思路和方法上，大大减轻了程序员的负担。使用这一时期的高级语言，程序员的注意力集中在算法（用计算机在有限步骤内解决问题的方法）的描述上。程序员不仅要告诉计算机“做什么”，而且还要告诉计算机“如何做”。用这类编程语言编程，通常把解题的过程看成是数据加工的过程。正是由于用这些语言来解决问题时需要详细地描述解题过程，因此这类语言也可称为“面向实现的语言”或“面向过程的语言”。

为了更好地描述与解决现实中的问题，人们又形成了一种面向对象的程序设计思想。面向对象的程序设计（OOP，Object-Oriented Programming）方法和编程技术不同于面向过程的程

程序设计的一点是：它把要处理的数据以及处理这些数据的算法“封装”在一起，用于抽象地描述现实世界的对象，所以这类语言又可称为“面向现实的”语言。以面向对象的观点来观察世界，世界上所有的事物都可以看成是对象，每个对象都有其特征，具有相同特征的对象同属一个类。对象的特征分为静态特征（属性）和动态特征（方法）。对象的属性用具有一定数据结构的数据来表示，对象的方法用施加于这些数据之上的操作来表示。

面向对象的程序设计方法与编程技术另一个不同于面向过程的程序设计的地方是：程序设计人员在进行面向对象的程序设计时，不再是为了描述解题过程，而是考虑如何利用类来创建对象，然后设置对象的属性，调用对象的方法，利用对象来简化程序设计。对象的创建，既可以使用外部提供的类，也可以是用户自己定义的类，甚至可以以一个现成的类为基础，通过“继承”来获得一个满足用户需求的类。提高代码的可重用性，有利于编程效率的提高。这样对问题的解决就变成了对问题的描述，即设置对象的属性和调用对象的某个方法。这种思维方法与人们对现实世界的认识相一致，比较容易被人们理解和掌握。

1.2 面向对象的基本概念

面向对象的思想就是把世界看成是由对象组成的。对象可以是任何概念上相对独立的事物，比如一个人、一辆汽车、一个杯子、一张桌子都可以被看成是对象，甚至一件事情、一段程序也都可以被看成是对象。

类是对象的模板，是对象的归纳和总结。即把同属一个类的对象所具有的共同特征（属性和方法）的描述提取出来，作为一个概念来研究。同属一个类的对象拥有相同的属性和方法，所不同的只是各个对象的属性值是不一样的，因此属性和方法是区分类的特征，属性值则可以区分同一个类的不同对象。比如，杯子类的对象的属性有形状、手柄、盖子、材质等，杯子的方法有：盛水；而汽车类的对象的属性有：轮胎、发动机、车身等，方法有启动、前进、倒车、停车等。从这个例子可以看出，杯子和汽车这两类有不同特征（属性和方法），以此可以区别不同的类。同属于一个类的对象有相同的属性和方法，它们之间的区别是属性值不同。以卡车和轿车为例，它们都属于汽车类。但是，卡车的轮胎、发动机、车身等属性对应的属性值为：直径大、发动机功率大、带车斗；轿车对应的属性为：直径小、发动机功率小、不带车斗。因此，卡车和轿车是同一个类的不同对象，但是它们都有共同的方法：启动、前进、倒车、停车等。

应注意的是，在面向对象的程序设计中，类和对象的获得与现实世界有一些区别。在现实世界中，是先有具体的对象后有概念上的类。以人类为例，人类这个概念是通过对所有具体的人（即一个个对象）进行考察，然后用归纳和总结的方法获得的。而在面向对象的程序设计中，程序员是先根据程序设计的需要定义出一个类，然后再以这个类为模板“制造”出一个个对象（每个对象属性值不同），即先有概念上的类后有具体的对象。面向对象的程序设计就是模拟人对世界的认识，先用事先预制好的类定义各个对象，再通过调用对象的方法实现自己的目的。

Visual Basic .NET 是基于 .NET 平台的面向对象程序设计语言。.NET 框架提供了庞大的类库供程序设计人员使用。

1.3 .NET 框架（.NET Framework）概述

.NET 的本质涵盖了一整群新技术，这些技术包括 Web Services、.NET Framework、Visual Studio .NET、.NET My Services、.NET Enterprise Servers 等。但是 .NET 中最重要的新技术首推 Web Services，但对 Windows 开发人员来说，最重要的是 .NET Framework。

