



浩强创作室

HAO QIANG STUDIO

精通

Visual Basic.NET

中文版

刘炳文 编著



机械工业出版社
CHINA MACHINE PRESS



精通 Visual Basic.NET 中文版

刘炳文 编著



机械工业出版社

本书通过大量实例,系统全面地介绍了 Visual Basic. NET 程序设计。内容包括 Visual Basic. NET 程序开发环境、基本数据类型、数据输入输出、常用 Windows 窗体控件、Visual Basic. NET 控制结构、复合数据类型(数组、结构、集合)、通过程、键盘与鼠标事件、菜单程序设计、多窗体程序设计、多文档界面(MDI)、对话框程序设计、数据文件、类的建立、继承、多态、图形程序设计、自定义 Windows 窗体控件等。本书基于正式发布的 Visual Basic. NET 编写,内容翔实,概念准确,编排合理,适合初学者自学,也可作为高等院校非计算机专业或计算机专业学生的参考书。

图书在版编目(CIP)数据

精通 Visual Basic. NET 中文版/刘炳文编著. —北京:机械工业出版社,2004.1
ISBN 7-111-12919-9

I. 精... II. 刘... III. BASIC 语言-程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2003)第 072540 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

策 划:胡毓坚

责任编辑:赵 慧

责任印制:施 红

三河市宏达印刷有限公司印刷·新华书店北京发行所发行

2004 年 1 月第 1 版·第 1 次印刷

787mm×1092mm $\frac{1}{8}$ ·45.25 印张·2 插页·1122 千字

0 001—4 000 册

定价:66.00 元

凡购本图书,如有缺页、倒页、脱页,由本社发行部调换

本社购书热线电话:(010) 68993821、88379646

封面无防伪标均为盗版

前 言

2002年3月22日,Microsoft在北京正式发布了 Visual Studio .NET(VS.NET)中文版,同时发布了微软.NET应用程序的开发工具和平台.NET框架(.NET Framework)。据说,Visual Studio .NET和.NET框架是比尔·盖茨作为微软首席软件设计师以来首次完全监控的软件。比尔·盖茨预言:“我们将 Visual Studio .NET 视为下一次开发机遇,是建立 XML Web 服务的关键,它很快会成为所有主要新软件开发的基础。”微软对 Visual Studio.NET 的定位十分明确,即.NET是下一个网络时代的标准平台,而 Visual Studio.NET 是支持这一平台的标准开发工具。

Visual Basic.NET 以 Visual Studio.NET 为基础,是 Visual Studio.NET 的主要组成部分。它虽然从 Visual Basic 6.0 发展而来,但它不是对 Visual Basic 6.0 的简单改良,而是从基础开始,建立了一个新的平台。Visual Basic.NET 是在现有 Visual Basic 6.0 基础之上的一次重大飞跃,它是一个全新的平台,而不是 Visual Basic 6.0 的一个升级版本。Visual Basic.NET 与 Visual C++ .NET、Visual C# .NET 等语言共用一个编程环境,具有相同的基本数据类型和用户定义类型以及类和接口,实现了不同语言的交互。统一的多语言企业级开发环境使每个程序员能分享相同的体验,不必再为语言的优劣而争论不休,不必再为语言的变化而重新熟悉环境,不必再学习不同的类库和 APIs,不必再为 ASP 代码的管理和调试苦恼,等等,这些优势都为 Visual Studio.NET 的推广应用奠定了坚实的基础。

与以前的版本相比,Visual Basic.NET 的变化是革命性的,尤其是在面向对象方面。为了实现面向对象的程序设计,Visual Basic.NET 引入了很多新的和改进的性能,包括名称空间、继承、接口、重载、覆盖、成员共享、多态等,从而使 Visual Basic.NET 成为一种强大的、真正面向对象的编程语言。

