

# Visual Basic.NET程序设计

Visual Basic.NET Programming

周霭如 林伟健 解晓萌 编著



高等教育出版社  
HIGHER EDUCATION PRESS

# Visual Basic.NET 程序设计

---

Visual Basic.NET Chengxu Sheji

周霭如 林伟健 解晓萌 编著



高等教育出版社·北京  
HIGHER EDUCATION PRESS BEIJING

内容提要

本书从高级语言程序设计的基本规律出发,介绍了 Visual Basic.NET 语言的基本概念和编程技术。本书共分为 11 章。第 1 章至第 6 章以控制台程序介绍了面向过程和面向对象程序设计的基本原理和方法。第 7 章至第 10 章介绍了应用.NET 框架的界面设计技术、文件应用和访问数据库技术。第 11 章通过两个综合用例,详细介绍了 Visual Basic.NET 应用程序开发的思路、过程和技术。

本书概念清晰、循序渐进、用例丰富、便于理解,适合作为高等学校程序设计语言的教材,也可作为 Visual Basic.NET 程序设计爱好者的参考材料。

图书在版编目(CIP)数据

Visual Basic. NET 程序设计/周靛如,林伟健,解晓萌  
编著. —北京:高等教育出版社,2013.2  
ISBN 978-7-04-036860-4

I. ①V… II. ①周… ②林… ③解… III. ①BASIC  
语言-程序设计-高等学校-教材 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2013) 第 013897 号

|          |          |          |          |
|----------|----------|----------|----------|
| 策划编辑 饶卉萍 | 责任编辑 张海波 | 封面设计 于文燕 | 版式设计 余 杨 |
| 责任校对 刘 莉 | 责任印制 赵义民 |          |          |

|      |                   |      |                                                                       |
|------|-------------------|------|-----------------------------------------------------------------------|
| 出版发行 | 高等教育出版社           | 咨询电话 | 400-810-0598                                                          |
| 社 址  | 北京市西城区德外大街 4 号    | 网 址  | <a href="http://www.hep.edu.cn">http://www.hep.edu.cn</a>             |
| 邮政编码 | 100120            |      | <a href="http://www.hep.com.cn">http://www.hep.com.cn</a>             |
| 印 刷  | 北京鑫海金澳胶印有限公司      | 网上订购 | <a href="http://www.landracom.com">http://www.landracom.com</a>       |
| 开 本  | 787mm×1092mm 1/16 |      | <a href="http://www.landracom.com.cn">http://www.landracom.com.cn</a> |
| 印 张  | 23.75             | 版 次  | 2013 年 2 月第 1 版                                                       |
| 字 数  | 580 千字            | 印 次  | 2013 年 2 月第 1 次印刷                                                     |
| 购书热线 | 010-58581118      | 定 价  | 37.00 元                                                               |

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换。

版权所有 侵权必究

物 料 号 36860-00

# 前 言

Visual Basic.NET (VB.NET) 是 Microsoft 公司推出的 Visual Basic 语言的新一代产品, 它保留了 BASIC 语言易学易懂的特点, 并且具备了面向对象编程语言的所有特征, 包括类、接口、封装、继承和多态性等, 还具有结构化的异常处理和多线程功能, 使其成为一种强大的面向对象的程序设计语言。同时, Visual Basic.NET 基于 .NET Framework, 在强大的类库支持下, 提高了综合安全性, 有效地进行内存管理, 能够便捷地开发各种用途的应用程序。

本书注重两个特点: 一是基础性, 二是实用性。所谓基础性, 要求在难度适当的前提下, 讲清基本概念和基本方法, 注重学生的能力培养。实用性体现在教学内容取材的先进性, 缩短学生从课堂到应用的过程。本书的教学目标是, 通过学习 Visual Basic.NET 语言, 让学生掌握计算机程序设计的基本规律, 掌握开发一个小型应用程序的方法, 并具备进一步学习的能力。

考虑到程序设计课程的教学特点, 本书第 1 章到第 6 章使用控制台程序例程, 介绍 VB.NET 的基本语法, 使得读者注意力集中于程序设计的逻辑思维。在之后的章节中, 逐步展开 Windows 应用程序的界面设计技术, 以及文件、数据库的应用。第 11 章通过两个综合应用实例, 引导学生掌握开发一个简单应用程序的方法。

全书共分 11 章。主要内容包括: 高级程序设计语言的基本概念, 基本数据类型与表达式, 程序流程控制, 复合数据类型, 过程, 面向对象程序设计基础, 常用控件, 文件, 复杂界面设计, 数据库应用, 综合应用实例。

本书是一本高级语言程序设计的基础教材, 所以没有涵盖 VB 的全部的功能。相信有兴趣的读者具备程序设计的基本知识后, 可以根据不同需要参阅相关资料学习。

本书是在华南理工大学非计算机专业多年开设程序设计课程的基础上编写的, 任课教师和学生对此提出过许多有益的建议, 在此表示感谢。限于时间和编者的水平, 书中不妥之处难免, 敬请批评指正。

