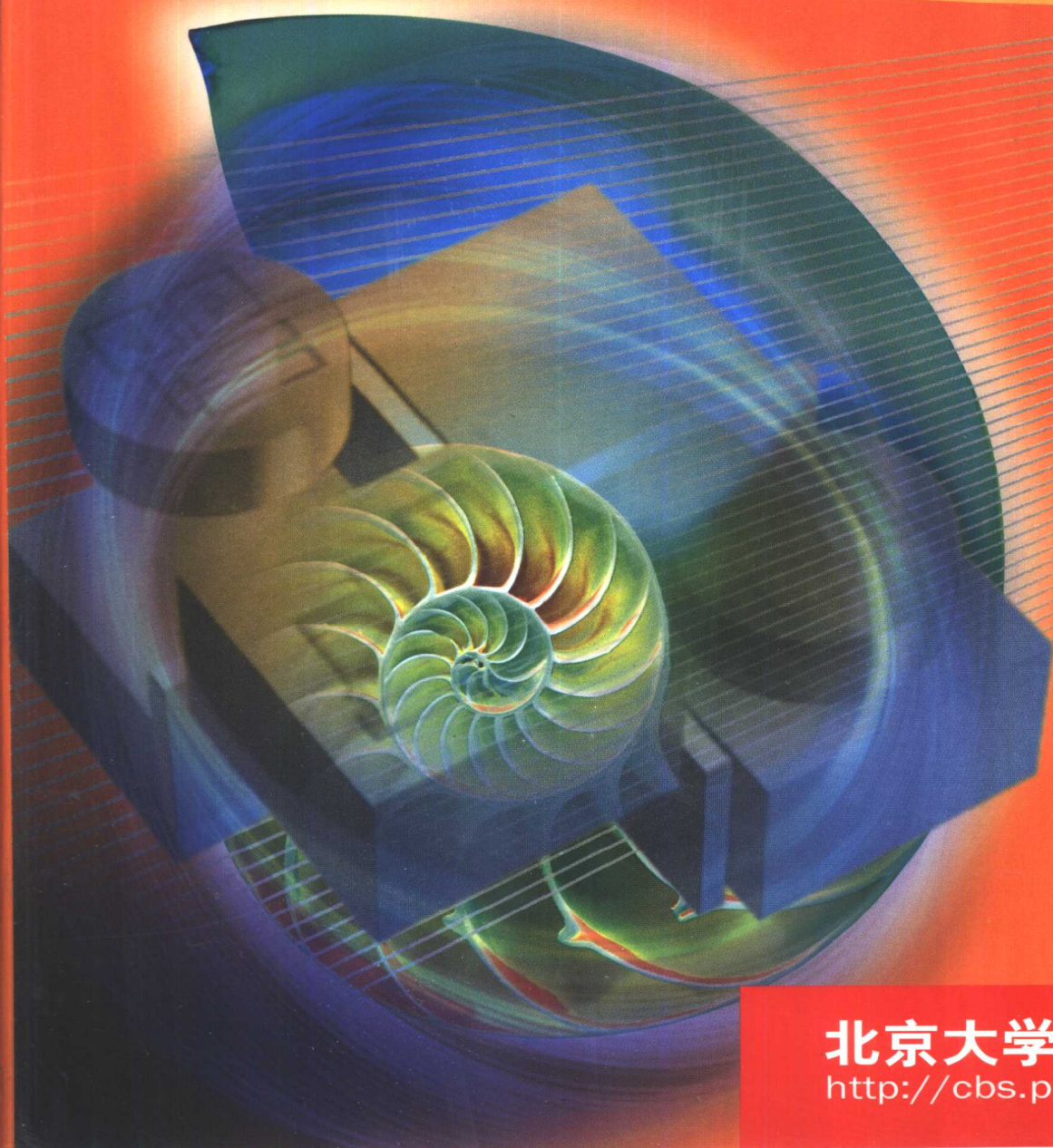


Visual Basic .NET 学习教程

康晓明 姚小磊 编著



北京大学出版社
<http://cbs.pku.edu.cn>

Visual Basic.NET 学习教程

康晓明 姚小磊 编著

北 京 大 学 出 版 社

北 京

内 容 简 介

Visual Basic.NET (又称 Visual Basic 7.0 或 VB 7.0) 是 Microsoft 公司在 Visual Basic 6.0 之后推出的最新版本, 集成在 Visual Studio.NET 7.0 中, 与 Visual C++.NET 以及 C# 组成了 “.NET” 构架。Visual Basic 是 Windows 环境下简单、易学、高效的一种编程语言, 其快速开发的特性深受程序员的喜爱, 但是 .NET 以前版本的 Visual Basic 面向对象的能力远远不能满足程序员的需要, 这也是项目越大越少用到 Visual Basic 的原因。Visual Basic.NET 增加了许多新的面向对象的特性, 比如继承、重载等。语言的新特点也包括了对进程的控制和低层结构的操作, 这些新特性使得 Visual Basic.NET 再次成为程序员关注的焦点。