本书通过大量实例,深入浅出、系统全面地介绍了 Visual Basic.NET 程序设计。全书共 20 章,内容包括:Visual Basic.NET 程序开发环境、基本概念(类、对象、名称空间等)、基本数据类型、数据输入输出、常用 Windows 窗体控件、Visual Basic.NET 控制结构、复合数据类型(数组、结构、集合)、通用过程、键盘与鼠标事件、菜单程序设计、多窗体程序设计、多文档界面(MDI)、对话框程序设计、数据文件、类的建立、继承、方法重载、共享、覆盖、隐藏、虚拟方法、抽象基类、接口、委托、多态、图形程序设计、建立自定义的 Windows 窗体控件等。

严格地说,Visual Basic.NET 是一种新的语言,而不是原有 Visual Basic 的简单升级。它提供了一个全新的环境,改变了以往的软件开发规则,需要从头开始学习和适应。在编写本书时,笔者已充分注意到了这一特点。

本书由浩强创作室组织策划,由刘炳文老师编著。本书内容较为全面,但它是为适应初学者的需要而编写的,它不要求读者具有专门的计算机专业知识,也不要求有其他高级语言(包括 Visual Basic)的编程经验。也就是说,只要具有 Windows 2000 的基础知识,就可以从头开始学习本书,并能达到较高的水平。针对初学者的特点,本书在体系结构和内容编排上注意了由简及繁、由浅入深、循序渐进以及理论与实践的密切结合,力求概念准确、叙述顺畅、通俗易

懂。考虑到 Visual Basic.NET 用于应用软件的开发,本书中的例题主要用来加深对概念的理解。只有真正理解了这些基本概念,才能用 Visual Basic.NET 设计较为复杂的和有实用价值的应用程序;在掌握了本书的内容之后,就可以登堂入室,达到更高的境界。

Visual Basic.NET 功能强大,内容十分丰富,涉及很多方面,要在一本书中面面俱到地讲述全部功能是不现实的。本书篇幅不小,但也只能介绍其中的一部分内容,这些内容是在用 Visual Basic.NET 开发应用程序时必须用到和经常用到的。需要说明的是,本书从 Visual Basic.NET 的基础入手,适用于初学者,但内容比较全面,不限于基础知识。

本书文稿的录入、程序的编写、运行以及插图的截取都是在 Windows 2000 环境下进行的,所有程序都已在 Visual Basic.NET 中文版环境中顺利运行通过。

在我国,Visual Basic 已经得到了广泛的应用,一些院校已把它列入教学计划。Visual Basic.NET 的推出和推广普及,无疑会使 Visual Basic 的应用更加深入。本书既适用于初学者,同时也适用于具有一定程序设计实践的 Visual Basic 用户,衷心希望它能在 Visual Basic.NET 的推广和普及中起到一些作用。由于笔者水平有限,实践经验不多,可供参考的资料太少,加之时间仓促,错谬疏漏之处在所难免,诚望专家和广大读者不吝赐教,批评指正。

作 者

目 录

前言

第 1 章 Visual Basic.NET 程序开发环境	1
1.1 概述	1
1.1.1 Visual Basic.NET 的新特点	1
1.1.2 Visual Basic.NET 的软硬件需求	3
1.2 从 Visual Basic 6.0 到 Visual Basic.NET	4
1.2.1 功能的变化	4
1.2.2 关键字的变化	5
1.3 启动和退出 Visual Basic.NET	6
1.4 标题、菜单和工具栏	11
1.4.1 标题栏和菜单栏	11
1.4.2 工具栏	13
1.5 起始页窗口	14
1.6 Windows 窗体设计器、解决方案资源管理器与属性窗口	19
1.6.1 Windows 窗体设计器和解决方案管理器窗口	19
1.6.2 属性窗口和工具箱窗口	22
1.7 窗口管理	23
1.7.1 主窗口	23
1.7.2 其他窗口	25
第 2 章 类、对象与名称空间	28
2.1 面向对象和类	28
2.1.1 面向对象	28
2.1.2 类和对象	31
2.2 名称空间	32
2.2.1 什么是名称空间	32
2.2.2 .NET 中的名称空间	33
2.2.3 名称空间的使用	37
2.3 对象	40
2.3.1 对象的属性、事件和方法	40
2.3.2 对象属性设置	42
2.4 窗体	44
2.4.1 窗体的结构与属性	45
2.4.2 窗体事件	50
2.5 控件	50