.NET 框架是微软于 2002 年正式推出的一个程序开发平台，是创建、部署和运行 Web 服务及其他应用程序的一个环境。.NET 框架有两个主要组件：公共语言运行库和 .NET Framework 类库。它不仅是 Visual Basic .NET 程序的开发平台，同时也是 Visual Basic .NET 程序的运行平台。同样地，用 Visual C++ .NET、C# 等语言编写的程序也依赖于 .NET 框架。由于 .NET 框架是 Visual Basic .NET 的基础，是 Visual Basic .NET 较以前版本 Visual Basic 先进的根源，因此在了解 Visual Basic .NET 之前，有必要了解一下 .NET 框架。

1. .NET 框架的优势

(1) 跨语言。.NET 支持多种语言的互操作，程序员可以用他们选择的开发语言编写应用程序，却仍能充分利用其他语言编写的运行库、类库和组件。

(2) 跨平台。.NET 通过将各种语言先编译成中间语言，编译器再将其编译成本地平台代码来实现异构平台下对象的互操作。

(3) 安全。.NET 通过公共语言运行库来实现资源对象和类型的安全。

(4) 对开放互联网标准和协议的支持。.NET 通过对 HTTP、XML、SOAP、WSDL 等 Internet 标准的强劲支持，提供在异构网络环境下获取远程服务、连接远程设备、交互远程应用的编程界面。

2. .NET 框架的构成

.NET 框架是由 CLR（Common Language Runtime，公共语言运行库）和 .NET 框架类库构成的。应用程序建立在这两个层次之上。图 1-1 刻画了 .NET 框架的整体结构。

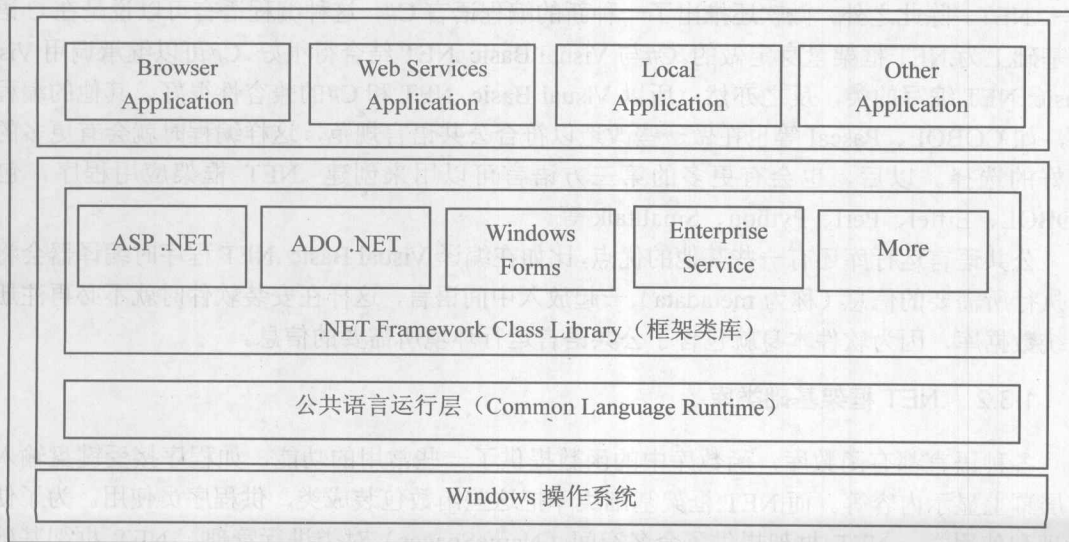


图 1-1 .NET 的框架

1.3.1 公共语言运行库

公共语言运行库 (CLR, Common Language Runtime) 是 .NET Framework 中一切东西的基础。每一个 .NET Framework 应用程序都要使用它, 不论这个应用程序以什么语言编写。它同时也是 .NET Framework 类库的基础, 是理解 .NET 框架的起点。

公共语言运行库主要是为了解决各种不同编程语言之间相互调用同一个程序的问题。例如, 同一个项目下的程序设计人员由于各自掌握不同的语言, 他们之间编写的程序要相互调用。另外, 每种编程语言都有自己的优点和缺点, 程序设计人员往往希望能同时使用几种语言来编程, 以弥补各种语言的缺陷。各种编程语言的表达方式基本一样, 即这些语言的语义基本相同, 但语法却不一样, 比如, C# 语言用 “;” 来表示语句结束, 而 Visual Basic .NET 不允许使用它。

因此, 有必要将这些语义定义一份标准实现, 并允许各种语言以不同的语法来表达这些语义。CLR 提供了上述的标准实现。通过一套通用的数据类型, 以及继承机制的规范, CLR 为建造于其上的语言定义出一套通用语义。至于语法, CLR 并无任何限制。一门具体语言的语法完全由那门语言的设计者自行定夺。

