

普通高等学校“十二五”规划教材

Visual Basic.NET 程序设计

周元哲 编著



西安电子科技大学出版社
<http://www.xduph.com>

普通高等学校“十二五”规划教材

Visual Basic.NET 程序设计

周元哲 编著

西安电子科技大学出版社

内 容 简 介

本书共分 12 章, 内容包括 VB.NET 技术基础、Hello World.vb 和基本控件、VB.NET 语言基础、流程控制结构、数组与集合、VB.NET 的过程、算法设计、用户界面设计、面向对象程序设计基础、文件操作、数据库应用程序开发、异常处理和程序调试等。本书内容精练、由浅入深, 注重学习的连续性和渐进性, 章节之间的实例具有关联性。

本书适合作为高等院校计算机相关专业 VB.NET 程序设计课程的教材或教学参考书, 也可以作为各类计算机应用开发技术人员的应用参考书, 还可以作为全国计算机等级考试、软件技术资格与水平考试的培训资料。

图书在版编目(CIP)数据

Visual Basic.NET 程序设计/周元哲编著. —西安: 西安电子科技大学出版社, 2014.11

普通高等学校“十二五”规划教材

ISBN 978-7-5606-3439-5

I. ① V… II. ① 周… III. ① BASIC 语言—程序设计—高等学校—教材 IV. ① TP312

中国版本图书馆 CIP 数据核字(2014)第 246300 号

策 划 云立实

责任编辑 王 斌 云立实

出版发行 西安电子科技大学出版社(西安市太白南路2号)

电 话 (029)88242885 88201467 邮 编 710071

网 址 www.xduph.com 电子邮箱 xdupfb001@163.com

经 销 新华书店

印刷单位 陕西华沐印刷科技有限责任公司

版 次 2014 年 11 月第 1 版 2014 年 11 月第 1 次印刷

开 本 787 毫米×1092 毫米 1/16 印张 15.5

字 数 365 千字

印 数 1~3000 册

定 价 29.00 元

ISBN 978-7-5606-3439-5/TP

XDUP 3731001-1

*** 如有印装问题可调换 ***

本社图书封面为激光防伪覆膜, 谨防盗版。



前 言

Visual Basic 程序设计语言一直以来以简单易用、功能强大,深受广大编程爱好者的青睐。继 Visual Basic 6.0 后,新近发布的 Visual Basic.NET(VB.NET)可以说是具有划时代意义的产品,它是运行于 Windows 平台上的交互式可视化集成开发环境,也是微软(Microsoft)的 Visual Studio.NET 的四种编程语言之一。

本书结合作者多年基于 Visual Basic 开发软件项目的实际经验,注重基本理论和基本技能。在内容的选取上力图精简,摒弃陈旧和繁杂的语法规则,只介绍该语言的一些基本语法规则和面向对象的基本特征,主要培养学生掌握 Visual Basic 程序设计的基本方法及提高其应用开发能力。

Visual Basic.NET 是完全面向对象的程序设计语言,具有简单、现代、类型安全、性能优良等特点,是面向对象程序设计教学的主干语言之一。本书从实用的角度对 Visual Basic.NET 进行了全面阐述,全书共分 12 章,内容包括 VB.NET 技术基础、Hello World.vb 和基本控件、VB.NET 语言基础、流程控制结构、数组与集合、VB.NET 的过程、算法设计、用户界面设计、面向对象程序设计基础、文件操作、数据库应用程序开发、异常处理和程序调试。

在编写本书的过程中,西安邮电大学计算机科学与技术学院的郝羽及管理工程学院的高华阅读了部分手稿,并进行了修正。西安邮电大学计算机科学与技术学院的宋辉、陈皓、王玉清,陕西新北邦高科技开发有限公司开发部长皮煜峰,西安电子科技大学出版社的云立实、王斌编辑等对本书提出了很多宝贵的意见。另外,本书参阅了大量中英文专著、教材、论文、报告及网上的资料,由于篇幅所限,未能一一列出。在此,对以上老师和作者一并表示敬意和衷心的感谢。

本书的所有程序都在 VB.NET 2005 的集成开发环境(IDE)中进行了调试和运行。由于作者水平有限,时间紧迫,本书难免有不足之处,恳请广大读者批评指正。作者的电子邮箱是 zhouyuanzhe@163.com。