2.5.1	Windows 窗体控件	51
2.5.2	工具箱的管理	52
2.6	控件的画法和基本操作	54
2.6.1	控件的画法	54
2.6.2	控件的基本操作	55
第 3 章	简单 Visual Basic.NET 程序设计	57
3.1	语句	57
3.2	编写简单的 Visual Basic.NET 应用程序	60
3.2.1	用 Visual Basic.NET 开发应用程序的一般步骤	60
3.2.2	编写 Visual Basic.NET 应用程序	61
3.2.3	代码编辑器	67
3.3	程序的保存和装入	73
3.3.1	保存程序	74
3.3.2	装入程序	74
3.4	程序的运行	76
3.5	Visual Basic.NET 应用程序的工作方式	77
第 4 章	Visual Basic.NET 程序设计基础	79
4.1	数据类型	79
4.1.1	基本数据类型	79
4.1.2	枚举类型	82
4.2	常量和变量	85
4.2.1	常量	85
4.2.2	变量	88
4.3	变量的隐式声明与类型转换	91
4.3.1	变量的隐式声明	91
4.3.2	类型转换	95
4.4	变量的作用域和生存期	98
4.4.1	变量的作用域	98
4.4.2	变量的生存期	100
4.5	常用内部函数	101
4.6	字符处理与字符串函数	104
4.6.1	Visual Basic.NET 的字符处理机制	104
4.6.2	字符串函数	106
4.7	运算符与表达式	110
4.7.1	算术运算符	110
4.7.2	复合运算符	112
4.7.3	关系运算符和逻辑运算符	113
4.7.4	表达式的执行顺序	117
第 5 章	数据输入输出	119

5.1 数据输入——InputBox 函数	119
5.2 MsgBox 函数和 MsgBox 语句	123
5.2.1 MsgBox 函数	123
5.2.2 MsgBox 语句	127
5.3 MessageBox.Show 方法	128
5.4 控制台输入输出	130
5.4.1 控制台输入输出方法	130
5.4.2 控制台应用程序的建立和执行	132
5.5 在集成环境外建立控制台应用程序	136
5.5.1 Visual Basic .NET 编译器	137
5.5.2 程序举例	138
5.6 数值的格式输出	140
第6章 常用 Windows 窗体控件	143
6.1 文本控件	143
6.1.1 标签	143
6.1.2 文本框	145
6.2 图形控件	149
6.2.1 图片框的主要属性	150
6.2.2 图形文件的装入	151
6.3 按钮控件	154
6.3.1 属性和事件	155
6.3.2 应用举例	156
6.4 选择控件——复选框和单选按钮	158
6.4.1 复选框和单选按钮的属性和事件	158
6.4.2 应用举例	159
6.5 选择控件——列表框和组合框	161
6.5.1 列表框	161
6.5.2 组合框	167
6.6 滚动条	171
6.7 计时器	173
6.8 分组框控件	176
6.9 焦点与 Tab 顺序	179
6.9.1 焦点	179
6.9.2 Tab 顺序	180
第7章 Visual Basic.NET 控制结构	182
7.1 选择控制结构	182
7.1.1 单行结构条件语句	182
7.1.2 块结构条件语句	183
7.1.3 If 函数	187

7.2 多分支控制结构	189
7.3 For 循环控制结构	194
7.4 当循环控制结构	199
7.5 Do 循环控制结构	202
7.6 多重循环	205
7.7 GoTo 语句	209
第 8 章 复合数据类型	211
8.1 数组	211
8.1.1 数组的定义	211
8.1.2 默认数组与嵌套数组	215
8.2 数组的初始化和引用	217
8.2.1 数组的初始化	218
8.2.2 数组元素的引用	220
8.3 动态数组	221
8.3.1 动态数组的定义	221
8.3.2 数组的清除和重定义	223
8.4 数组的基本操作	225
8.4.1 数组元素的输入、输出和复制	225
8.4.2 For Each...Next 语句	229
8.5 程序举例	230
8.6 结构	235
8.6.1 引例	235
8.6.2 结构类型与结构变量的定义	237
8.6.3 结构变量的初始化及其引用	240
8.7 结构数组	243
8.8 集合	246
8.8.1 建立集合并向集合中添加项目	246
8.8.2 集合成员的删除和检索	250
第 9 章 过程	253
9.1 Sub 过程	253
9.1.1 建立 Sub 过程	253
9.1.2 调用 Sub 过程	255
9.1.3 通用过程与事件过程	258
9.2 Function 过程	260
9.2.1 建立 Function 过程	260
9.2.2 调用 Function 过程	262
9.3 参数传送	268
9.3.1 形参与实参	268
9.3.2 引用	270