编者

2012 年 9 月

# 目 录

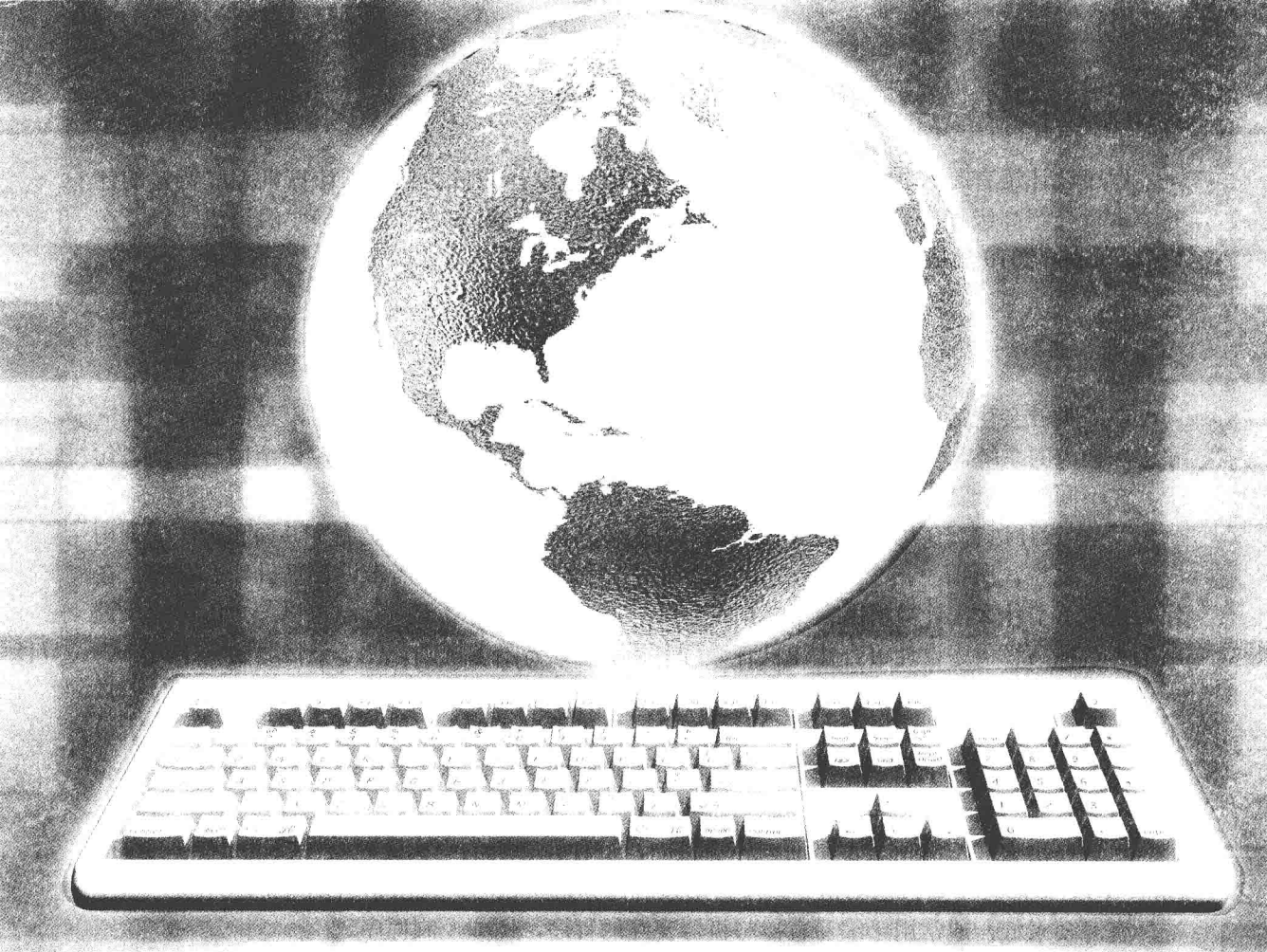
|                                            |    |                                  |     |
|--------------------------------------------|----|----------------------------------|-----|
| 第1章 概述 .....                               | 1  | 2.5.3 字符串函数 .....                | 37  |
| 1.1 程序设计语言 .....                           | 2  | 2.5.4 日期/时间函数 .....              | 40  |
| 1.2 程序设计 .....                             | 3  | 2.5.5 转换函数 .....                 | 41  |
| 1.3 Visual Studio 2010 集成开发环境 .....        | 5  | 2.6 控制台数据的输入和输出 .....            | 43  |
| 1.3.1 主窗口 .....                            | 5  | 2.6.1 键盘输入 .....                 | 43  |
| 1.3.2 菜单栏 .....                            | 6  | 2.6.2 控制台输出 .....                | 43  |
| 1.3.3 工具栏 .....                            | 7  | 习题 2 .....                       | 47  |
| 1.3.4 项目、解决方案和项目<br>工作区 .....              | 8  | 第3章 程序流程控制 .....                 | 50  |
| 1.3.5 Visual Studio 2010 帮助<br>系统的使用 ..... | 8  | 3.1 分支结构 .....                   | 51  |
| 1.4 建立控制台应用程序 .....                        | 9  | 3.1.1 单行结构条件语句 .....             | 51  |
| 习题 1 .....                                 | 15 | 3.1.2 块结构条件语句 .....              | 53  |
| 第2章 基本数据类型与表达式 .....                       | 16 | 3.1.3 GoTo 语句 .....              | 57  |
| 2.1 基本字符集和词汇集 .....                        | 17 | 3.1.4 情况语句 .....                 | 58  |
| 2.1.1 字符集 .....                            | 17 | 3.2 循环结构 .....                   | 61  |
| 2.1.2 词汇 .....                             | 17 | 3.2.1 循环概念 .....                 | 61  |
| 2.2 基本数据类型 .....                           | 19 | 3.2.2 While...End While 语句 ..... | 62  |
| 2.2.1 数值类型 .....                           | 20 | 3.2.3 Do...Loop 语句 .....         | 64  |
| 2.2.2 字符类型 .....                           | 22 | 3.2.4 For...Next 语句 .....        | 67  |
| 2.2.3 其他数据类型 .....                         | 23 | 3.2.5 嵌套循环与退出循环 .....            | 70  |
| 2.3 常量与变量 .....                            | 23 | 3.3 暂停和结束语句 .....                | 75  |
| 2.3.1 常量 .....                             | 24 | 习题 3 .....                       | 76  |
| 2.3.2 变量 .....                             | 25 | 第4章 复合数据类型 .....                 | 84  |
| 2.4 表达式 .....                              | 27 | 4.1 数组 .....                     | 85  |
| 2.4.1 算术表达式 .....                          | 28 | 4.1.1 一维数组及其使用 .....             | 85  |
| 2.4.2 字符串表达式 .....                         | 30 | 4.1.2 二维数组及其使用 .....             | 90  |
| 2.4.3 逻辑表达式 .....                          | 30 | 4.1.3 For Each...Next 语句 .....   | 96  |
| 2.4.4 表达式的运算顺序 .....                       | 33 | 4.1.4 数组的整体赋值 .....              | 97  |
| 2.5 常用内部函数 .....                           | 33 | 4.1.5 数组的重定义 .....               | 97  |
| 2.5.1 数学函数 .....                           | 34 | 4.2 结构 .....                     | 100 |
| 2.5.2 生成随机数函数 .....                        | 36 | 4.2.1 结构类型和结构变量说明 .....          | 100 |
|                                            |    | 4.2.2 结构变量数据成员的引用 .....          | 101 |
|                                            |    | 4.2.3 结构变量的输入与输出 .....           | 102 |