编 者

目 录

第1章 VB.NET 技术基础.....1	3.2.2 变量声明.....31
1.1 计算机基础知识.....1	3.2.3 变量赋值.....31
1.1.1 计算机组成.....1	3.3 常量.....31
1.1.2 软件和程序.....2	3.4 运算符.....32
1.1.3 程序设计语言.....2	3.4.1 算术运算符.....32
1.2 VB 发展历史.....3	3.4.2 连接运算符.....33
1.3 .NET 框架理念.....4	3.4.3 关系运算符.....34
1.4 VB.NET 的功能及特点.....5	3.4.4 逻辑运算符.....34
1.5 VB.NET 的集成开发环境.....6	3.4.5 复合运算符.....35
1.5.1 进入 VB.NET.....6	3.5 表达式.....36
1.5.2 解决方案资源管理器.....7	3.5.1 表达式的组成.....36
1.5.3 属性窗口.....8	3.5.2 优先级.....36
1.5.4 代码设计窗口.....9	3.5.3 类型转换.....37
1.6 VB.NET 帮助系统.....9	3.6 常用系统函数.....39
1.7 习题.....10	3.6.1 命名空间.....39
第2章 Hello World.vb 和基本控件.....11	3.6.2 数学函数.....39
2.1 Hello World.vb.....11	3.6.3 转换函数.....40
2.1.1 “Windows Application” 模板编程.....11	3.6.4 字符串函数.....42
2.1.2 “Console Application” 模板编程.....14	3.6.5 日期函数.....43
2.2 控件及其通用属性.....15	3.7 习题.....44
2.2.1 窗体.....17	第4章 流程控制结构.....46
2.2.2 标签、文本框、命令按钮.....18	4.1 语句.....46
2.2.3 单选按钮、复选框、框架.....22	4.2 顺序结构.....46
2.3 综合实例.....23	4.2.1 赋值语句.....47
2.3.1 算法设计.....23	4.2.2 输入与输出.....48
2.3.2 界面设计.....23	4.3 选择结构.....53
2.3.3 代码设计.....24	4.3.1 If 语句.....54
2.4 习题.....26	4.3.2 Select Case 语句.....58
第3章 VB.NET 语言基础.....28	4.4 循环结构.....60
3.1 数据类型.....28	4.4.1 For...Next 语句.....60
3.1.1 基本数据类型.....28	4.4.2 While 语句.....62
3.1.2 复合数据类型.....30	4.4.3 Do...Loop 语句.....63
3.2 变量.....30	4.4.4 For Each...Next 语句.....66
3.2.1 命名规则.....30	4.5 辅助语句.....66

4.5.1 跳出循环控制.....	66
4.5.2 循环嵌套.....	66
4.5.3 With 语句.....	70
4.6 习题.....	71
第 5 章 数组与集合	74
5.1 数组概述.....	74
5.1.1 数组的引入.....	74
5.1.2 数组的初始化.....	76
5.1.3 Array 类.....	77
5.1.4 动态数组.....	79
5.2 数组操作.....	80
5.2.1 输入与输出.....	80
5.2.2 数组元素查找.....	82
5.2.3 数组元素插入.....	85
5.2.4 数组元素删除.....	87
5.2.5 数组元素排序.....	90
5.3 结构和枚举.....	95
5.3.1 结构类型.....	95
5.3.2 枚举类型.....	97
5.4 集合类.....	99
5.4.1 Collection 集合类.....	99
5.4.2 ArrayList 集合类.....	100
5.4.3 HashTable 集合类.....	101
5.5 习题.....	103
第 6 章 VB.NET 的过程	105
6.1 过程概述.....	105
6.1.1 Function 过程.....	105
6.1.2 Sub 过程.....	107
6.2 实参和形参.....	108
6.2.1 概念.....	108
6.2.2 传递参数的方式.....	109
6.2.3 使用数组作为形参.....	111
6.3 变量的时空性.....	112
6.3.1 变量的作用域.....	112
6.3.2 变量的生存期.....	113
6.4 习题.....	115
第 7 章 算法设计	118
7.1 算法概述.....	118
7.1.1 算法特性.....	118

7.1.2 程序流程图.....	119
7.2 有特点的整数.....	119
7.2.1 最小、最大值.....	119
7.2.2 完全数.....	120
7.2.3 亲密数.....	121
7.2.4 水仙花数.....	122
7.2.5 与素数有关的数.....	123
7.3 经典趣味题.....	130
7.3.1 鸡兔问题.....	130
7.3.2 百钱买百鸡问题.....	131
7.3.3 猴子吃桃问题.....	133
7.4 枚举.....	133
7.5 递归.....	135
7.6 动态规划.....	138
7.7 程序设计方法与风格.....	142
7.7.1 源程序文档化.....	143
7.7.2 语句书写的注意事项.....	144
7.7.3 学习 VB.NET 的建议.....	144
7.8 习题.....	145
第 8 章 用户界面设计	147
8.1 概述.....	147
8.2 控件.....	147
8.2.1 列表框和组合框.....	148
8.2.2 滚动条和进度条.....	156
8.2.3 定时器.....	157
8.2.4 图片箱和图像列表.....	158
8.2.5 驱动器、目录和文件列表框.....	160
8.3 鼠标与键盘.....	163
8.3.1 键盘事件.....	163
8.3.2 鼠标事件.....	164
8.4 通用对话框.....	165
8.5 菜单设计.....	168
8.6 工具栏和状态栏.....	171
8.6.1 工具栏.....	171
8.6.2 状态栏.....	173
8.7 多文档.....	176
8.8 习题.....	177
第 9 章 面向对象程序设计基础	180
9.1 面向对象与面向过程的不同.....	180