本书主要内容如下: Visual Basic.NET 入门、工程管理、Visual Basic.NET 语言体系结构、Visual Basic.NET 的常用控件、应用程序界面、Visual Basic.NET 的 OOP 结构、数据库访问技术、Visual Basic.NET 的多线程、Visual Basic.NET 的数组、文件和出错处理、从 Visual Basic 6.0 到 Visual Basic.NET。

本书不仅是一本初学者可以用来入门的教程, 而且对于专业编程人员来说, 也是一本内容翔实、可供借鉴的参考书。

图书在版编目 (CIP) 数据

Visual Basic.NET 学习教程/康晓明, 姚小磊编著. —北京: 北京大学出版社, 2001.11

ISBN 7-301-05303-7

I. V… II. ①康…②姚… III. BASIC 语言—程序设计—教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2001) 第 076919 号

书 名: Visual Basic.NET 学习教程

著作责任者: 康晓明 姚小磊

责任编辑: 段志刚 黄庆生

标准书号: ISBN 7-301-05303-7/TP·0624

出 版 者: 北京大学出版社

地 址: 北京市海淀区中关村北京大学校内 100871

网 址: <http://cbs.pku.edu.cn>

电子信箱: xxjs@pup.pku.edu.cn

印 刷 者: 河北省滦县印刷厂

发 行 者: 北京大学出版社

经 销 者: 新华书店

787 毫米×1092 毫米 16 开本 17.125 印张 438 千字

2001 年 11 月第 1 版 2001 年 11 月第 1 次印刷

定 价: 26.00 元

NJ505/06

前 言

几年前，Visual Basic 还是一个小语言，一本“从入门到精通”可以包括它的方方面面，但是从第一版到现在，Visual Basic 经历了巨大的演变，它已成为一个重要的开发环境，包括编程的各个方面，从教育应用程序到数据库编程，从财务应用程序到 Internet 构件。编写一本简介所有这些课题的书籍已经不可思议，更不用说“精通”了。

Visual Basic.NET（又称 Visual Basic 7.0 或 VB 7.0）是 Microsoft 公司在 Visual Basic 6.0 之后推出的最新版本，集成在 Visual Studio.net 7.0 中，与 Visual C++.NET 以及 C# 组成了“.NET”构架。Visual Basic 是 Windows 环境下简单、易学、高效的一种编程语言，其快速开发的特性深受程序员的喜爱，但是.net 以前版本的 Visual Basic 面向对象的能力远远不能满足程序员的需要，这也是越大项目越少用到 Visual Basic 的原因。Visual Basic.NET 增加了许多新的面向对象的特性比如继承、重载等。语言的新特点也包括了对进程的控制和低层结构的操作，这些新特性使得 Visual Basic.NET 再次成为程序员关注的焦点。

本书的主要内容如下：Visual Basic.NET 入门、工程管理、Visual Basic.NET 语言体系结构、Visual Basic.NET 的常用控件、应用程序界面、Visual Basic.NET 的 OOP 结构、数据库访问技术、Visual Basic.NET 的多线程、Visual Basic.NET 的数组、文件和出错处理、Visual Basic 6.0 到 Visual Basic.NET。

本书旨在让大多数平均水平的编程人员充分地利用 Visual Basic，其中包括对大多数 Visual Basic 编程人员特别有用的课题。Visual Basic 是个相当理想的编程环境，我们必须在选择大量课题和深入介绍几个课题之间进行权衡取舍。为了介绍更多的课题，我们割舍了大量的参考资料和长长的程序清单。

许多书中提供了长长的编号，说明详细的操作步骤，这的确可以简化工作，但是编程决不可循规蹈矩，按部就班。编程是一种创造性的劳动，一种技术应当能在多种不同的情况下随机应变地使用。要创造性地介绍像 Visual Basic 这样的语言能力，必须介绍其原理和编程模型，培养一种将这个语言用在日常编程情景中的感觉。

本书的章节主要是涉及一些 Visual Basic.NET 的基本功能的实现用法，通过一些例子可以对 Visual Basic 的基本操作有一定的认识，是那些想学习 Visual Basic 的入门者和已经对 Visual Basic 的以前的版本有一定的了解而想学习 Visual Basic 7.0 的先前的 Visual Basic 编程人员的入门书籍。