|                                   |     |                             |     |
|-----------------------------------|-----|-----------------------------|-----|
| 4.2.4 结构数组 .....                  | 104 | 6.2.1 类的定义 .....            | 149 |
| 4.3 集合 .....                      | 106 | 6.2.2 对象的定义与访问 .....        | 151 |
| 4.3.1 集合对象的建立 .....               | 106 | 6.2.3 属性的定义及其使用 .....       | 153 |
| 4.3.2 向集合添加数据 .....               | 106 | 6.2.4 事件的定义及其触发 .....       | 154 |
| 4.3.3 集合中数据的引用 .....              | 107 | 6.2.5 方法的定义及其调用 .....       | 157 |
| 4.3.4 Contains 方法 .....           | 107 | 6.3 构造函数和析构函数 .....         | 158 |
| 4.3.5 删除集合中的数据 .....              | 108 | 6.3.1 构造函数 .....            | 158 |
| 4.4 枚举 .....                      | 110 | 6.3.2 析构函数 .....            | 159 |
| 4.4.1 枚举类型说明 .....                | 110 | 6.4 继承与派生 .....             | 159 |
| 4.4.2 枚举变量说明 .....                | 111 | 6.4.1 基类与派生类的概念 .....       | 159 |
| 4.4.3 枚举变量输入/输出 .....             | 111 | 6.4.2 派生类的定义 .....          | 160 |
| 习题 4 .....                        | 112 | 6.4.3 派生类的构造函数 .....        | 161 |
| 第 5 章 过程 .....                    | 118 | 6.5 类的多态性 .....             | 162 |
| 5.1 VB.NET 的项目 .....              | 119 | 6.5.1 重写 .....              | 162 |
| 5.2 过程的定义与调用 .....                | 120 | 6.5.2 重载 .....              | 163 |
| 5.2.1 Sub 过程的定义与调用 .....          | 121 | 6.6 命名空间 .....              | 165 |
| 5.2.2 Function 过程的<br>定义和调用 ..... | 123 | 6.6.1 命名空间的概念 .....         | 165 |
| 5.3 参数传递 .....                    | 125 | 6.6.2 命名空间的使用 .....         | 166 |
| 5.3.1 形式参数和实际参数 .....             | 125 | 6.6.3 自定义命名空间 .....         | 166 |
| 5.3.2 传值参数 .....                  | 125 | 习题 6 .....                  | 168 |
| 5.3.3 可选参数 .....                  | 126 | 第 7 章 常用控件 .....            | 172 |
| 5.3.4 引用参数 .....                  | 127 | 7.1 Windows 程序开发环境 .....    | 173 |
| 5.3.5 关于参数的讨论 .....               | 128 | 7.1.1 开发环境 .....            | 173 |
| 5.4 数组、结构和集合参数 .....              | 130 | 7.1.2 建立 Windows 应用程序 ..... | 178 |
| 5.4.1 数组参数 .....                  | 130 | 7.2 控件概述 .....              | 180 |
| 5.4.2 结构参数 .....                  | 133 | 7.3 窗体和消息框 .....            | 183 |
| 5.4.3 集合参数 .....                  | 134 | 7.3.1 窗体 .....              | 183 |
| 5.5 过程与变量的作用域 .....               | 135 | 7.3.2 输入框和消息框 .....         | 186 |
| 5.5.1 过程的作用域 .....                | 135 | 7.4 文本控件 .....              | 190 |
| 5.5.2 变量作用域与生存周期 .....            | 136 | 7.4.1 标签 .....              | 190 |
| 5.6 递归 .....                      | 139 | 7.4.2 网页浏览控件 .....          | 191 |
| 习题 5 .....                        | 142 | 7.4.3 文本框 .....             | 192 |
| 第 6 章 面向对象程序设计基础 .....            | 147 | 7.4.4 关联文本 .....            | 197 |
| 6.1 基本概念 .....                    | 148 | 7.5 图片框 .....               | 202 |
| 6.1.1 类和对象 .....                  | 148 | 7.5.1 图片框 .....             | 202 |
| 6.1.2 属性、事件和方法 .....              | 148 | 7.5.2 图片列表 .....            | 203 |
| 6.2 类和对象的定义 .....                 | 149 | 7.6 键盘和鼠标事件 .....           | 205 |
|                                   |     | 7.6.1 焦点和 Tab 顺序 .....      | 205 |

