

普通高等院校计算机基础教育“十三五”规划教材
以培养创新能力为核心的信息技术基础系列教材



Visual Basic.NET 程序设计教程

Visual Basic.NET CHENGXU SHEJI JIAOCHENG

胡浩民 主 编
张晓梅 王泽杰 向珏良 副主编

课外读

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

普通高等院校计算机基础教育“十三五”规划教材

Visual Basic.NET 程序设计教程

胡浩民 主 编

张晓梅 王泽杰 向珏良 副主编

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书是面向非计算机专业本科学生学习程序设计类课程的教材，内容涵盖了 Visual Basic.NET 概述、Visual Basic 语言基础、Visual Basic.NET 控件、控制结构、数组、过程、用户界面设计、面对对象程序设计、Visual Basic.NET 绘图以及文件等。每章后配有课后习题，以帮助读者巩固相应的知识点，并提高通过程序设计解决实际问题的能力。

本书适合作为高等学校程序设计课程的教材、计算机等级考试的指导用书，也可作为程序设计爱好者的自学参考书。

图书在版编目 (CIP) 数据

Visual Basic.NET 程序设计教程/胡浩民主编. —北京:
中国铁道出版社, 2019.1

普通高等院校计算机基础教育“十三五”规划教材

ISBN 978-7-113-25354-7

I. ①V… II. ①胡… III. ①BASIC 语言-程序设计-高等
学校-教材 IV. ①TP312.8

中国版本图书馆 CIP 数据核字 (2019) 第 012823 号

书 名: Visual Basic.NET 程序设计教程
作 者: 胡浩民 主编

策 划: 曹莉群

读者热线: (010) 63550836

责任编辑: 陆慧萍 冯彩茹

封面设计: 刘 颖

责任校对: 张玉华

责任印制: 郭向伟

出版发行: 中国铁道出版社 (100054, 北京市西城区右安门西街 8 号)

网 址: <http://www.tdpress.com/51eds/>

印 刷: 北京虎彩文化传播有限公司

版 次: 2019 年 1 月第 1 版 2019 年 1 月第 1 次印刷

开 本: 787 mm×1 092 mm 1/16 印张: 14.25 字数: 345 千

书 号: ISBN 978-7-113-25354-7

定 价: 42.00 元

版权所有 侵权必究

凡购买铁道版图书, 如有印制质量问题, 请与本社教材图书营销部联系调换。电话: (010) 63550836

打击盗版举报电话: (010) 51873659

» 前 言



D. E. KNUTH 在《计算机程序设计的艺术》一书中提到：“为一台数字计算机编写程序的过程是饶有趣味的，因为它不但具有经济和科学价值，而且也是犹如赋诗或作曲那样的美学实践……”五十多年来，程序设计体现的创造力令人瞩目。尽管目前大数据、人工智能、云计算、物联网等新的领域不断地拓展，但其深层次的研究依然离不开程序设计方法的应用。

程序设计进入大学课堂由来已久，本书就是根据编者在讲授“Visual Basic 程序设计课程”多年来所取得的实践经验和教学研究成果的基础上编写而成的。编写的过程中，我们始终把握着“两个关键”：

(1) 技术本身的发展。现如今，新的程序设计语言层出不穷，原有的程序设计语言也在不断地发展完善。Visual Basic 作为一种多范式、面向对象的编程语言，随着微软公司 .NET 软件开发平台的演进，已经于 2017 年 3 月升级为 Visual Basic 15.0。然而，作为一门好的程序设计入门语言，Visual Basic 多年来仍然是保持相对稳定，依然具备入门学习的两大优势：一是语法简单。Visual Basic 具有功能强大的内置数据结构，减轻了记忆背诵的负担，能够将学习者的精力聚焦到寻找问题解决方法上来。二是应用广泛。Visual Basic 广泛应用于工业界和学术界，常用于解决网络访问、数据库操作等实际问题，能够为学习者学以致用提供更大的空间。