本书由康晓明、姚小磊主编，另外，韩永鹏、魏宝晶、朱明峰、袁怡翔、肖依永、邱亚希、苗泽纯、孟鸽、喻湘宁、张晟、盖宇、黄涛、朱凯、刘梓超、翟丽君、裴浴、叶萱、白致铭、黎君雪、谢宏威、杨继运等也参加了本书的编写工作。

由于编者水平有限，本书难免有不足之处，欢迎广大读者批评指正。

编 者

2001 年 11 月

目 录

第1章 Visual Basic.NET 入门	1
1.1 Visual Basic.NET 概述	1
1.2 Visual Basic.NET 的新特点	1
1.3 Visual Basic.NET 的集成开发环境	4
1.4 Visual Basic.NET 的第一个应用程序	6
1.4.1 创建新的 Visual Basic.NET 工程	6
1.4.2 创建应用程序的用户界面	7
1.4.3 设置用户界面中各对象的属性	8
1.4.4 编写程序代码	9
1.4.5 保存和运行程序	10
1.4.6 创建可执行文件	10
1.4.7 小结	10
第2章 工程管理	11
2.1 工程概述	11
2.2 工程文件的操作	13
2.2.1 创建、打开及保存工程	13
2.2.2 在工程中添加、删除和保存文件	13
2.3 定制工程特性	14
2.3.1 定制工程的属性	14
2.3.2 在工程中添加引用	16
第3章 Visual Basic.NET 语言体系结构	18
3.1 Visual Basic.NET 的数据类型	18
3.1.1 Numeric 数据类型	18
3.1.2 Byte 数据类型	19
3.1.3 String 数据类型	19
3.1.4 Boolean 数据类型	20
3.1.5 Date 数据类型	21
3.1.6 Object 数据类型	21
3.1.7 用户自定义类型	21
3.1.8 数组	22
3.2 Visual Basic.NET 的运算	25

3.2.1 算术运算	26
3.2.2 赋值运算符	26
3.2.3 二进制运算符	27
3.2.4 比较运算符	27
3.2.5 连接运算符	28
3.2.6 逻辑运算符	28
3.2.7 运算符的优先级	29
3.3 Visual Basic.NET 的常量和变量	29
3.3.1 常量	29
3.3.2 变量	31
3.4 Visual Basic.NET 的流程和控制结构	34
3.4.1 条件分支结构	34
3.4.2 循环结构	36
3.5 Visual Basic.NET 的过程和函数	39
3.5.1 Sub 过程	39
3.5.2 Function 过程	40
3.5.3 调用过程	41
3.5.4 向过程传递参数	42
3.6 Visual Basic.NET 的类和对象基础	44
第4章 Visual Basic.NET 的常用控件	47
4.1 Label 控件	47
4.2 Button 控件	48
4.3 TextBox 控件	49
4.4 MainMenu 控件	51
4.5 CheckBox 控件	52
4.6 RadioButton 控件	53
4.7 GroupBox 控件	54
4.8 PictureBox 控件	55
4.9 ListBox 控件	56
4.10 CheckedListBox 控件	60
4.11 ComboBox 控件	61
4.12 ListView 控件	64
4.13 TreeView 控件	67
4.14 ImageList 控件	68
4.15 Timer 控件	69
4.16 HScrollBar、VScrollBar 控件	70
4.17 ProgressBar 控件	71
4.18 ToolBar 控件	72
4.19 StatusBar 控件	73

第 5 章 应用程序界面	75
5.1 界面样式	75
5.2 多文档界面应用程序	77
5.2.1 创建 MDI 应用程序	78
5.2.2 MDI NotePad 应用程序	79
5.2.3 MDI 窗体的进一步研究	80
5.3 关于窗体的其他描述	81
5.4 对话框	83
5.4.1 模式与无模式的对话框	83
5.4.2 预定义对话框的使用	84
5.4.3 用窗体作为自定义对话框	85
5.4.4 各种显示类型的设计	86
5.5 界面设计的基本原则	87
第 6 章 Visual Basic.NET 的 OOP 结构	94
6.1 OOP 的相关概念	94
6.2 VB.NET 的面向对象性	96
6.2.1 VB.NET 的共享成员 (Share Members)	99
6.2.2 类模块和标准代码模块的区别和比较	100
6.2.3 对象浏览器	101
6.3 建立和使用对象	102
6.3.1 对象的生命周期: 对象的建立和销毁	102
6.3.2 设置和重设属性	103
6.3.3 用方法来表现动作	104
6.3.4 对象变量的声明	105
6.3.5 对一个对象进行多种操作	106
6.3.6 使用 New 关键字	108
6.3.7 释放对对象的引用	109
6.3.8 把对象传递到一个过程	110
6.4 在程序运行中得到一个类的信息	111
6.4.1 发现对象属于哪个类	111
6.4.2 用一个字符名称调用一个属性或者方法	111
6.5 在 Visual Basic.NET 中创建一个类	112
6.5.1 给一个类添加方法	112
6.5.2 命名属性, 方法和事件	113
6.5.3 事件和事件处理	113
6.5.4 委托 (delegates)、safe function points 以及 addressof 的概念	116
6.5.5 声明和引发一个事件的例子	117
6.5.6 事件处理的例子	118
6.5.7 创建一个类的实例	121