|                                         |     |                            |     |
|-----------------------------------------|-----|----------------------------|-----|
| 7.6.2 键盘事件 .....                        | 206 | 8.6.2 保存文件对话框 .....        | 254 |
| 7.6.3 鼠标事件 .....                        | 207 | 习题 8 .....                 | 256 |
| 7.7 按钮 .....                            | 209 | <b>第 9 章 复杂界面</b> .....    | 258 |
| 7.7.1 命令按钮 .....                        | 209 | 9.1 菜单 .....               | 259 |
| 7.7.2 复选框 .....                         | 210 | 9.1.1 菜单的形式 .....          | 259 |
| 7.7.3 单选按钮 .....                        | 212 | 9.1.2 菜单栏 .....            | 260 |
| 7.8 颜色和字体对话框 .....                      | 213 | 9.1.3 快捷菜单 .....           | 264 |
| 7.8.1 颜色对话框 .....                       | 213 | 9.2 工具栏 .....              | 265 |
| 7.8.2 字体对话框 .....                       | 215 | 9.3 状态栏 .....              | 267 |
| 7.9 列表框 .....                           | 217 | 9.4 视图 .....               | 269 |
| 7.9.1 标准列表框 .....                       | 217 | 9.4.1 树视图 .....            | 270 |
| 7.9.2 复选列表框 .....                       | 221 | 9.4.2 列表视图 .....           | 274 |
| 7.9.3 组合框 .....                         | 223 | 9.5 多窗体 .....              | 279 |
| 7.10 容器 .....                           | 225 | 9.5.1 多重窗体界面 .....         | 279 |
| 7.10.1 分组框 .....                        | 226 | 9.5.2 多文档界面 .....          | 282 |
| 7.10.2 其他容器 .....                       | 227 | 9.5.3 窗体之间的数据传递 .....      | 287 |
| 7.11 滚动条 .....                          | 230 | 9.5.4 启动窗口 .....           | 291 |
| 7.12 计时器 .....                          | 231 | 习题 9 .....                 | 292 |
| 习题 7 .....                              | 232 | <b>第 10 章 数据库应用</b> .....  | 294 |
| <b>第 8 章 文件</b> .....                   | 236 | 10.1 数据库的概念 .....          | 295 |
| 8.1 文件概述 .....                          | 237 | 10.2 建立数据库 .....           | 296 |
| 8.1.1 文件的概念及其命名 .....                   | 237 | 10.3 SQL 语言简介 .....        | 299 |
| 8.1.2 文件的分类 .....                       | 237 | 10.3.1 SQL 的语句 .....       | 299 |
| 8.1.3 文件操作的基本步骤 .....                   | 239 | 10.3.2 SQL 中的数据操作 .....    | 300 |
| 8.1.4 读/写文件的 3 种方法 .....                | 239 | 10.4 VB.NET 中的数据库操作 .....  | 303 |
| 8.2 System.IO 命名空间简介 .....              | 240 | 10.4.1 连接数据库 .....         | 304 |
| 8.2.1 流的概念及其操作 .....                    | 240 | 10.4.2 数据库设计工具 .....       | 307 |
| 8.2.2 System.IO 命名空间中用于<br>文件操作的类 ..... | 240 | 10.4.3 显示数据表 .....         | 309 |
| 8.3 文件流类 FileStream .....               | 240 | 10.4.4 查询数据库 .....         | 315 |
| 8.4 读/写文本文件 .....                       | 242 | 习题 10 .....                | 324 |
| 8.4.1 StreamWriter 类 .....              | 242 | <b>第 11 章 综合应用实例</b> ..... | 326 |
| 8.4.2 StreamReader 类 .....              | 244 | 11.1 计算器实例 .....           | 327 |
| 8.5 读/写二进制文件 .....                      | 246 | 11.1.1 项目需求分析 .....        | 327 |
| 8.5.1 BinaryWriter 类 .....              | 247 | 11.1.2 计算器用户界面设计 .....     | 327 |
| 8.5.2 BinaryReader 类 .....              | 248 | 11.1.3 程序实现 .....          | 329 |
| 8.6 文件对话框 .....                         | 251 | 11.2 个人开销管理系统实例 .....      | 337 |
| 8.6.1 打开文件对话框 .....                     | 251 | 11.2.1 数据库设计 .....         | 337 |
|                                         |     | 11.2.2 创建项目 .....          | 340 |

|                          |     |                       |     |
|--------------------------|-----|-----------------------|-----|
| 11.2.3 个人开销管理系统的实现 ..... | 345 | A.3 异常处理 .....        | 364 |
| 附录 A 程序调试 .....          | 359 | A.3.1 结构化异常处理 .....   | 364 |
| A.1 调试工具 .....           | 360 | A.3.2 非结构化异常处理 .....  | 366 |
| A.2 程序调试练习 .....         | 361 | 附录 B ASCII 码字符集 ..... | 368 |
|                          |     | 附录 C 文件后缀说明 .....     | 370 |





# 第 1 章

## 概 述

程序设计语言是人指挥计算机工作的工具，程序设计的目的是要求计算机完成特定的任务。本章介绍高级程序设计语言的基本概念，简单的 Visual Basic 语言程序的构成，以及在 Visual Studio 2010 集成开发环境下创建控制台应用程序的基本方法。

## 1.1 程序设计语言

在人类社会生活中，“语言”是人与人之间用来表达、交流思想的工具，是由语音、词汇和语法构成的系统。人类的思维、感情相当丰富，所以语言系统非常复杂，以至同一个词、句子，在不同的场景以不同的语气表达，都可能解释成完全不同的意思。

“程序设计语言”（也称算法语言）是人指挥计算机工作的工具，是由字、词和语法规则构成的指令系统。所谓“指令”，就是命令。使用程序设计语言，只需告诉计算机什么时候、在什么条件下干什么，计算机就会根据指令一条一条地执行，并且把执行结果以一定的形式反馈出来。一种高级程序设计语言往往只有一百多条词汇、若干条规则，便于记忆、易于学习。

最早期的计算机语言只有二进制语言。这种语言的代码全部由二进制符号“0”和“1”按不同的方式排列组合而成，编写的程序能够被计算机直接识别执行。二进制代码记忆难度大，容易出错。随后出现了由一些助记符表示二进制代码的语言，称之为汇编语言，或符号语言。用汇编语言编写的程序只需要经过简单的翻译就可以被计算机执行。