(2) 结合学生的需求。全书内容涵盖了 Visual Basic.NET 概述、Visual Basic 语言基础、Visual Basic.NET 控件、控制结构、数组、过程、用户界面设计、面向对象程序设计、Visual Basic.NET 绘图以及文件等，内容设计力求突出实践性与应用性，在培养学生程序设计基本能力的同时，加深和拓宽知识面，使学生具有应用 Visual Basic .NET 解决专业技术领域问题的编程能力。此外，我们更加希望借由本书，探索培养学生的计算思维（Computational Thinking）能力。计算思维是运用计算机科学的基本概念进行问题求解、系统设计以及人类行为理解的、涵盖计算机科学之广度的一系列思维活动，通俗地讲就是要做到“像计算机科学家一样思考”。作为基础学科的教师，我们应该改变长期以来存在的“计算机只是工具”“计算机就是程序设计”和“计算机基础课程主要是讲解软件工具的应用”等刻板印象，主动承担起培养学生综合素质与能力的重任。简言之，计算机基础教学不应只是要学生会如何使用计算机或进行程序设计，而程序设计类课程应该将培养学生的计算思维能力作为主要目标，本系列教材在编写时始终遵循这一原则。



本书由胡浩民任主编，张晓梅、王泽杰、向珏良任副主编。具体编写分工如下：第 1 章由胡浩民、向珏良编写，第 2~4 章由周晶、胡浩民编写，第 5 章由王泽杰编写，第 6 章由张晓梅编写，第 7~8 章由刘惠彬、张晓梅编写，第 9 章由向珏良编写，第 10 章由胡浩民编写。全书由胡浩民统稿。在编写的过程中还得到了陈强、赵毅、黄容、潘勇、胡建鹏等老师的帮助，在此一并表示感谢。