具体的办法是, 首先要求程序语言符合公共语言规范 (CLS, Common Language Specification)。在程序调试成功后, 因为程序语言符合公共语言规范, 所以可以把它编译成与具体语言规则无关的中间语言 (IL, Intermediate Language)。运行程序时再由相应计算机上的公共语言运行环境把 IL 编译成当前计算机能识别的二进制语言。在 Windows 环境下, 用支持 .NET 的语言所编写的程序被编译成 MSIL (Microsoft Intermediate Language, 微软中介语言), 而非特定机器的相关二进制码。MSIL 是一套与 CPU 无关的指令集, 用来执行诸如装载、存储信息、调用方法之类的典型动作。在系统上安装一个 .NET Framework 应用程序, 意味着将 “内含 MSIL 而非特定机器的相关二进制码” 的磁盘文件拷贝上去。

要使程序语言符合公共语言规范也就要求每种语言都要做或多或少的修改。将 Visual Basic 6.0 做改造的结果就是崭新的 Visual Basic .NET, Visual C++ 6.0 也被改造为 Visual C++ .NET。除此之外, 微软还推出了一种新的编程语言 C#, 这种编程语言可以说是在 C 语言的基础上为 .NET 框架量身定做的。C# 与 Visual Basic .NET 结合得很好, C# 可以继承调用 Visual Basic .NET 编写的类, 反之亦然, 所以 Visual Basic .NET 和 C# 的兼容性很好。其他的编程语言, 如 COBOL、Pascal 等也在做一些改造以符合公共语言规范, 这样编程时就会有更多的、更好的选择。以后, 也会有更多的第三方语言可以用来创建 .NET 框架应用程序, 包括 COBOL、Eiffel、Perl、Python、Smalltalk 等。

公共语言运行库还有一些其他的优点, 比如在编译 Visual Basic .NET 程序时编译器会将程序执行所需要的信息 (称为 metadata) 一起放入中间语言, 这样在安装软件时就不必再注册到系统数据库, 因为软件本身就包含了公共语言运行环境所需要的信息。

1.3.2 .NET 框架基础类库

各种语言都有函数库, 函数库中的函数提供了一些常用的功能, 如程序接受键盘输入、在屏幕上显示内容等。而 .NET 框架基础类库把这些函数包装成类, 供程序员使用。为了便于管理和使用类, .NET 框架提供了命名空间 (NameSpaces) 对类进行管理。 .NET 框架基础类库中的类分别隶属于不同的命名空间。

.NET 框架类库被组织为一套具有层次结构的命名空间，所有的类按相关功能的逻辑类别进行分组。每个命名空间可以包含值类型 (Types)、类 (Classes)、界面 (Interfaces) 以及其他次级命名空间 (Sub-Namespace)。System 命名空间是 .NET 框架类库的源头，命名空间与 Windows 文件夹类似。命名空间使开发人员更容易在代码中浏览和引用类，命名空间的层次结构用 “.” 来表示。比如要制作一个按钮，就可以在 System.Windows.Forms 命名空间中找到 button 这个类，通过这个类生成一个按钮对象。

.NET 框架类库非常庞大，一个人不可能把 .NET Framework 中所有的命名空间内容完全掌握，每个开发人员只要掌握自己感兴趣的一部分内容，其他的只要做一般性的了解即可。

1.4 Visual Studio .NET

Visual Studio .NET 是一套完整的开发工具，用于生成 Web 应用程序、XML Web Services、桌面应用程序和移动应用程序。Visual Basic .NET、Visual C++ .NET、Visual C# .NET 和 Visual J# .NET 全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。

本节将通过一个简单的例子介绍如何在 Visual Studio .NET 下开发一个 Visual Basic .NET 的 Windows 应用程序，并认识一下集成开发环境 (IDE)，以便于后续章节的学习。

1.4.1 新建一个项目

假设已经安装完成了 Visual Studio .NET (安装过程并不复杂，只要按照提示一步步操作即可)。要打开 Visual Studio .NET，请单击“开始”→“程序”→Microsoft Visual Studio .NET 2003→Microsoft Visual Studio .NET 2003 命令，.NET 开发环境起始页的窗口如图 1-2 所示。