二进制语言和汇编语言都是面向计算机的，每种计算机的中央处理器（Central Processing Unit, CPU）都有一套自己的指令系统，所以称之为低级语言。低级语言编写的程序执行速度快，资源要求低，通常用于编写直接与机器交互的程序，例如控制程序。

人们要求计算机处理的问题越来越复杂，求解方式通常可用数学模型进行抽象。20 世纪 60 年代出现了接近于人类语言描述方式构成的指令系统，称之为“高级语言”。高级语言提供常用的数据描述和对数据操作的规则描述。这些规则是“脱机”的，程序员只需要专注于问题的求解，不必关心机器内部的结构和实现细节。

人们所说的“程序设计语言”通常是指高级语言。用高级语言编写的程序称为“源程序”。源程序代码不能被计算机直接识别，必须编译成二进制程序才能在机器上运行。编译后的程序称为“目标程序”。一旦编译成功，目标程序就可以反复高速执行。每种高级语言都配有解释或编译系统。

计算机应用的领域很广泛。为了适应不同的需要，往往程序设计语言又各具特点。例如，有适合于编写系统软件的，有的适于进行科学计算的，或用于数据库管理、图形设计及人工智能等。还有一些语言具备多种功能。从应用角度说，难以对程序设计语言做严格分类，而且随着计算机科学的发展及应用领域的迅速扩展，各种语言的版本仍在不断变化，功能也在不断更新和增强。每个时期都有一批程序设计语言流行，又有一批程序设计语言消亡，因此应该掌握程序设计语言中本质性的、规律性的知识。

为了满足人们各种特定的需求，计算机工作者开发了各种工具软件，例如文字处理、表格处理、图形处理、多媒体管理以及各种系统管理工具等。它们是一些专用程序集。用户在界面上与计算机交互操作，就是调用其中的各个子程序。

然而，世界总是复杂且千变万化的，人们要求计算机做的工作也千差万别。可以说，现代

社会对计算机的要求是无止境的。计算机厂家不断生产功能越来越强大的计算机系统，软件开发者也在不断设计满足不同需要的应用程序。不管计算机结构如何，人们总是用某种形式的“语言”与计算机交流，使计算机完成特定的处理。

Visual Basic 是从 BASIC (Beginner's All-purpose Symbolic Instruction Code, 初学者通用符号指令代码) 语言发展而来的。BASIC 语言是 20 世纪 60 年代中期由 Dartmouth 大学的教授 John Kemeny 和 Thomas Kurtz 开发的，主要用于编写简单的程序。

BASIC 语言简单易用，各种类型计算机上都可以实现。随着应用范围的扩展，BASIC 语言也在不断改进。20 世纪 80 年代末 90 年代初，微软公司开发出图形用户界面 (Graphical User Interface, GUI)，并于 1991 年推出 Visual Basic，使得 Windows 应用程序开发变得极为方便。

20 世纪 90 年代后期，编程工具和消费型电子设备的发展对程序设计语言提出了很多新的挑战。例如，系统预定义组件的集成，基于 Web 应用程序的开发，移动电子设备通过 Internet 与软件组件交互等。为了满足这些新需求，微软公司在 2000 年公布了 .NET 策略，提供了 Microsoft Visual Studio (VS.NET) 平台。它是一套完整的开发工具，Web 应用程序可以通过这个平台分布到各种设备 (如 PDA 和手机) 及桌面计算机上。利用 VS.NET，可以生成 ASP Web 应用程序、XML Web Services、桌面应用程序和移动应用程序。在 VS.NET 中，Visual Basic.NET、Visual C#.NET 和 Visual J#.NET 都使用相同的集成开发环境 (Integrated Development Environment, IDE)，允许由不同语言开发的项目共享这些工具，创建混合语言的解决方案。.NET 自 2000 年已经推出了多个版本。

本书是学习程序设计的入门教材。学习 Visual Basic，不但是为了掌握一种实用的计算机软件设计工具，而且通过本课程的学习，应该对计算机的工作原理和工作方式有初步了解，掌握程序设计语言的基本语法规则，掌握可视化程序设计的基本方法，为进一步学习和应用打下良好基础。

本书以 Visual Basic.NET 2010 (以下简称 VB.NET) 作为讨论蓝本。

## 1.2 程 序 设 计

程序是针对特定的问题，使用某种程序设计语言设计的计算机指令序列。在 .NET 框架中提供了各种不同类型项目的模板。从界面风格上，有用于创建命令行应用程序的控制台应用程序模板，还有用于创建具有 Windows 用户界面应用程序的 Windows 窗体应用程序模板。

控制台应用程序适用于开发不需要可视化图形界面、对存储空间要求比较低且要求运行速度比较快的系统。控制台程序代码简洁，是语言入门的一种便捷的方法。

Windows 应用程序基于图形界面，易于操作，是当今计算机应用程序的主要风格。若由程序员直接开发这种可视化应用程序，无疑工作量十分浩大。.NET 框架提供了一套标准的工具，包括窗体、按钮、菜单、文本框和图片框等，程序员根据程序的功能和与用户信息交互的需要，可以方便地规划界面的布局和设计不同的界面风格。

不管是设计控制台应用程序还是 Windows 应用程序，都需要把问题模型转换成为计算机代码可以表示的数据和对数据的处理形式。

## 1. 数据表示

数据表示是把被处理的信息描述成计算机可以接受的数据形式，如整型、实型、字符、数组等。

信息是对现实世界的抽象反映。现实世界中充满信息，人们每时每刻接触到各种各样的信息，而且频繁地传播、加工和利用这些信息。

信息可用人工或自动装置进行记录、解释和处理。使用计算机进行信息处理时，这些信息必须转换成可被机器识别的“数据”，如数值、文字、图形、声音等。不管什么数据，计算机都以二进制形式存储和加工处理。数据是信息的载体，信息依靠数据来表达。