由于编者水平有限，加之时间仓促，书中难免存在疏漏和不足之处，敬请专家和读者批评指正。

编 者

2018 年 11 月于上海

» 目 录



第1章 Visual Basic.NET概述 1

1.1 计算机编程语言 1

1.1.1 编程语言的分类 1

1.1.2 Visual Basic 语言 1

1.2 Visual Basic.NET 集成开发环境 3

1.2.1 建立 Visual Basic.NET 解决方案 4

1.2.2 集成环境窗口的标题栏、 菜单栏和工具栏 6

1.2.3 对象的可视化布局窗口 6

1.2.4 代码编程窗口 7

1.2.5 “解决方案资源管理器” 窗口 8

1.2.6 其他窗口 9

1.2.7 窗口的布置 9

1.2.8 打开已经存在的项目 10

1.3 可视化程序设计步骤 10

1.3.1 程序设计三步骤 10

1.3.2 Visual Basic.NET 项目树 结构 12

1.3.3 Visual Basic.NET 程序结构 和编码规则 13

1.4 .NET 框架 14

1.4.1 .NET 框架概述 14

1.4.2 帮助文档 15

课后习题 17

第2章 Visual Basic.NET语言基础 20

2.1 数据类型 20

2.1.1 基本数据类型 21

2.1.2 标识符 22

2.2 变量与常量 23

2.2.1 常量 23

2.2.2 变量 24

2.3 运算符和表达式 26

2.3.1 运算符 26

2.3.2 表达式 27

2.4 常用函数 28

2.4.1 数学函数 28

2.4.2 类型转换函数 28

2.4.3 字符串函数 29

2.4.4 日期和时间函数 30

2.4.5 随机函数 31

2.5 命名空间 31

2.5.1 命名空间概念 31

2.5.2 命名空间的使用 32

课后习题 33

第3章 Visual Basic.NET控件 38

3.1 基本概念 38

3.1.1 类和对象 38

3.1.2 控件类与控件对象 39

3.1.3 对象的属性、方法和事件 40

3.2 窗体 41

3.2.1 窗体的属性 42

3.2.2 窗体的事件 43

3.2.3 窗体的常用方法 43

3.3 基本控件 45



3.3.1 标签	45	5.2.2 一维数组的基本处理	91
3.3.2 文本框	46	5.2.3 多维数组	92
3.3.3 命令按钮	48	5.2.4 动态数组	94
3.3.4 复选框	49	5.3 数组处理的常用算法	96
3.3.5 单选按钮和分组框	50	5.3.1 排序	96
3.4 更多属性、事件和方法	52	5.3.2 插入、删除元素	100
3.4.1 属性	52	5.4 结构数据类型与结构数组	101
3.4.2 事件	53	5.4.1 结构数据类型	101
3.4.3 方法	53	5.4.2 结构数组	103
3.4.4 共享事件	54	5.5 Array 类与控件数组	104
课后习题	55	5.5.1 使用 Array 类处理数组	104
第4章 控制结构	59	5.5.2 控件数组的定义与使用	106
4.1 顺序结构	59	课后习题	107
4.1.1 赋值语句	59	第6章 过程	113
4.1.2 数据输入和数据输出	60	6.1 过程的概念	113
4.2 选择结构	63	6.2 子过程	115
4.2.1 If 条件语句	63	6.2.1 子过程的定义	115
4.2.2 Select Case 语句	66	6.2.2 子过程的调用	115
4.2.3 选择结构的嵌套	68	6.3 函数过程	117
4.2.4 条件函数	69	6.3.1 函数过程的定义	117
4.3 循环结构	69	6.3.2 函数过程的调用	117
4.3.1 Do...Loop 循环语句	69	6.4 参数传递	118
4.3.2 For...Next 循环语句	74	6.4.1 形参和实参	118
4.3.3 循环结构的嵌套	75	6.4.2 传值和传地址	119
4.3.4 循环结构的三要素	79	6.4.3 数组参数	120
4.4 异常处理与调试	79	6.5 变量的作用域与生存期	122
4.4.1 错误的种类	80	6.5.1 变量的作用域	122
4.4.2 调试和排错	81	6.5.2 静态变量	124
课后习题	83	6.6 递归	125
第5章 数组	88	6.7 多线程	126
5.1 数组的概念	88	6.7.1 程序、进程与线程	126
5.2 数组的声明和引用	88	6.7.2 操纵多线程	127
5.2.1 一维数组的声明和初始化	89	课后习题	130

第7章 用户界面设计	134	8.2.5 定义事件	162
7.1 常用控件	134	8.2.6 定义构造函数和析构函数	163
7.1.1 列表框和组合框	134	8.2.7 对象及其成员的访问	163
7.1.2 滚动条和进度条	138	8.3 继承和派生	166
7.1.3 图片框	139	8.3.1 继承的概念	166
7.1.4 定时器	141	8.3.2 继承的实现	166
7.2 通用对话框	143	8.3.3 派生类的构造函数	166
7.2.1 “颜色”对话框	143	8.4 接口与多态	168
7.2.2 “字体”对话框	144	8.4.1 接口	168
7.3 多窗体程序设计	145	8.4.2 多态	170
7.3.1 添加窗体	146	课后习题	172
7.3.2 窗体的显示和隐藏	146	第9章 Visual Basic.NET绘图	173
7.3.3 设置启动窗体	147	9.1 画布概述	173
7.3.4 窗体间的数据传递和引用	147	9.1.1 画布绘图	173
7.3.5 多窗体的综合应用	147	9.1.2 画布书写	174
7.4 菜单设计	148	9.2 GDI+绘图基础	174
7.4.1 下拉菜单	148	9.2.1 GDI+绘图的基本类和画图	175
7.4.2 快捷菜单	150	9.2.2 GDI+绘图的相关对象	175
7.5 拓展学习	151	9.2.3 画布的默认坐标系与变换	176
7.5.1 鼠标和键盘事件	151	9.3 画布绘图与书写	177
7.5.2 更多通用对话框	153	9.3.1 画笔与绘图方法	177
课后习题	155	9.3.2 画刷与填充方法	181
第8章 面向对象程序设计	157	9.3.3 字体与书写方法	182
8.1 面向对象的基本概念	157	9.4 绘制函数图形	184
8.1.1 面向对象技术	157	9.4.1 绘制 $y=f(x)$ 函数图形	184
8.1.2 类和对象	157	9.4.2 绘制参数方程曲线	187
8.1.3 类的基本特性	158	9.5 绘图扩展	189
8.2 创建类和对象	159	9.5.1 坐标变换与坐标系变换	189
8.2.1 定义类	159	9.5.2 窗体造型	191
8.2.2 定义数据成员	161	课后习题	193
8.2.3 定义属性	161		
8.2.4 定义方法	162		