9.3.3 传值	272
9.3.4 数组参数的传送	275
9.4 可选参数与可变参数	278
9.4.1 可选参数	279
9.4.2 可变参数	280
9.5 局部内存分配	281
9.6 递归	286
9.7 Shell 函数	290
第 10 章 键盘与鼠标事件过程	293
10.1 KeyPress 事件	293
10.2 KeyDown 和 KeyUp 事件	296
10.3 鼠标事件	304
10.3.1 压下、松开与移动鼠标	305
10.3.2 MouseEnter 和 MouseLeave 事件	308
10.4 鼠标光标的形状	311
10.4.1 Cursor 属性	311
10.4.2 设置鼠标光标形状	312
10.5 拖放	314
10.5.1 与拖放有关的属性、类型、事件和方法	314
10.5.2 拖放的实现	317
第 11 章 菜单程序设计	321
11.1 Visual Basic.NET 中的菜单	321
11.1.1 菜单概述	321
11.1.2 主菜单	322
11.2 菜单的属性、事件和方法	323
11.2.1 菜单的属性	323
11.2.2 事件和方法	327
11.3 建立菜单	328
11.3.1 界面设计	329
11.3.2 编写程序代码	331
11.4 菜单项的控制	333
11.4.1 有效性控制	333
11.4.2 菜单项标记	334
11.4.3 键盘选择	338
11.5 弹出式菜单	339
11.5.1 弹出式菜单控件	339
11.5.2 建立弹出式菜单	340
11.6 菜单项的基本操作	343
第 12 章 多重窗体程序设计	346

12.1	Visual Basic.NET 中的窗体	346
12.1.1	设置窗体的特殊效果	346
12.1.2	与多重窗体程序设计有关的方法	348
12.2	建立多重窗体应用程序	350
12.2.1	建立界面	350
12.2.2	编写程序代码	353
12.3	多窗体程序设计中应注意的问题	360
12.3.1	指定启动窗体	360
12.3.2	窗体文件名与窗体名	360
12.4	Visual Basic.NET 程序结构	363
12.4.1	Visual Basic.NET 中的文件类型	363
12.4.2	代码编写顺序	365
12.5	模块与窗体	367
12.5.1	模块与窗体的构成	367
12.5.2	程序举例	369
12.6	闲置循环与 DoEvents 方法	372
第 13 章	多文档界面(MDI)	376
13.1	MDI 窗体与 MDI 子窗体	376
13.1.1	MDI 子窗体	376
13.1.2	引例	377
13.1.3	几点说明	380
13.2	MDI 特性	381
13.2.1	与 MDI 有关的属性、方法和事件	381
13.2.2	Dim 语句和 WindowState 属性	382
13.3	建立 MDI 应用程序	383
13.3.1	建立 MDI 应用程序的一般过程	383
13.3.2	MDI 应用程序举例	384
13.4	MDI 应用程序中的菜单	392
13.4.1	MDI 中菜单的建立	393
13.4.2	MDI 菜单应用程序举例	393
13.5	用 MDI 建立工具栏	397
13.5.1	用 MDI 建立工具条的一般步骤	397
13.5.2	程序举例	398
第 14 章	对话框程序设计	402
14.1	概述	402
14.1.1	对话框的分类与特点	402
14.1.2	自定义对话框	403
14.1.3	通用对话框控件	406
14.2	打开文件对话框	407