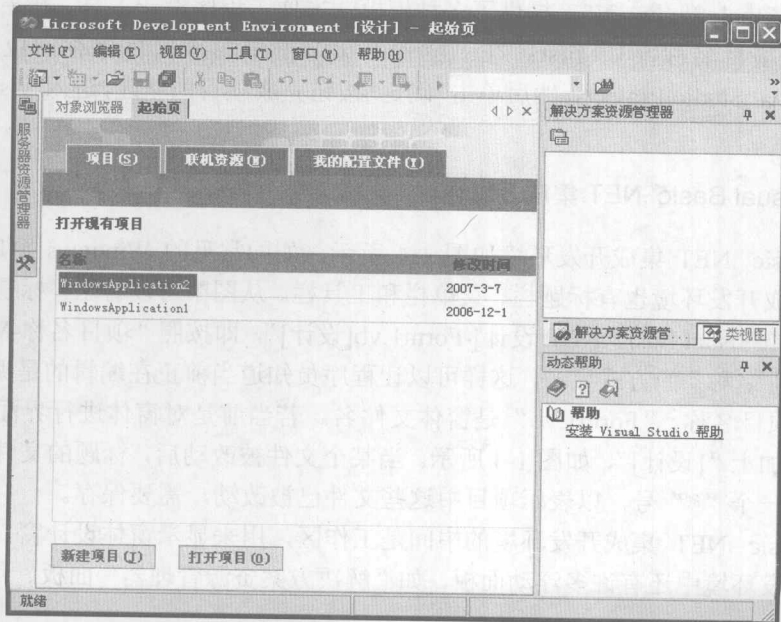


图 1-2 .NET 开发环境起始页的窗口

项目是一个程序所需相关文件的集合, 通常包括窗体文件、图标文件等其他相关文件。要编写一个新程序就要新建一个项目, 单击起始页窗口中的“新建项目”按钮; 要打开一个程序则单击“打开项目”按钮。

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“文件”→“新建”→“项目”命令, 打开如图 1-3 所示的“新建项目”对话框。

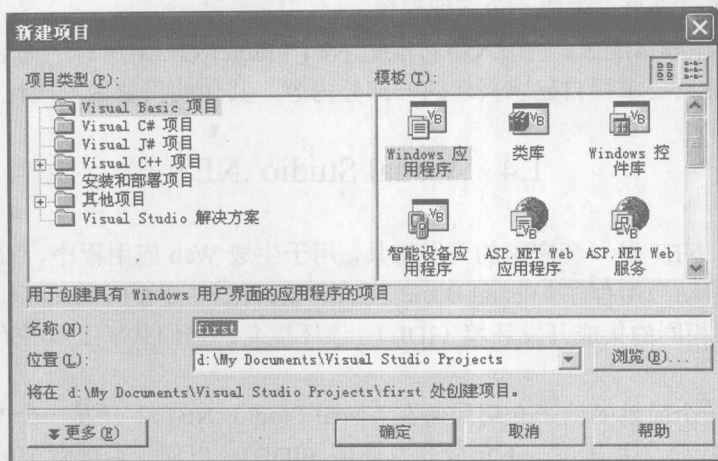


图 1-3 “新建项目”对话框

要建立一个基于 Visual Basic .NET 的 Windows 应用程序, 首先在对话框左侧的“项目类型”窗格中选择“Visual Basic 项目”, 然后在右侧的模板窗格中选择“Windows 应用程序”模板。模板提供了程序所需的基本代码和一个窗体。有了模板, 编写程序就像在做填空题, 只要在规定的地方填上必要的代码即可完成程序设计。如果选择“空项目”模板, 则需要自己编写初始化代码。Visual Studio .NET 提供了多种模板, 方便了程序设计人员。接着, 在“名称”文本框中填入项目名称, 如 first; 在“位置”组合框中填入项目保存的位置, 如 d:\My Documents\Visual Studio Projects。然后单击“确定”按钮完成项目的新建, 进入 Visual Basic .NET 集成开发环境。

1.4.2 Visual Basic .NET 集成开发环境

Visual Basic .NET 集成开发环境如图 1-4 所示, 如同常见的 Windows 窗口一样, Visual Basic .NET 集成开发环境也有标题栏、菜单栏和工具栏。从图中可以看出, 标题栏上显示的是“first-Microsoft Visual Basic .NET[设计]-Form1.vb[设计]”, 即按照“项目名称-Microsoft Visual Basic .NET-文件名称”的方式显示, 这样可以让程序员知道当前正在编辑的是哪个文件。本例中“first”是项目名称, “Form1.vb”是窗体文件名。若当前是对窗体进行界面设计, 则标题在文件名后面加上“[设计]”, 如图 1-4 所示。当某个文件被改动后, 标题的文件名以及项目名称后面将加上一个“*”号, 以表示项目中这些文件已被改动, 需要保存。