9.2 面向对象的基本概念.....	181	11.1.2 结构化查询语言.....	217
9.2.1 对象和类.....	181	11.1.3 客户机/服务器模式.....	219
9.2.2 对象的三大特性.....	183	11.2 数据绑定.....	219
9.3 实例讲解.....	184	11.2.1 简单绑定与复杂绑定.....	219
9.3.1 类和对象.....	184	11.2.2 DataGridView 控件.....	221
9.3.2 对象的属性和方法.....	186	11.3 ADO.NET.....	221
9.3.3 事件.....	190	11.3.1 Connection 对象.....	222
9.3.4 继承性.....	192	11.3.2 Command 对象.....	223
9.3.5 构造函数与析构函数.....	195	11.3.3 DataAdapter 对象.....	224
9.3.6 接口.....	200	11.3.4 DataReader 对象.....	224
9.3.7 多态性.....	202	11.4 DataSet 对象.....	225
9.4 习题.....	204	11.5 DataTable 对象.....	227
第 10 章 文件操作.....	205	11.6 习题.....	228
10.1 文件概述.....	205	第 12 章 异常处理和程序调试.....	229
10.1.1 文件结构.....	205	12.1 错误类型.....	229
10.1.2 文件的分类.....	205	12.1.1 语法错误.....	229
10.1.3 文件访问方式.....	206	12.1.2 运行时错误.....	230
10.2 VB 传统语句访问文件.....	207	12.1.3 逻辑错误.....	230
10.2.1 顺序文件.....	207	12.2 异常处理.....	230
10.2.2 随机文件.....	210	12.2.1 非结构化异常处理.....	230
10.2.3 二进制文件.....	211	12.2.2 结构化异常处理.....	232
10.3 System.IO 模型访问文件.....	212	12.3 调试.....	233
10.3.1 Stream 类.....	212	12.3.1 调试策略.....	234
10.3.2 FileStream 类.....	212	12.3.2 VB.NET 的三种模式.....	234
10.3.3 StreamReader 类与 StreamWriter 类.....	213	12.3.3 常用的调试窗口.....	235
10.4 习题.....	215	12.3.4 三种调试工具.....	236
第 11 章 数据库应用程序开发.....	216	12.4 习题.....	239
11.1 数据库概述.....	216	参考文献.....	240
11.1.1 关系模型.....	216		



第1章 VB.NET 技术基础

本章从计算机的相关基础知识入手,对 VB.NET 编程语言的发展历史、功能、特点以及 VB.NET 的集成开发环境进行了介绍,最后讲述了 VB.NET 的帮助系统。

1.1 计算机基础知识

本节从计算机组成、软件和程序、程序设计语言等方面进行介绍。

1.1.1 计算机组成

计算机由两部分组成:硬件系统和软件系统。所有可视的设备和外围设备都属于硬件系统。1944年,美籍匈牙利数学家冯·诺依曼提出了计算机基本结构和工作方式的设想,为计算机的诞生和发展提供了理论基础。时至今日,尽管计算机软硬件技术飞速发展,但计算机本身的体系结构并没有明显的突破,计算机的体系结构仍属于冯·诺依曼架构。冯·诺依曼提出的理论要点有以下两点:

(1) 计算机硬件设备由存储器、运算器、控制器、输入设备和输出设备五个部分组成。其中,运算器和控制器组成中央处理单元(Center Process Unit, CPU)。中央处理单元用于执行指令,如算数操作、从别的设备写入或读出数据。存储器分为内存和外存。CPU从内存中读取所需要的数据,进行处理。内存中存储的数据是临时的,当程序退出或者计算机关机时,数据将会丢失。如果需要永久的存储数据,需要用到外存,如硬盘、闪存等设备。键盘、鼠标等输入设备用于接受用户输入数据和指令,显示器通常作为输出设备。

(2) 存储程序思想——把计算过程描述为由许多命令按一定顺序组成的程序,然后把程序和数据一起输入计算机,计算机对已存入的程序和数据处理后,输出结果。图 1.1 为计算机体系结构图。

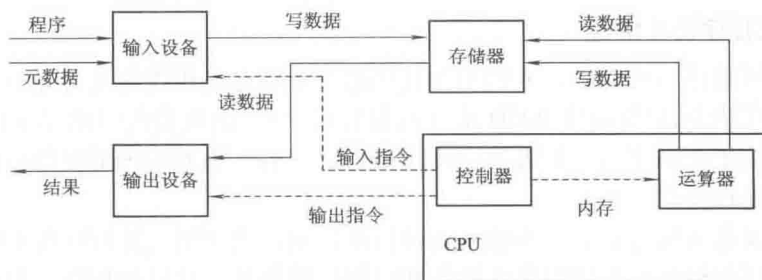


图 1.1 计算机体系结构图



1.1.2 软件和程序

相对于硬件组成而言,软件是由一些“不可视”的部分组成的,它是一系列按照特定顺序组织的计算机数据和指令的集合,如程序、数据、音频、视频等。实际上,不论指令还是数据都以二进制编码形式存在计算机中。在二进制系统中只有 0 和 1 两个数,这是因为计算机硬件组成的物理器件具有两种稳定状态,如门电路的导通与截止、电压的高与低,恰好对应表示 1 和 0 两个数。