有些数据，程序设计语言可以直接用系统预定义的“数据类型”来描述，例如数值和字符等。另外一些数据，程序设计语言没有提供直接定义，可以由程序组织或者用特定的方法使用外部数据。VB.NET 可以通过控件使用数据库、图形、声音等不同格式的文件。

## 2. 数据处理

数据处理是指对数据进行输入、输出、整理、计算、存储、维护等一系列活动。数据处理的目的是提取所需的数据成分，获得有用的资料。

通常用子程序实现对数据的特定操作。数据可以通过外部设备，例如键盘、磁盘输入，也可以在程序内部使用初始化、赋值方式获得。程序的执行结果可以输出到屏幕、打印机、文件，或者传送给其他程序。

VS.NET 用解决方案来管理项目。一个解决方案可以包含多个项目，每一个项目是一个独立运行的应用程序。但在实际应用中，一个解决方案中包含一个项目不失为一种好的习惯。

## 3. 程序错误类型

在 VS 平台上，不同类型项目所建立的解决方案，其保存、编译、调试、运行方式等都是相同的。利用 IDE 中的“文件”、“生成”、“调试”等菜单，可方便地完成应用程序的开发、测试和运行。

编写源程序时难免存在一些错误。这些错误分成如下 3 类。

### (1) 编译错误

编译源程序时发现的语法错误。例如，表达式  $(A+B*(C-D))$ ，缺了右括号。

### (2) 运行错误

执行目标程序时发现的错误。例如，执行标准函数  $\text{Sqr}(x)$ ，求  $x$  的平方根时， $x$  的值为负数。

### (3) 逻辑错误

编译和运行时均不能发现的错误。例如，由于程序员的笔误，把  $x^3$  写成  $x^5$ ，对  $x$  求立方变成求  $x$  的 5 次方了。

## 4. 程序设计目标

一个程序经常要经过反复的调试、验证才能完善，投入使用。为此，编写的程序应力求具有以下 3 个特点。

### (1) 正确性

这要求程序员熟悉所用的程序设计语言，避免语法、语义错误；设计简单易行的算法达到预期目的。对复杂的问题，则应考虑使用有效的程序设计方法。

## (2) 易读性

一个程序的结构清晰、易读，才能便于查错，便于修改。

## (3) 运行高效率

程序运行效率高是指运行时间较短，且占用的存储空间较小。

为达到以上目标，需要在不断学习和实践中提高程序设计水平。“程序”是人的智力产品。从理论上说，程序是永远不会损坏的。实际上，程序在整个生存周期都会根据需要进行修改和维护，都可能产生错误。程序错误有时会产生悲剧性的后果。程序的生产和维护比硬件产品复杂得多，计算机科学界期望有一套工程化的方法进行程序的开发和维护。为了体现这种工程思想，除源程序代码外，还应有一套开发、维护、使用完备的文档。程序加上这些相关文档称为软件。

# 1.3 Visual Studio 2010 集成开发环境

Visual Studio 2010 集成开发环境是设计、运行和测试应用程序所需的各种工具集成的工作环境。这些工具互相协调、互相补充，大大降低了程序员开发应用程序的难度。VS.NET 提供了各种类型的项目，但具有一致界面的集成开发环境，当用户进入 VS 2010，创建指定类型的项目或打开一个已有项目时，系统会自动打开相应的开发环境。例如，用户选择新建项目的类型为 Visual Basic，则系统打开的就是 VB.NET 的集成开发环境。

## 1.3.1 主窗口

在正确安装 VS 2010 后，单击“开始”按钮，打开“程序”→“Microsoft Visual Studio 2010”菜单，然后单击“Microsoft Visual Studio 2010”菜单命令，或双击桌面上的 Microsoft Visual Studio 2010 快捷图标启动 VS 2010 系统。启动成功后，屏幕上出现 VS 2010 集成开发环境主窗口，如图 1.1 所示。

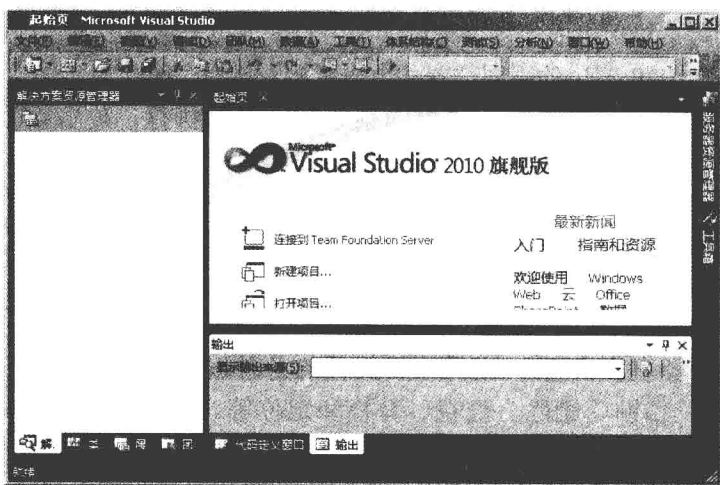


图 1.1 VS 2010 集成开发环境主窗口

VS 2010 集成开发环境主窗口包括标题栏、菜单栏、工具栏、项目工作区（包括解决方案资源管理器窗口等）、起始页窗口/编辑窗口和输出窗口等。

启动 VS 2010 后，IDE 中部是起始页。随着打开文件、建立项目，这个区域将以分页方式显示文件，并提供编辑功能。图 1.2 是打开一个 VB.NET 项目后显示的编辑窗口。