6.6	Visual Basic.NET 的接口 (Interfaces)	128
6.7	类的继承 (Inheritance)	130
6.7.1	继承的规则	131
6.7.2	用继承建立一个继承类	132
6.7.3	重载 Windows 控件	133
6.7.4	什么时候使用继承	135
6.7.5	命名空间 (Namespace)	138
6.7.6	关于继承的例子	139
6.8	多态性	143
6.8.1	用继承实现多态性	143
6.8.2	用接口来实现多态性	144
第 7 章	数据库访问技术	147
7.1	理解数据库	147
7.2	Visual Basic.NET 开发数据库初步, 数据绑定显示	148
7.2.1	数据集 (DataSets) 的概念	148
7.2.2	在 Windows 窗体中显示单个表的数据	150
7.2.3	在窗体上显示带参数的数据查询	156
7.2.4	用 Data Form Wizard 生成数据绑定控件	162
7.2.5	利用 Data Form Wizard 创建一个 Master/Detail 窗体	166
7.3	SQL 语言基础	170
7.3.1	SQL 发展历史	170
7.3.2	数据类型	171
7.3.3	运算符	172
7.3.4	变量	173
7.3.5	流控制语句	173
7.4	SQL 语句函数	175
7.4.1	日期函数	175
7.4.2	字符串函数	175
7.4.3	数学函数	176
7.4.4	集合函数	177
7.4.5	文本和图像函数	177
7.4.6	转换函数	177
7.5	表、视图与索引	178
7.5.1	表	178
7.5.2	表数据操作	179
7.5.3	索引	181
7.5.4	视图	182
7.5.5	查询	182
7.5.6	统计	185

7.5.7 利用查询结果创建新表	187
7.5.8 使用 UNION 运算符实现多查询联合	187
7.5.9 连接	188
7.5.10 子查询	189
7.6 数据库访问对象 DAO	190
7.6.1 何时使用 DAO	190
7.6.2 DAO 和 Jet 数据库引擎	191
7.6.3 DAO 对象模型	192
7.7 远程数据对象 RDO	197
7.7.1 RDO 对象模型	197
7.8 ODBC API	202
7.8.1 ODBC 结构	202
7.8.2 使用 ODBC API 访问数据库	204
7.9 ADO 数据对象	206
7.9.1 ADO 对象模型	206
7.10 ADO.NET	211
7.10.1 ADO.NET 简介	211
7.10.2 使用 ADO.NET 的基本方法	212
7.11 用面向对象思想处理数据库的例子	214
第 8 章 Visual Basic.NET 的多线程	232
8.1 自由线程 (Free Threading)	232
8.1.1 建立和使用一个新的线程	232
8.1.2 关于线程的参数和返回值的问题	232
8.1.3 并发性	233
8.2 自由线程的举例	234
第 9 章 VB.NET 的数组、文件和出错处理	238
9.1 数组	238
9.1.1 数组 (Arrays)	238
9.1.2 数组列表 (ArrayLists)	239
9.2 出错处理	240
9.2.1 出错处理	240
9.2.2 多种出错处理 (Multiple Exceptions)	241
9.2.3 忽略错误 (Throwing Exceptions)	243
9.3 文件处理 (File Handling)	243
9.3.1 文件对象 (file object)	243
9.4 文件处理的出错处理	244
9.5 检测文件的结束 (Testing for End of File)	245
9.6 文件的静态方法 (Static File Methods)	246

9.7 VBFile 类	246
第 10 章 从 VB 6.0 到 VB.NET	249
10.1 语言的变化	249
10.1.1 语法的变化	249
10.1.2 函数中增加的语句	251
10.1.3 Visual Basic.NET 中的数字类型	251
10.1.4 赋值号的变化	251
10.2 面向对象的变化	252
10.2.1 Properties 的变化	252
10.2.2 Visual Basic.NET 的类	253
10.2.3 窗体及控件的变化	254
10.3 Windows API 函数的变化	255
10.4 一个应用程序的比较	257
10.4.1 应用程序的界面	257
10.4.2 Visual Basic 6.0 中的工程	258
10.4.3 Visual Basic.NET 中的工程	259
10.4.4 代码比较	264

第 1 章 Visual Basic.NET 入门

本章包括:

Visual Basic.NET 概述

Visual Basic.NET 的新特点

Visual Basic.NET 的集成开发环境

Visual Basic.NET 的第一个应用程序

1.1 Visual Basic.NET 概述

Visual Basic.NET (又称 Visual Basic 7.0 或 VB 7.0) 是 Microsoft 公司在 Visual Basic 6.0 之后推出的最新版本, 集成在 Visual Studio.NET 7.0 中, 与 Visual C++ 7.0 以及 C# 组成了 “.NET” 构架。

Visual Basic 是 Windows 环境下简单、易学、高效的一种编程语言, 其快速开发的特性深受程序员的喜爱, 但是 .NET 以前版本的 Visual Basic 面向对象的能力远远不能满足程序员的需要, 这也是越大项目越少用到 Visual Basic 的原因。Visual Basic.NET 新增和加强了许多新的面向对象的特性比如继承、重载等等。语言的新特点也包括了对进程的控制和低层结构的操作, 这些新特性使得 Visual Basic.NET 再次成为程序员关注的焦点。

1.2 Visual Basic.NET 的新特点

一直以来, 大家总是针对 Visual Basic 究竟是“面向对象”还是“基于对象”, 甚至“面向组件”的语言而争论不休。但是, 这些争论在 Visual Basic.NET 面前, 马上就会偃旗息鼓——因为, 不管依照怎样的标准来定义“面向对象”, Visual Basic.NET 绝对是符合面向对象的每一个标准的。下面就来看看它有着怎样的特性。

1. 构造函数

当一个对象被创建的时候, 它是否能够被正确地初始化, 这是我们比较关心的问题。利用构造函数我们就可以同时给这个对象中的成员赋初值, 这样有助于我们正确地初始化一个对象。Visual Basic 以前的版本中, 在创建对象的时候, 必须要等到对象创建完毕之后, 另外调用一个独立的方法来对它进行初始化。在 Visual Basic.NET 中, 我们可以利用构造函数给对象赋初值——这样我们就不需要再去进行烦琐的调用赋初值了。构造函数简化了编码的过程, 也减少了出错的机会。

2. 封装特性

从字面上理解,封装就是将某事物包围起来,使外界不知道其实际内容。在程序设计中,封装的意思就是程序提供一个包含了一系列过程和函数的接口,其他的程序可以直接利用对象的这个接口,而不需要去了解接口里面的具体代码。接口中的代码一般来说总是实现一些相对完整的功能。

3. 自由线程

线程是进程中的一个实体,是被系统独立调度和分派的基本单位。线程是一个很复杂的概念,一时半会儿解释不了。但是,基本上来讲,线程的意思就是让我们的程序可以在同一时间段内做两件以上的事情。比如,我们也许需要让程序在后台打印一个文档,同时还要让它响应当前用户不断发出的命令。线程在创建具有高度扩展性服务端程序的时候显得比较有用,并且,利用线程来编写具有极强交互性用户界面的时候也十分容易,例如我们可以为一个需要长时间运行的程序加上一个“Cancel”按钮,以使用户有更多的选择。以前 Visual Basic 编程的时候对线程想尽了办法,调试线程的复杂和不稳定性使线程编写相当的复杂(因为只能使用 API 来写,还是伪线程的,说是多线程不如说是多进程恰当)。现在,Visual Basic 提供了编写线程的能力,线程编写在调试上是相当复杂的,Visual Basic.NET 终于以一种相对简单的形式实现出来。

4. 继承

继承是面向对象系统中的另一个重要的概念,一个语言是否具有继承特性常被人们用作判断是否是面向对象语言的关键标准。继承所表达的就是一种类的相交关系。它使得某类可以继承另外一种类的特征和能力,类间具有继承关系应有三种特征:类间具有共享特征;类间具有细微的差别或新增部分;类间具有层次结构。

从对象的角度来看,继承是这样的一种概念:如果一个对象能够获得另外一个对象的接口和方法,并且可以扩展这些接口和方法,我们就称这个对象继承了另一个对象。举个实际的例子,我们在产品存储程序中,可以创建一个通用的处理所有产品的产品对象“Product”。从这个对象中,根据是否需要上税,又可以派生出一个免税产品对象“NonTaxableProduct”和一个需税产品对象“TaxableProduct”。两个对象都将继承原始产品对象 Product 的接口和所有方法,但是将根据各自不同的实际情况在需要的地方修改或者扩展原始对象的方法。

5. 初始化函数

以前在 Visual Basic 中还需要首先声明一个变量,然后才能对它赋初值——这样用户需要写两行代码。初始化函数可以让用户将这两个步骤合并在一行代码中完成。虽然这是一个微小的变化,但是从代码的角度来看,这个改进提供了更少、更简单、更容易维护的代码。