计算机软件一般分为系统软件和应用软件两大类。系统软件为计算机用户提供最基本的功能,一般是操作系统和通用平台,如 UNIX、Windows、Linux 和 Android 等,帮助用户管理计算机的硬件。应用软件则是为了特定目的而设计的软件,不同的应用软件根据用户和所服务的领域提供不同的功能,如 Office、PhotoShop、游戏软件等。通常运行时,软件包括以下一些内容:

- (1) 能够提供所要求功能和性能的指令或计算机程序集合。
- (2) 程序能够满意地处理信息的数据结构。
- (3) 描述程序功能需求、如何操作程序和使用所要求的文档。

软件和程序(Program)是两个概念,初学者往往会混淆。其实,这发生在软件发展历史的第一阶段(20 世纪 50 年代初期至 60 年代中期),由于软件的生产个体化,规模较小,功能单一,软件只有程序而无文档,形成了“软件等于程序”的错误观念。而程序是为实现特定目标或解决特定问题用计算机语言编写的命令序列的集合,通过使用与自然语言具有相似的语法和语义的程序设计语言编写源代码,利用特定的工具将其翻译成 CPU 所能执行的指令,达到特定的目的。

1.1.3 程序设计语言

程序设计语言的发展经过了以下几个阶段。

1. 第一代程序设计语言

第一代程序设计语言是机器语言,它是用二进制代码表示的计算机能直接识别和执行的一种机器指令的集合,指令是由 0 和 1 组成的一串代码,通过线路变成电信号,让计算机执行各种不同的操作。机器语言具有直接执行的特点。编程时,编程人员需要熟记所用计算机的全部的由 0 和 1 组成的指令代码的含义。因此,机器语言具有难读、难编、难记和易出错的缺点。

2. 第二代程序设计语言

为了克服机器语言的缺点,人们用与代码指令实际含义相近的英文缩写词、字母和数字等符号来取代指令代码(如用 ADD 表示运算符号“+”的机器代码)作为助记符号编写程序,于是就产生了汇编语言。汇编语言比用机器语言的二进制代码编程要方便些,在一定程度上简化了编程过程。

汇编语言又称为符号语言,不能直接被机器识别,要使用一种程序将汇编语言翻译成机器语言。使用汇编语言来编制系统软件和过程控制软件,其目标程序占用内存空间少,运行速度快,有着高级语言不可替代的用途。



3. 第三代程序设计语言

由于机器语言、汇编语言依赖于硬件体系,要求使用者必须对硬件结构及其工作原理都十分熟悉,因此人们又发明了与人类自然语言相接近且能为计算机所接受的规则明确、通用易学的计算机语言,其语法和结构具有类似文字的表现形式。1954年,第一个面向科学计算的高级计算机语言——FORTRAN 语言被正式推广使用, FORTRAN 语言是 Formula Translation 的缩写,意为“公式翻译”,是数值计算领域所使用的主要语言。1972年,作为程序语言的里程碑, C 语言不但具有高级语言的特点,又具有汇编语言的特点,逐渐成为教学科研和软件开发的主要语言。

计算机并不能直接接收和执行用高级语言编写的源程序,源程序需要“翻译”成机器语言形式的目标程序,计算机才能识别和执行。这种“翻译”通常有两种方式:一种是编译执行;另一种是解释执行。编译执行是指源程序代码先由编译器编译成可执行的机器码,然后再执行;解释执行是指源代码程序被解释器直接读取执行。C、C++ 语言等采用编译执行方式,解释执行的有 Python、Java Script、HTML 等编程语言。编译执行和解释执行各有优缺点,编译执行可一次性将高级语言源程序编译成二进制的可执行指令,通常执行效率高。而解释执行是由该语言(如 HTML)的运行环境(如浏览器)读取一条该语言的源程序,然后转变成二进制指令交给计算机执行,通常可以灵活地跨平台运行。高级程序设计语言具有通用性强、便于移植的特点。

4. 第四代程序设计语言

面向对象程序设计语言、脚本语言、人工智能语言等通常被认为是第四代程序设计语言。Simula 67 是第一个面向对象程序设计语言,特别是 1995 年 5 月由 Sun 公司推出的 Java 程序设计语言,可以撰写跨平台应用软件。第四代程序设计语言提供了功能强大的非过程化定义手段,用户只需告知系统做什么,而无需说明怎么做,大大提高了软件生产效率。VB.NET 就属于第四代程序设计语言。