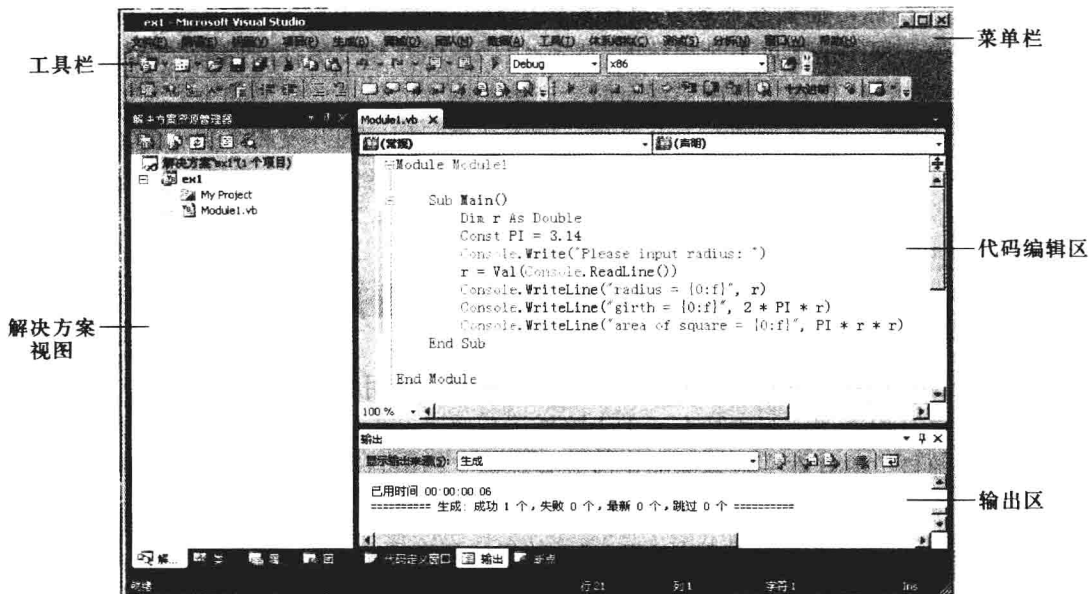


图 1.2 打开文件后的编辑窗口

下面对 VS 2010 集成开发环境主窗口的各个组成部分作简单介绍。

### 1.3.2 菜单栏

VS 2010 集成开发环境的菜单栏提供了开发应用程序的主要工具。菜单栏由文件、编辑、视图、窗口、帮助等菜单项组成。每个菜单项又由若干下拉子菜单或菜单项组成。

当鼠标指针指向 IDE 窗口的不同位置并单击鼠标右键时，可以弹出相应的快捷菜单。单击快捷菜单中的菜单项，可以执行与所处环境有关的命令。下面仅介绍几个常用的菜单项。

#### 1. “文件”菜单

“文件”菜单的功能是建立、打开、保存、打印解决方案文件。单击“文件”菜单项，弹出一个下拉文件菜单，如图 1.3 所示。

#### 2. “编辑”菜单

“编辑”菜单如图 1.4 所示，提供项目文件的各种编辑功能，如剪切、复制、粘贴、删除、查找、替换和定位等。

#### 3. “视图”菜单

“视图”菜单用于控制集成开发环境各个工作窗口的显示、打开、切换等，如图 1.5 所示。



图 1.3 “文件”菜单

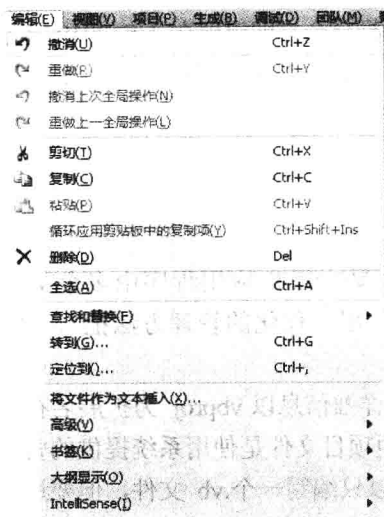


图 1.4 “编辑”菜单

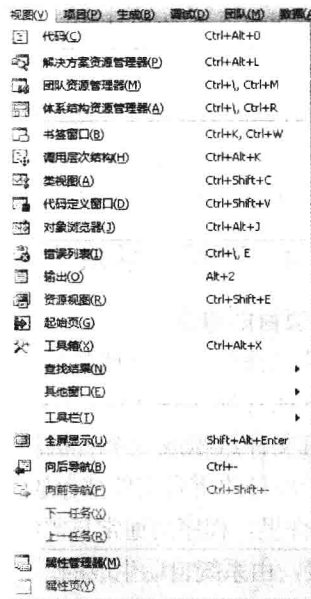


图 1.5 “视图”菜单

#### 4. “生成”菜单

“生成”菜单如图 1.6 所示，提供解决方案、项目、文件的编译、连接、生成、优化及清理等一系列功能，是生成项目执行文件的重要工具。

#### 5. “调试”菜单

“调试”菜单如图 1.7 所示，该菜单是程序测试的重要工具，包括分步执行、设置断点、建立监视项等功能。

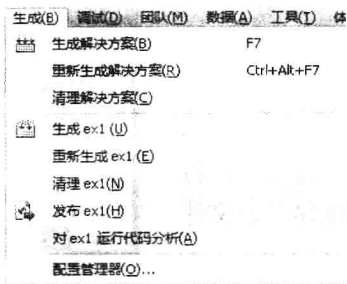


图 1.6 “生成”菜单

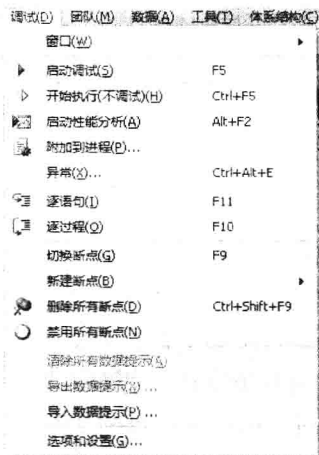


图 1.7 “调试”菜单

### 1.3.3 工具栏

VS 2010 集成开发环境包含十几种工具栏。工具栏提供图形化的按钮操作界面，相当于一



些常用的菜单命令的快捷方式。

在一般情况下，系统只显示标准工具栏。要使用其他工具栏，可以在主窗口的菜单栏上单击鼠标右键，在弹出的快捷菜单中选择所需项目，相应的工具栏图标就会立即出现在 VS 2010 主窗口中。