6. 基于对象

对于一种计算机语言来讲,做到基于对象要比面向对象相对容易些。不过,什么是基于对象,到现在也只有一个很模糊的定义——大概是指一种语言具有直接和对象进行交互的能力。VB 3.0 就已经是基于对象的语言了,当时它能够和 DAO 对象以及控件交互;后来随着版本的提升,这些可交互的对象已经扩大到了 ActiveX 控件、RDO、ADO 对象等。

7. 面向对象

一种面向对象的语言至少需要满足三个条件。封装性(Visual Basic 4.0 中就已经实现了),继承性(Visual Basic.NET 中才有),多态性(多态性指的是多形态,就是对各个对象

的相同接口, Visual Basic 3.0 中就有了)。所以在 Visual Basic.NET 中, 实现了完全的面向对象, 因为它已经完全满足了这三个条件。

8. 重载

重载的含义是指通过为函数或运算符创建附加定义而使它们的名字可以重载。也就是说相同名字的函数或运算符在不同场合可以表现出不同的行为。对象们经常在类似的数据上开展同样的功能, 比如, 我们需要能够加服务和产品项到一个订购对象上。现在, 我们必须分开对订购对象执行一个方法(添加服务)和另一个方法(添加产品项)。利用函数重载, 我们仅需要执行 ASS 方法, 然后让 Visual Basic 语言自己根据情况去选择执行正确的函数。

9. 覆盖

当使用继承功能的时候, 新类从父类那里得到了一切方法。但是, 也许我们需要让其中的某一种或者几种函数执行另外的动作, 可以通过覆盖原始的方法来实现这样的功能。我们的新代码也可以调用原始的父亲对象的原始方法, 这样的过程可以使用 XX 关键字来实现(比如 MY)。

10. 多态性

简单地说, 这就是让两个不同的类型的对象, 执行同一种方法的能力。另一种解释的方法是, 描述同一个消息可以根据发送消息对象的不同采用多种不同的行为方式。Visual Basic 支持后来的捆绑, 而在 Visual Basic 5.0 中通过执行关键字和多个 COM 接口来实现多态。现在, 在 Visual Basic 版本中, 我们将看到继承接口同样也可以具有多态性。最后, Visual Basic.NET 在执行多态性的时候给程序相当大的灵活性。

11. 共享成员

共享成员又叫做静态成员和类级成员。共享成员就是对类的所有的实例来说, 相同的方法或变量。每一个对象创建的时候, 基于一个给定的类。分享这些相同的变量和函数。这意味着, 比如, 我们可以计算出一个类有多少个实例创建了, 只要我们声明一个共享的计数器变量。

12. 结构化错误处理

替代了不稳定的也极不灵活的“On Error Goto”语句, 我们有一个新的结构来处理错误。这个结构包含了“TRY”、“CATCH”和“END TRY”等关键字。在“TRY”和“CATCH”之间的代码块被保护, 如果有个错误出现了, 这个“CATCH”代码块就会运行, 在“CATCH”运行完之后, 模块内的代码就做错误结束处理完之后的清除和收尾的工作。

13. 类型安全保证

今天, Visual Basic 在变量的类型转换上显得特别前卫。它会自动将数字转换成字符串打印出来, 并且还有更多的类似的功能。有时候这样的情况会导致不可知的副作用。有了类型安全控制后, 我们将可以自己选择让 Visual Basic 避免自动做类型转换, 在程序编写时避免一些不可预知的错误是十分有用的。

14. 用户界面继承

光有语言继承还是不够的, Visual Basic.NET 还可以实现表单的继承特性。这意味着我们能创建一个基本的表单模板, 也许模板中包含一个标准的工具条、公司标志和颜色, 然后从这个模板中派生出我们所需要的其他模板。所有其他的模板将从原始模板继承模板的样式和代码。一个对于原始模板的改变, 可以自动地波及所有由此模板派生出来的其他表单, 只需要简单的编译即可。

15. Web 表单

很多年以来, Visual Basic 很广泛地被认为是强大的窗体 (Form) 设计工具。我们也喜欢并习惯于在控件上双击出来一个编码窗口, 然后我们就可以在窗口里面写那个控件的事件了。WEB 表单将这个特性带给了 HTML 开发。WEB 表单是浏览器为平台的, 所有的代码都运行在 WQEN 服务器上, 但同时它给我们拉放表单设计和双击控件编写服务端的时间代码的能力, 而且这是真正的 Visual Basic 代码不是 VBS 代码。

16. Web 服务

一个 Web 服务是一个组件, 运行于一个 Web 服务器上并且允许客户程序通过 HTTP 调用其方法。在组件上的每个方法都表现为一个 URL 并且可以返回数据 (或许是 XML 文档) 和接受参数输入。这样的技术基于的是开放 SOAP 标准, 所以现在用户的服务端组件实际上可以被任何客户访问到, 而不管是什么语言或者平台。