著名计算机科学家沃思提出了一个公式:程序 = 数据结构 + 算法。其中,数据结构是指数据的类型和数据的组织形式,解决了“如何描述数据?”的问题;而算法是指解决问题的步骤,解决了“如何操作数据?”的问题。计算机编程语言实质就是进行数据处理的工作,解决数据结构和算法的问题。

1.2 VB 发展历史

1964 年,美国的 John G. Kemeny 和 Thomas E. Kurtz 设计了 BASIC 语言, BASIC 是 Beginner's All-purpose Symbolic Instruction Code(初学者通用符号指令代码)的缩写。BASIC 语言自问世以来,其发展经历了以下几个阶段:

(1) 第一阶段(1964—1970 年): 1964 年, BASIC 语言问世。

(2) 第二阶段(1970—1980 年): True BASIC、Quick BASIC 和 Turbo BASIC 等语言时代。

(3) 第三阶段(1980—1990 年): 结构化 BASIC 语言时代。

(4) 第四阶段(1991—2001 年): Visual BASIC(又写为 Visual Basic)语言时代。1991 年 4 月, Microsoft 公司推出 Visual BASIC 1.0, 以可视化工具为界面设计, 以结构化 BASIC 语



言为基础,以事件驱动为运行机制。1998 年的 Visual BASIC 6.0 功能更强大、完善,应用面更广,成为面向对象程序设计语言学习的入门语言,Visual BASIC 衍生语言包括开发桌面应用程序的高级语言(如 VB)、嵌入 Office 系列软件的宏语言(如 VBA)、开发 Web 动态网页的脚本语言(如 VBS)。

(5) 第五阶段(2001 年以来): Visual BASIC.NET 时代。2001 年 Microsoft 将 Visual BASIC 整合到 Microsoft.NET 框架中,形成 Visual BASIC.NET 语言。Visual BASIC.NET 是一种为高效地生成类型安全和面向对象的应用程序而设计的语言,它是基于 Microsoft.NET Framework 之上的面向对象的中间解释性语言,是继 Visual BASIC 6.0 之后的新一代语言。2003 年, Visual BASIC.NET 2003 和 .NET Framework 1.1 发布。

1.3 .NET 框架理念

Microsoft.NET 代表了一个集合、一个环境、一个编程的基本结构。.NET 开发技术包含两部分内容:一方面是应用程序得以运行的框架——.NET Framework 3.0;另一方面是.NET 的开发环境 Visual Studio.NET 平台。其中 .NET 框架是创建、部署和运行新一代基于 Internet 的应用程序的一个统一开发、执行环境,是 .NET 应用程序运行的平台,相当于 Java 的虚拟机,是.NET 程序的基础。

.NET Framework 3.0 架构图如图 1.2 所示,它包括编程语言、公共语言规范(Common Language Specification, CLS)、.NET 开发组件、基础类库(Base Class Library, BCL)及公共语言运行时(Common Language Runtime, CLR)等。其中,公共语言运行时(CLR)运行在操作系统上,负责管理内存分配、启动或删除线程和进程、实施安全性策略等。基础类库(BCL)为开发人员提供了一个统一、面向对象、层次化、可扩展的类库集,在公共语言运行时的基础上运行。

从 Java 语言开始,一种新的程序语言执行方式产生了,这就是“中间码+虚拟机”执行机制。Java 源代码被编译成一种特殊的中间码,而这种中间码是不能直接被执行的,需要“虚拟机”管理和执行,既可以解释执行也可以编译执行。由于“虚拟机”参与和管理程序代码的执行,因此解决了垃圾内存回收、安全性检查等传统编译语言无法解决的问题。.NET 框架也采用了这种模式,Java 中的管理和执行中间码“虚拟机”在.NET 框架中就是公共语言运行时(CLR),由其负责管理和执行由.NET 编译器编译产生的中间语言代码。图 1.3 给出了 .NET 程序的执行原理。在开发环境 Visual Studio .NET 中编写 VB.NET 代码,然后编译成中间语言代码,最后由操作系统中 .NET Framework 的组件 CLR 管理和执行。

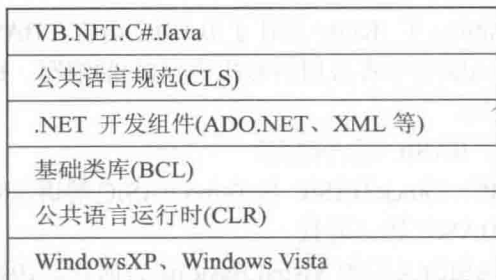


图 1.2 .NET Framework 3.0 架构图

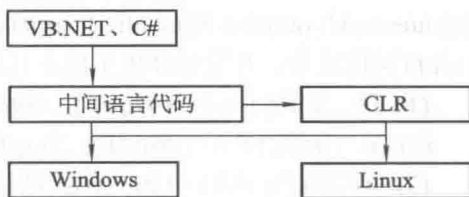


图 1.3 .NET 程序的执行原理



中间语言原理图如图 1.4 所示。汉语、日语、法语等国家的学者交流，他们就需要通过中间语言——英语来交流。汉语、日语、法语就像是 .NET 平台上的各种编程语言 C#、VB.NET 等，而英语相当于 .NET 框架中的中间语言，.NET 框架会将 C# 或者 VB.NET 等编程语言翻译成相同的中间语言代码，从而使得 C# 可以被 VB.NET 调用，同样 VB.NET 也可以被 C# 等其他语言调用。