工具栏中包含很多个按钮，只要把鼠标指针指向这些按钮，并且稍微停留，相应的命令的名称就被显示出来。单击这些按钮就会执行对应的菜单命令。

### 1.3.4 项目、解决方案和项目工作区

#### 1. 项目的概念

文件是操作系统处理数据和代码的基本单位。一个 VB.NET 应用程序由多个文件组成，为了更好地管理这些文件，VS.NET 引入了项目的概念，用工程化的管理方法把一个应用程序中的所有文件组织成一个有机的整体。

一个项目的所有文件都存放在项目文件夹中，项目的详细信息以.vbproj 为扩展名存放在一个 XML 文件里。程序员通常只需编写源程序代码文件，其他项目文件是使用系统提供的资源经过编译、连接，由系统自动生成的。一个最简单的用户程序可以只编写一个.vb 文件，但要运行这个程序，必须使用相关的系统资源。因此，经过编译和连接后，系统会自动生成有关项目文件。

#### 2. 解决方案

解决方案是把程序和资源整合在一起的一种机制，这些程序和资源代表着某种数据处理的解决方法。

实际上，解决方案是存放一个项目或多个项目的信息的文件夹。一个解决方案的一个项目或多个项目的信息存储在扩展名为.sln 和.suo 的文件中。当新建一个项目时，除非选择添加项目到一个已存在的解决方案，否则系统会自动生成一个新的解决方案。从程序开发的角度讲，每一个项目都应该有对应的解决方案。本书的每一个例程都有一个单独的项目和它自己的解决方案。

#### 3. 项目工作区

VS 2010 以工作区的形式来组织文件和项目，即项目置于工作区的管理之下，因而该工作区通常称为项目工作区。一个项目工作区可以包含各种文件及文件夹。在 VS 2010 的 IDE 中，项目工作区窗口以树状形式列出当前项目的文件。用户通过项目工作区窗口可以方便地操作这些文件。

项目工作区窗口有 4 个选项卡：解决方案资源管理器、类视图、属性管理器和团体资源管理器。用户可以用不同的方式操作项目。

解决方案资源管理器（如图 1.8 所示）用来管理组成解决方案的所有项目和项目中的文件。单击“解决方案资源管理器”选项卡，将显示项目中的各类文件，若双击某个文件，文件编辑区就显示该文件的内容。此时，用户可以对文件进行编辑和修改。



图 1.8 解决方案资源管理器

### 1.3.5 Visual Studio 2010 帮助系统的使用

与其他的 Windows 应用程序一样，VS 2010 也有一个功能强大的帮助系统，称为 MSDN。MSDN 包含开发人员所需的信息、文档、示例代码、技术文章等，是学习 VB.NET 和开发应用程序的有力助手。



若要在 VS 2010 环境中使用帮助, 需要做一些相关的设置。方法是打开“帮助”菜单, 选择“管理帮助设置”(Ctrl+Alt+F1), 打开 Help Library 管理器, 如图 1.9 所示, 可以安装本地帮助库或选择联机帮助。设置完成后, 就可以使用 F1 热键或者“帮助”菜单查看 MSDN 的相关帮助文档。

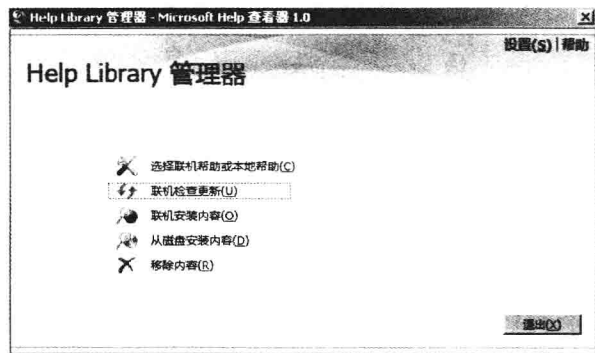


图 1.9 帮助窗口

## 1.4 建立控制台应用程序

控制台程序是学习 VB.NET 程序设计基础的一个简捷方式。一个 VB.NET 程序的开发需要经过编辑、编译、连接和运行等步骤, 一般过程如图 1.10 所示。本节通过一个简单例程学习 VB.NET 程序设计的上机实际操作方法。

### 【例 1-1】

**问题** 输入圆的半径, 计算输出圆的周长和面积。

**分析** 用户输入一个数值, 在接收用户的输入信息后, 用以下的公式计算圆的周长和面积:

$$\text{圆周长} = 2\pi r$$

$$\text{圆面积} = \pi r^2$$

然后把结果输出到屏幕上。

### 1. 创建新项目

#### (1) 输入源程序

① 启动 VS 2010 后, 屏幕上出现 VS 2010 集成开发环境主窗口, 如图 1.1 所示。

② 单击“文件”→“新建”→“项目”菜单命令(或者按下 Ctrl+Shift+N 组合键), 打开“新建项目”对话框, 如图 1.11 所示。

③ 在“模板”框中选择“Visual Basic”、“控制台应用程序”。

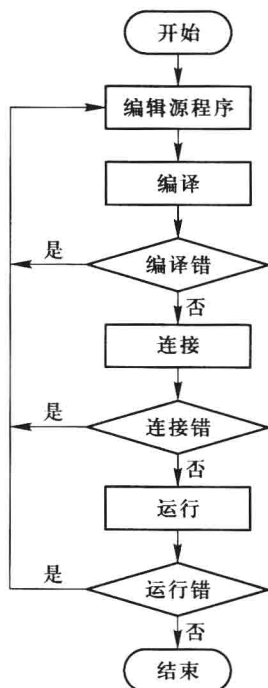


图 1.10 C++程序的开发过程