14.2.1 主要属性、方法和事件	407
14.2.2 打开文件对话框举例	410
14.3 保存文件对话框	412
14.4 其他对话框	415
14.4.1 颜色(Color)对话框	415
14.4.2 字体(Font)对话框	417
第 15 章 数据文件	422
15.1 文件概述	422
15.2 文件的打开与关闭	424
15.2.1 文件的打开(建立)	424
15.2.2 文件的关闭	427
15.3 文件操作语句和函数	427
15.3.1 文件指针	427
15.3.2 其他语句和函数	428
15.4 顺序文件的写操作	430
15.4.1 Print 函数和 PrintLine 函数	430
15.4.2 与 Print 函数有关的函数	433
15.5 顺序文件的读操作	438
15.6 随机文件	442
15.6.1 随机文件的打开与读写操作	443
15.6.2 随机文件中记录的增加、代换与删除	450
15.7 用控件显示和修改随机文件	453
15.8 二进制文件	457
15.9 流与 System.IO 模型	460
15.9.1 System.IO 模型所提供的资源	461
15.9.2 System.IO 名称空间的功能	462
15.10 流输入/输出程序举例	463
15.10.1 有关的方法	463
15.10.2 程序举例	467
15.11 文件和目录操作	473
15.11.1 文件操作	473
15.11.2 目录操作	475
15.11.3 用 Visual Basic.NET 函数执行文件/目录操作	478
第 16 章 对象和类	481
16.1 Visual Basic.NET 中预定义的类和对象	481
16.1.1 预定义对象和类	481
16.1.2 对象的操作	482
16.2 对象变量	484
16.2.1 对象变量的声明和赋值	484

16.2.2 实例	487
16.2.3 绑定	488
16.3 类的定义	489
16.3.1 Class 语句	490
16.3.2 建立类	491
16.4 属性(Property)过程	500
16.4.1 属性的定义	500
16.4.2 属性过程举例	504
16.5 向类中添加事件	510
16.5.1 有关的语句和关键字	510
16.5.2 事件处理的基本概念	513
16.5.3 建立事件处理程序	514
16.6 构造函数	519
16.7 重载与共享	524
16.7.1 方法重载	525
16.7.2 共享成员	530
第 17 章 继承	534
17.1 继承与派生	534
17.1.1 基类与派生类	534
17.1.2 继承的实现	536
17.2 重载与覆盖	541
17.2.1 方法重载	541
17.2.2 覆盖	547
17.2.3 扩展式覆盖与覆盖重载	551
17.3 虚拟方法与隐藏	554
17.3.1 虚拟方法	554
17.3.2 隐藏	558
17.4 基类、类和对象	566
17.4.1 Me 关键字	566
17.4.2 MyBase 和 MyClass 关键字	570
17.5 构造函数与继承	573
17.5.1 无参构造函数的继承	573
17.5.2 带参数的构造函数	575
17.6 成员的作用域	577
17.7 事件和共享方法的继承	579
17.7.1 事件与继承	579
17.7.2 共享方法的继承	584
17.8 抽象基类	587
17.8.1 MustOverride 与 MustInherit 关键字	587

17.8.2 程序举例	589
第 18 章 接口、委托与多态	592
18.1 接口	592
18.1.1 接口的定义	592
18.1.2 接口的实现	595
18.2 委托	600
18.2.1 委托的声明和调用	600
18.2.2 程序举例	604
18.3 多态与后期绑定	606
18.3.1 什么是多态性	606
18.3.2 后期绑定	607
18.4 通过接口和继承实现多态性	612
18.4.1 用接口实现多态性	612
18.4.2 用继承实现多态性	614
18.5 类、结构和名称空间	616
18.5.1 类和结构	616
18.5.2 类与名称空间	619
18.6 类视图与对象浏览器	621
18.6.1 类视图	621
18.6.2 对象浏览器	622
第 19 章 图形程序设计	626
19.1 GDI+ 坐标系统	626
19.1.1 逻辑坐标与设备坐标	626
19.1.2 GDI+ 坐标系统和主要功能	627
19.2 画笔与像素	628
19.2.1 画笔	628
19.2.2 像素	630
19.3 直线	631
19.3.1 DrawLine 方法	631
19.3.2 画直线	632
19.3.3 清屏	637
19.4 矩形和多边形	639
19.4.1 画矩形	639
19.4.2 画多边形	641
19.5 圆、椭圆、弧和饼图	643
19.5.1 画圆和椭圆	643
19.5.2 画弧和饼图	645
19.6 刷子与填充	647
19.6.1 实心刷子	647