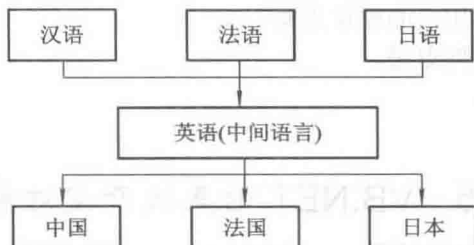


图 1.4 中间语言原理图

1.4 VB.NET 的功能及特点

VB.NET 在 BASIC 语言的基础上发展而来，吸收了 BASIC 语言的优点，并且基于 .NET 框架，具有以下强大的功能特点：

(1) 提供了易学易用的应用程序集成开发环境。VB.NET 不但语法比较简单，容易掌握，而且 VB.NET 集成开发环境集创建工程、设计界面、编辑代码、调试程序、直接运行及生成可执行文件等于一体，使用起来极为方便。

(2) 结构化程序设计。VB.NET 具有高级程序设计语言的优点：丰富的数据类型、大量的内部函数、三种流程控制结构和模块化的程序结构。

(3) 可视化编程技术。可视化编程技术给人一种所见即所得的感觉，VB.NET 提供“工具箱”，不编程或者使用很少的代码即可实现程序的界面设计，极为方便。如设计 Windows 操作系统自带的计算器软件，如图 1.5 所示。



图 1.5 计算器软件



(4) 面向对象的事件驱动编程机制。在面向对象程序设计中,窗口、命令按钮、文本框等都是一个对象,当按下命令按钮或者在文本框中输入信息都会触发相应的事件。在 VB.NET 中,代码是针对对象的事件进行编写。

事件驱动具有以下一些优点:

- ① 可以为用户提供即时反馈。
- ② 使程序设计更贴近用户的操作需要。
- ③ 使程序设计的目的性更强。
- ④ 减少程序的复杂性。

1.5 VB.NET 的集成开发环境

Visual Studio.NET 支持 C#、VB.NET 等编程语言的程序开发,它是 Microsoft.NET 框架中最重要的开发工具。Visual Basic .NET 的集成开发环境(IDE)是 Visual Studio.NET 的一部分,它集程序代码的编辑、编译、连接和调试等功能于一体,给编程人员提供了一个完整、方便的开发界面和众多有效的开发辅助工具。

1.5.1 进入 VB.NET

本节以 VB.NET 2005 的 IDE 为开发环境介绍 VB.NET 程序设计。VB.NET 的集成开发环境如图 1.6 所示。

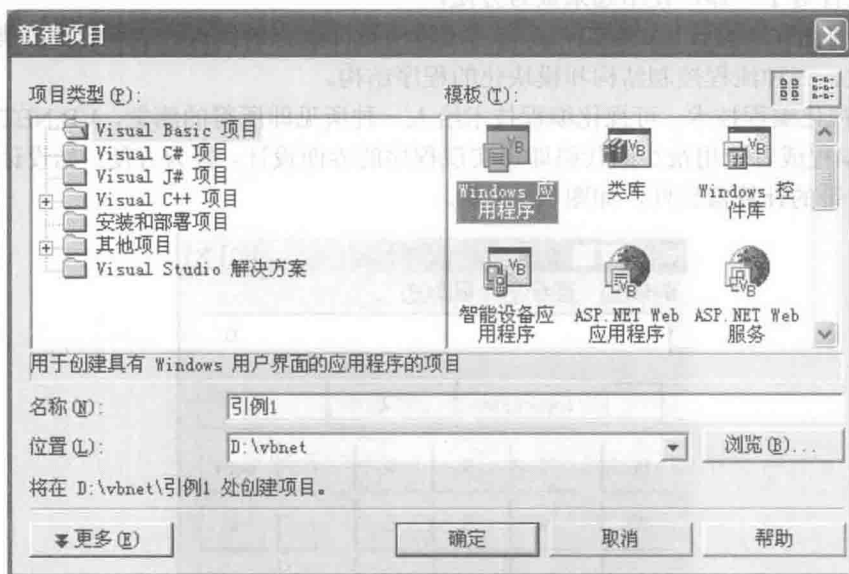


图 1.6 VB.NET 的集成开发环境

选择 VB.NET 的“Windows Application”模板,如图 1.7 所示,包括菜单栏、工具栏、选项卡组栏、工具箱、代码设计窗口、输出窗口、解决方案资源管理器窗口、属性窗口等。