有了这些改变及上述提到的这些新特性, Visual Basic.NET 成为了一个完全的面向对象的语言, Visual Basic 仍然是世界上最广泛使用的开发工具。

1.3 Visual Basic.NET 的集成开发环境

Visual Basic.NET 的集成开发环境 (IDE) 集成了许多功能, 如设计、编辑、编译、调试等, 新建一个 Visual Basic 项目时, 可以看到集成开发环境的界面, 如图 1.1 所示。

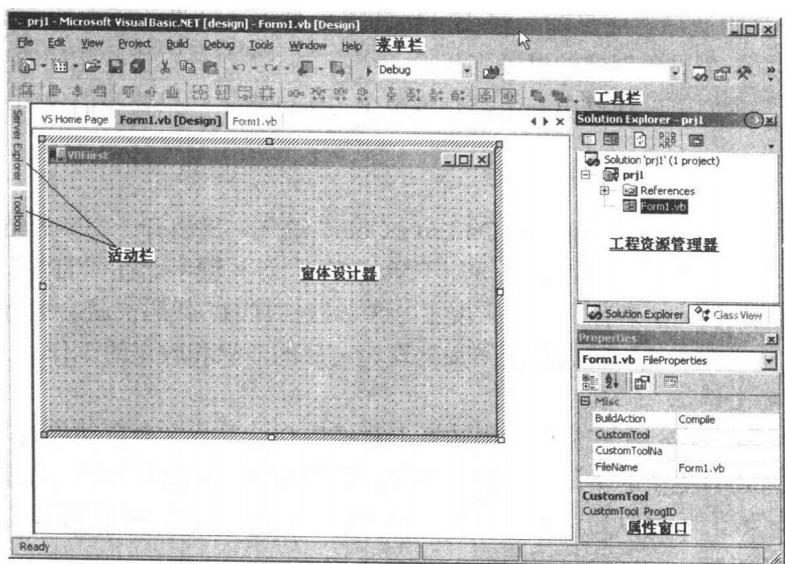


图 1.1 Visual Basic 集成开发环境

Visual Basic.NET 集成开发环境由以下元素组成:

- 菜单栏

显示所有 Visual Basic.NET 使用的命令与设置。除了提供标准的“File” (文件)、“Edit” (编辑)、“View” (视图)、“Tools” (工具)、“Windows” (窗口)、“Help” (帮助) 之外,

还提供了“Project”（项目）、“Format”（格式）、“Debug”（调试）、“Build”（全程编译）的功能菜单。

● 工具栏

在编程环境下提供对于常用命令的快速访问。单击工具栏上的按钮，则执行该按钮所代表的操作。将鼠标光标悬浮按钮上，会有按钮所对应的功能提示弹出。Visual Basic.NET 新增了“Debug”（调试）和“Search”（查找）的功能，方便了程序的调试。其他按钮可以从“View”（视图）菜单里的“Toolbar”（工具栏）子菜单选定。工具栏可依次紧贴在菜单之下，或以垂直条状紧贴在左边框上，如果将它从工具栏中拖开，则它将“悬”在窗口中。

● 工程资源管理窗口

列出当前工程中的所有的窗体和模块，可以按照“浏览器”（Solution Explorer）的形式浏览，也可以按“类试图”（Class View）的形式浏览。

● 属性窗口

列出所选定对象的属性值，为对象的初始化赋值。

● 窗体设计器

作为应用程序编写的基础，窗体设计器承担了应用程序界面的设计任务。在窗体上添加控件、组件、图形、图片等对象来设计出应用程序的外观。应用程序中的每一个窗体都有自己的窗体设计器窗口。

● 活动栏

Visual Basic.NET 新添了“活动栏”，可以将“服务器浏览器”及“工具箱”等收在左边框上，也可以单击“活动栏”，再单击活动栏上的“”按钮，使活动栏与工程资源管理器对称，同时“”变为“”形状，再单击“”，活动栏恢复原来状态。

● 代码编辑窗口

双击窗体设计器或单击工程资源管理器上的“”图标，则出现代码编辑窗口，如图 1.2 所示。



图 1.2 代码编辑窗口

可以看到代码编辑窗口新增了“”图标，点击这个图标后，图标后面对应的过程或函数将被挂起，只保留过程或函数的名字，整个代码界面将显得十分简洁。“对象列表框”显示所选对象的名称。单击列表框右边的箭头，显示与该窗体有关的所有对象的清单；“过程列表框”列出对象的过程或事件，在该列表框中选定过程的名称，选取该列表框右边的箭头可以显示这个对象的全部事件。