19.6.2 阴影刷子	650
19.6.3 纹理刷子	653
19.6.4 渐变刷子	654
19.7 曲线	656
19.7.1 闭合曲线与不闭全曲线	656
19.7.2 贝赛尔(Bezier)曲线	660
19.8 文本输出	663
19.8.1 简单文本输出	663
19.8.2 格式文本输出	668
第 20 章 建立 Windows 窗体控件	675
20.1 基础知识	675
20.1.1 控件源与自定义控件	675
20.1.2 基本概念	676
20.2 控件的基类	677
20.2.1 Control 类	677
20.2.2 UserControl 类	679
20.3 通过继承建立 Windows 窗体控件	680
20.3.1 建立自定义控件	680
20.3.2 测试自定义控件	684
20.3.3 自定义控件图标	686
20.4 建立复合控件	686
20.4.1 建立和测试控件	687
20.4.2 控件的继承	689
20.5 复合控件举例	694
20.6 通过继承 Control 类建立控件	701

第 1 章 Visual Basic.NET 程序开发环境

2000 年, Microsoft 公司推出了一个全新的概念, 即 .NET。到目前为止, 也许很难给它下一个明确的定义, 我们可以把它看成是一个集合、一个环境或一个编程的基础结构。为了实现 .NET 技术, Microsoft 公司开发了一整套工具组件, 这些组件被集成到 Visual Studio.NET 中, 而 Visual Basic.NET 就是它的一个组成部分。也就是说, Visual Basic.NET 与 Visual C#.NET、Visual C++ 等开发工具使用的是同一个开发环境, 这个开发环境就是 Visual Studio.NET。在这一章中, 我们将介绍 Visual Studio.NET 环境中与 Visual Basic.NET 有关的部分。

1.1 概述

Visual Basic.NET 是在 Visual Basic 6.0 的基础上产生的, 应当说, 它仍然是 Visual Basic 语言, 与 Visual Basic 有着相同的基本语法、关键字和编程方式。但是, Visual Basic.NET 具有大量的新特性, 增加了一些新的关键字、语法和数据类型, 同时废弃了一些关键字。这些变化不但大大增强了 Visual Basic 的性能, 而且使用起来更方便。确切地说, 应当把 Visual Basic.NET 看作是与 Visual Basic 6.0 语法相似的一门新语言, 而不是简单的“升级”。

1.1.1 Visual Basic.NET 的新特点

与 Visual Basic 6.0 相比, Visual Basic.NET 有着十分突出的特点, 其中有些特点是 Visual Basic 程序员企盼已久的, 现在终于如愿以偿。总的来看, Visual Basic.NET 有以下主要特点。

1. 统一的集成开发环境

Visual Studio.NET 为 Visual Basic.NET、Visual C++.NET 和 Visual C#.NET 等提供了统一的集成开发环境(Integrated Development Environment, 即 IDE), 其中集成了许多可视化辅助工具, 可以大大简化应用程序的开发, 提高编程效率, 同时使不同语言之间的数据和代码交换更加便利。在 Visual Studio.NET 环境中, 可以直接显示网页, 并采用了更有效的窗口管理策略, 诸如文档标签(选项卡)化、窗口自动隐藏等, 从而可以提高浏览效率、节省屏幕空间。但是, IDE 的改变远比表面显示的更深刻。所有 .NET 语言使用相同的 IDE, 而且 IDE 中的新工具功能强大又易于理解。可以把任何一个窗口设置为自动隐藏(就像自动隐藏 Windows 任务栏那样), 主工作区域是一系列选项卡, IDE 不再同时显示多个窗体和代码模块。当打开对象的源代码时, IDE 自动在它的主工作区域为工作的对象添加一个新的选项卡。