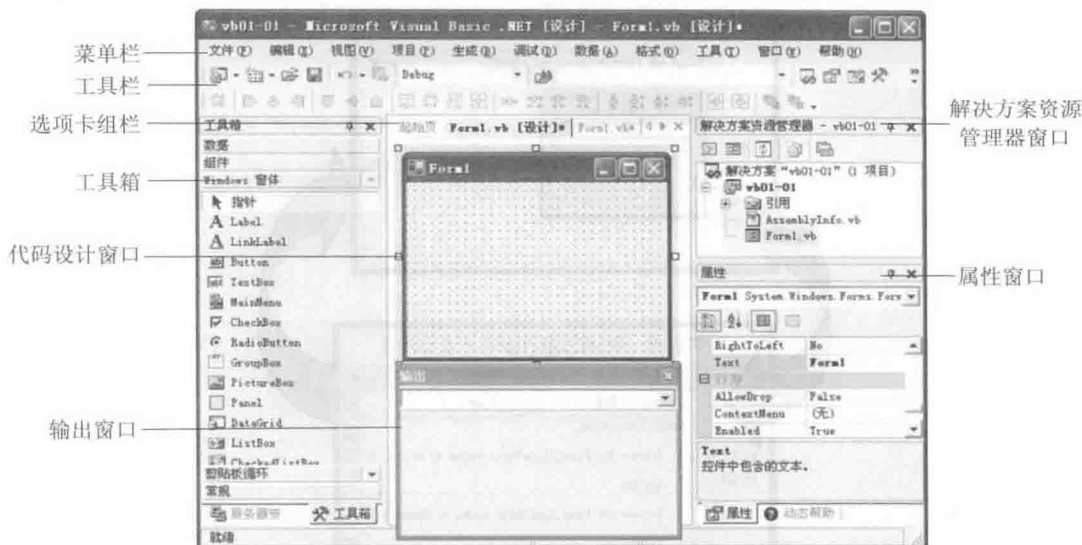


图 1.7 VB.NET 的“Windows Application”模板

1.5.2 解决方案资源管理器

解决方案资源管理器用于显示 VB.NET 程序所有相关的文件，如图 1.8 所示。

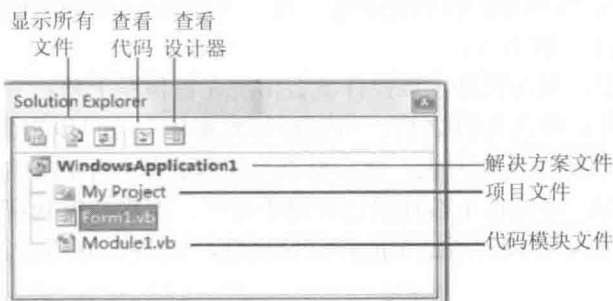


图 1.8 解决方案资源管理器窗口

VB.NET 的解决方案可以含有以下类型的文件：

- (1) 解决方案文件(.sln)：可包括一个或多个用不同语言开发的项目。
- (2) 项目文件(.vbproj)：每个项目对应一个项目文件，由引用的组件和代码模块组成。
- (3) 代码模块文件(.vb)：包括窗体文件、类模块或其他代码文件。图 1.8 中的两种文件

描述如下：

- ① Form1.vb 是窗体文件，运行时该文件存储窗体上使用的所有控件对象和有关的属性、对象相应的事件过程、程序代码。一个 Windows 应用窗体至少包含一个窗体文件。
- ② Module1.vb 是标准模块文件，存放所有公共变量和用户自定义的通用过程。通用过程是指可以被应用程序调用的过程。