1.4 Visual Basic.NET 的第一个应用程序

创建 Visual Basic.NET 应用程序有以下几个主要步骤：

- (1) 创建新的 Visual Basic 工程
- (2) 创建应用程序的用户界面
- (3) 设置用户界面中各对象的属性
- (4) 编写程序代码
- (5) 保存和运行程序
- (6) 创建可执行文件

为了实现上面这些过程，可以按照以下步骤来创建一个简单应用程序，该应用程序由一个文本框 (TextBox) 和一个按钮 (Button) 组成。单击按钮，文本框会出现“Hello,World!”，同时按钮上的文本由“Birthing”变成“Birthed”。

1.4.1 创建新的 Visual Basic.NET 工程

在 .NET 框架中创建 Visual Basic.NET 的过程有一点类似用 MFC 创建 VC 应用程序的过程，需要在向导中填写必要的信息，它的过程如下：

- (1) 打开 Visual Studio.NET。
- (2) 点击 File→New→Project...，弹出如图 1.3 所示的对话框。

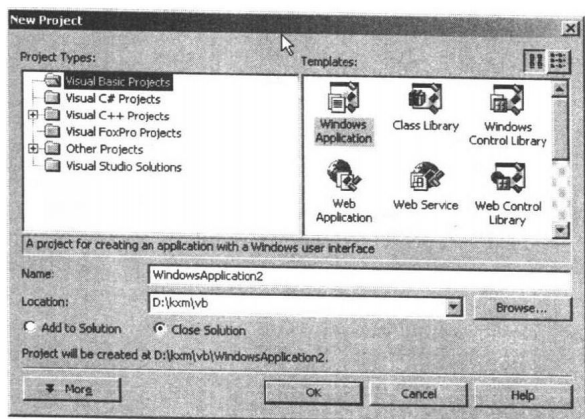


图 1.3 创建新的 Visual Basic.NET 工程

- (3) 在“Project Type”（项目类型）中选择“Visual Basic Projects”（Visual Basic 项目）

选项。

(4) 在“Templates”中选择“Windows Application”(Windows 应用程序)选项。

(5) 在“Name”中添加应用程序的名称(默认为“WindowsApplication&”, &为一常数)。

(6) 在“Location”(位置)中添加应用程序所在目录的名称,也可以通过“Browse...”直接选择目录。

(7) 其他项采用默认值,单击“OK”即可生成应用程序。

1.4.2 创建应用程序的用户界面

窗体是创建应用程序的基础。通过使用窗体可将窗体和对话框添加到应用程序中。也可以把窗体作为项的容器,这些项是应用程序界面中的不可视部分。例如,应用程序中可能有一个作为图形容器的窗体,而这些图形是打算在其他窗体中显示的。

建造 Visual Basic 应用程序的第一步是创建窗体,这些窗体将是应用程序界面的基础。然后在创建的窗体上绘制构成界面的元素,通常是在工具箱中选择控件或组件,对于当前的应用程序,可在工具箱中选择两个控件,具体过程如下:

(1) 单击活动栏“Toolbox”,再单击要绘制的控件的工具,在这里是“TextBox”控件。

(2) 将指针移到窗体上,该指针变成十字线。

(3) 将十字线放在控件的左上角。

(4) 拖动十字线画出适当的大小。

(5) 释放鼠标按钮。

(6) 以相同方法在窗体上添加“Button”控件。

经上述操作之后,控件将会出现在窗体上。

在窗体上添加控件的另一个简单方法是双击工具箱控件中的按钮。这样会在窗体缺省位置创建一个尺寸为缺省值的控件;然后再将该控件移到窗体中的其他位置。

对于绘制窗体的控件,有时还要调整它的大小、移动和锁定控件,下面对这些操作进行详细介绍。

● 调整控件尺寸

(1) 用鼠标单击需要调整大小的控件,选定的控件上出现尺寸句柄。

(2) 将鼠标指针定位到尺寸句柄上,拖动该尺寸柄直到控件达到所希望的大小为止。角上的尺寸句柄可以同时调整控件水平和垂直方向的大小,而边上的尺寸句柄调整控件一个方向的大小。

(3) 释放鼠标按钮。

也可以按【Shift】键加上箭头键调整选定控件的大小。

● 移动控件

(1) 用鼠标把窗体上的控件移动到一个新位置或用属性窗口改变“Top”和“Left”属性。

(2) 选定控件后,可用【Ctrl】键加箭头键每次移动控件的一个网格单元。如果该网格关闭,控件每次移动一个像素。

● 锁定所有控件位置

这个操作将把窗体中的所有控件锁定在当前位置,以防止已处于理想位置的控件因为