2. 真正的面向对象

长期以来, Visual Basic 一直被人讥笑为基于对象(Object-based)而不是面向对象(Object-oriented)的语言, 现在可以理直气壮地说, Visual Basic.NET 是完全面向对象的语言。从 Visual Basic 4.0 开始, Microsoft 就自称实现了封装性, 但是, 直到 Visual Basic 6.0, 其封装性仍不完善, 例如没有继承性, 当然更谈不上继承的友元。至于多态, 在 Visual Basic 6.0 中只能通过

接口来实现。为了实现面向对象的程序设计, Visual Basic.NET 引入了很多新的和改进的性能,包括继承、接口和重载等,从而使 Visual Basic.NET 成为一种强大的、真正面向对象的编程语言。

3. 丰富的数据类型

Visual Basic.NET 具有十分丰富的数据类型,可以满足各种运算需求。其中整型数就有 8 位、16 位、32 位和 64 位;浮点数除保留了原来的单精度(32 位)和双精度(64 位)类型外,还增加了 128 位的 Decimal 数据类型,该类型数据的精度可达 28 位有效数字。对于算术运算和字符串连接运算,可以使用形如“+”的运算符。此外,在变量的定义、变量及数组的初始化等方面,都提供了与 C 语言类似的功能,这大大方便了代码的编写,提高了编程效率。

4. 改进了的窗体引擎

在 Visual Basic.NET 中, Microsoft 废弃了旧的窗体引擎,代之以 Windows 窗体(Windows Form)。Windows 窗体是制作标准 Win32 屏幕的一种更高级的方法,这种技术最初是为 WFC(Windows Foundation Classes)建立的,而 WFC 是为 Visual J++ 开发的,所有基于通用语言运行库(Common Language Runtime,即 CLR)的语言都使用 Windows 窗体引擎。Windows 窗体的基本构架类似于任何基于 .NET 框架(.NET Framework)的语言中的窗体,它为该框架下的所有语言提供了一套丰富、统一的控件和绘图功能,以及用于图形和绘图的底层 Windows 服务的标准应用程序接口(API)。有了 Windows 窗体,任何图形和屏幕函数就不再需要使用内置的 Windows 图形接口了。

5. 结构化的错误处理

Visual Basic 没有结构化的错误处理方式,它通过 On Error GoTo 语句处理程序中可能出现的错误。这种全局性的错误处理机制缺乏控制,同时增加了编程的工作量。Visual Basic.NET 引入了结构化错误处理功能,它类似于 C++、Java 等语言中的错误处理机制,提供了嵌套、控制和易于理解的块结构,可以完成更健壮的结构化错误处理,并可大大提高代码的可读性。此外, Visual Studio.NET 中的所有语言都使用相同的调试器,可以实现对代码的各种调试,包括对 Visual Basic.NET、脚本和 SQL 等语言的交叉调试、通用语言运行库(CLR)和 Win32 应用程序的调试、对主机或远程主机运行程序的附加调试、多个程序的同时调试等。

6. 方便的 Web 开发

Visual Basic.NET 的一个重要特性是建立 Web 应用程序,即建立运行于 Web 服务器上的 Visual Basic 应用程序。Visual Basic.NET 提供了更为直观、方便的 Web 应用程序开发环境,它不再支持以前版本中的 IIS 应用程序或 DHTML 应用程序,而改为以直接编辑 ASP.NET 的方式开发 Web 应用程序。与以前版本的 ASP 相比,ASP.NET 的功能和效率都有较大的增强,可以大大简化 Web 应用程序的开发,提供更为丰富的用户界面。Web 应用程序的开发主要包括两个主题,即 Web 窗体和 Web 服务。使用 Web 窗体,可以迅速而方便地通过 ASP.NET 建立 Web 应用程序的用户界面。Web 窗体页面是现有 Web 开发工具的革命性的进步,它兼有速度和 RAD(Rapid-Application Development,即快速应用程序开发)开发环境的强大功能。Web 窗体可以在任何浏览器或移动设备上输出,并且能自动提交正确的、与浏览器风格、布局兼容的 HTML。利用 Web 服务,可以通过 Internet 协议调用其他组件或应用程序。它允许使用标准协议(如 HTTP)进行数据交换,而且能通过 XML 消息移动数据。Web 服务不依赖于某种特定组件技术或对象调用规范,因此无论使用什么操作系统,无论用什么语