在解决方案资源管理器中，代码设计和界面设计这两种不同的窗口之间可以进行切换，如图 1.9 所示。

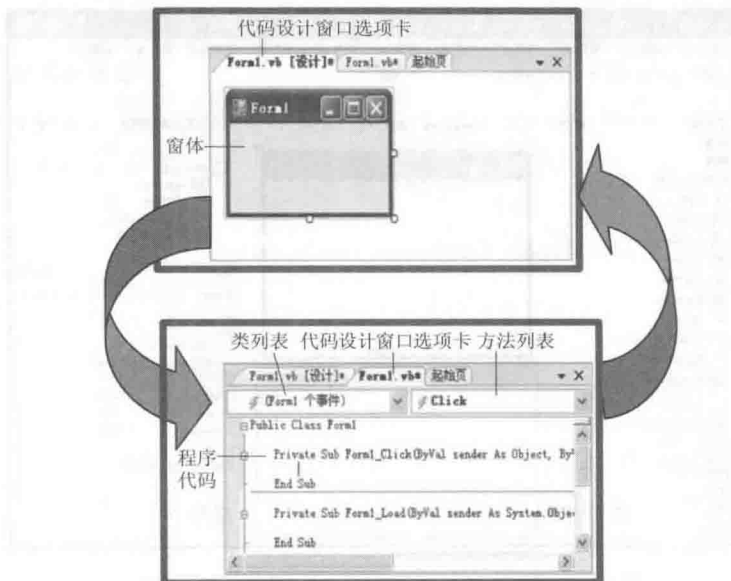


图 1.9 窗口的切换

1.5.3 属性窗口

在 VB.NET 中，窗体和控件都是对象，用一组属性来描述其外部特征，如颜色、大小等。添加控件有以下三种方法：

- (1) 在工具箱中，双击控件，该控件就会出现在窗体左上角。
- (2) 在工具箱中，单击控件之后，将鼠标移到窗体上，可以看到鼠标变成带小加号的形状，在合适的位置单击鼠标左键，该控件就会出现在鼠标单击的位置。
- (3) 在工具箱中，在控件上按住鼠标左键不放开，直接拖曳到窗体合适的位置。

属性窗口用于显示和设置所选定的控件的属性，如图 1.10 所示。



图 1.10 属性窗口



1.5.4 代码设计窗口

代码设计包括各种事件过程、过程和类等源程序代码的编写和修改, 代码设计窗口如图 1.11 所示。



图 1.11 代码设计窗口

打开代码设计窗口的方法有以下三种:

- (1) 在窗体或者窗体文件上单击鼠标右键, 在打开的右键菜单中单击“查看代码”选项。
- (2) 执行菜单栏中的“视图→代码”命令, 切换到代码设计窗口。
- (3) 直接双击所需编程的控件, 便显示此控件的代码设计窗口。

1.6 VB.NET 帮助系统

VB.NET 帮助文档又称为 MSDN(Microsoft Developer Network)Library, 是 Visual Basic 开发人员的重要参考资料。MSDN Library 包括示例代码、文档、技术文章、Microsoft 开发人员知识库以及其他 Microsoft 技术支持资料等。

- (1) 使用方式 1: 直接在帮助窗口查找“Now 函数”, 如图 1.12 所示。

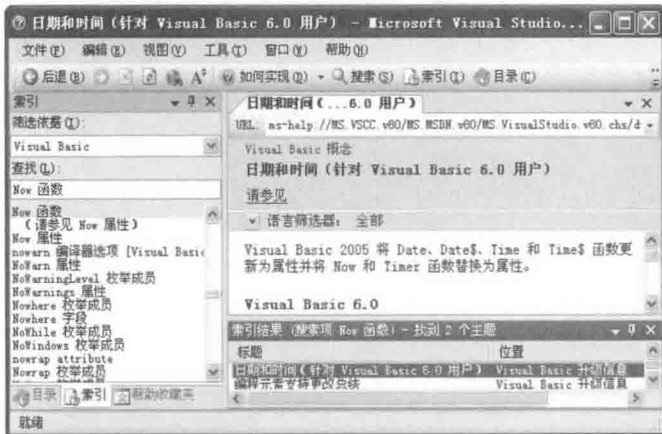


图 1.12 直接在帮助窗口查找“Now 函数”



(2) 使用方式 2: 选中要了解的对象, 按“F1”键, “Integer 数据类型”的帮助界面如图 1.13 所示。

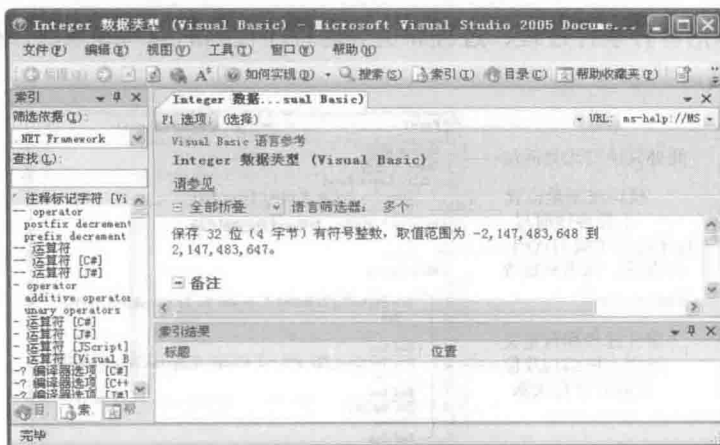


图 1.13 “Integer 数据类型”的帮助界面

1.7 习 题

1. 冯·诺依曼理论是什么?
2. 软件和程序是否一样?
3. 程序设计语言经过了哪些阶段?
4. 什么是 .NET 框架?
5. 简述 VB.NET 的功能和特点。
6. VB 经历了哪些历史发展阶段?
7. 熟悉集成开发环境中各主要组件的使用方法, 包括菜单栏、工具栏、工具箱、属性窗口、代码设计窗口、解决方案资源管理器窗口和帮助系统。
8. 安装 MSDN, 学习如何使用它。