Visual Basic .NET 集成开发环境的中间是工作区, 用来显示窗体设计窗口和代码编辑窗口。在集成开发环境中还有许多浮动面板, 如“解决方案资源管理器”面板、属性窗口、工具栏等。

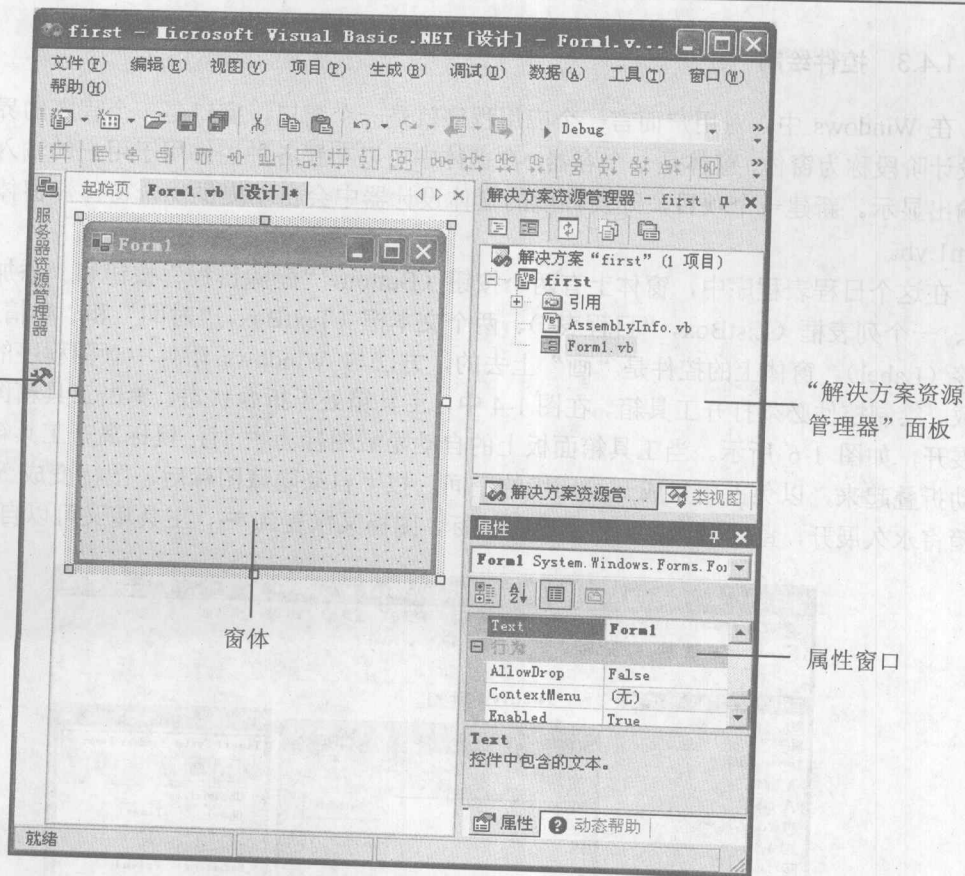


图 1-4 Visual Basic .NET 集成开发环境

本节通过完成 first 项目来介绍整个 .NET 集成开发环境以及 Visual Basic .NET 程序设计的步骤。该项目是一个日程安排程序，用户输入日程信息（时间和将要处理的事情），如 15:30 到会议室开会等。当时间到时，程序将弹出消息框提醒用户，程序的界面如图 1-5 所示。用户在“时间”文本框中以“时:分”的形式输入时间，然后在“事情”文本框中输入将要完成的事情，单击“添加日程”按钮，日程（时间+事情）即被添加到日程表中。程序将每分钟检查一次当前时间，判断当前时间是否与某个日程的时间一致，若一致，则弹出消息框显示该日程。若要将日程表中的某个日程删除，只要在日程表中选中该日程并单击“删除日程”按钮即可。

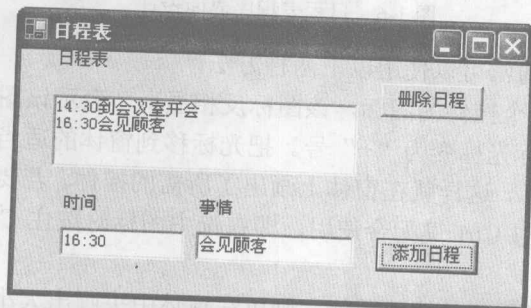


图 1-5 日程安排表程序