第10章 文件	195
10.1 文件概述	195
10.1.1 文件类型	195
10.1.2 Visual Basic.NET 访问文件 的方式	197
10.1.3 文件操作步骤	197
10.2 运行时函数访问方式	197
10.2.1 常用运行时函数	197
10.2.2 顺序文件访问	199
10.2.3 随机文件访问	204
10.2.4 二进制文件访问	207
10.3 System.IO 访问方式	209
10.4 My.Computer.FileSystem 访问 方式	215
课后习题	217
参考文献	220

第1章

Visual Basic.NET概述



自然语言是人与人沟通的桥梁，程序设计语言则是人与计算机交流的纽带。Visual Basic（简称 VB）是微软公司开发的一种通用的基于对象的程序设计语言。Visual Basic.NET（简称 VB.NET）是以 Visual Basic 为编程语言，以微软.NET 框架为核心的集成开发环境。本章从案例入手，展示 Visual Basic.NET 的可视化布局与设计、事件驱动与代码编写、调试与运行的集成环境，以及使用该环境进行程序设计的方法和步骤。



1.1 计算机编程语言

1.1.1 编程语言的分类

计算机语言可分为低级语言和高级语言。低级语言分又可分为机器语言和汇编语言。

机器语言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。它是计算机设计者通过硬件结构赋予计算机的操作功能。机器语言具有直接执行和速度快等特点，但是不同型号计算机的机器语言是不相通的，因此程序的可移植性差。此外，编程人员需要熟记所用计算机的全部指令代码和代码的含义，直观性差，容易出错。

汇编语言则是使用助记符和地址符来代表机器指令中的操作码和操作数，从而降低了程序的编写难度，增强了程序的可读性。但计算机不能直接识别这种使用汇编语言编写的程序，需要由汇编语言编译器转换成机器指令。汇编语言是面向机器的低级语言，通常是为特定的计算机或系列计算机专门设计的，缺乏可移植性。

高级语言是以人类自然语言为基础的编程语言，它与计算机的硬件结构及指令系统无关，因此具有更强的表达能力，程序可读性也得到很大的提高。高级语言不是特指某一种具体的语言，而是包括很多编程语言，如 C/C++、Java、C#、Pascal、Visual Basic、Python 等，不同语言的语法会有所不同。高级语言在执行时需要编译成计算机能识别的语言。

1.1.2 Visual Basic 语言

Visual Basic 是微软公司开发的一种通用的基于对象的程序设计语言，也是结构化、模块化、面向对象、并且包含协助开发环境的事件驱动为机制的可视化程序设计语言。Visual Basic.NET 则是以 Visual Basic 为编程语言，提供了代码编写、编译、调试与运行的集成开发环境。鉴于



Visual Basic 语言和 .NET 工具之间的依赖关系, 本书以 Visual Basic.NET 作为 Visual Basic 和 .NET 开发环境的统称。以下通过例子来说明如何在 Visual Basic.NET 中设计一个简单的程序。

例题 1.1 设计一个计算函数 $y = x^2 + 1$ 值的程序。实现从键盘输入 x 值, 通过计算后, 将结果值 y 输出。

设计分析: 从数学方程解题思路来看, 该问题求解可分为四个步骤: ① 设未知数变量; ② 未知数设初值; ③ 根据等量关系进行计算; ④ 答结果。程序设计与数学解题非常相似, 可以参照这些步骤来进行编程。

启动开发工具 Microsoft Visual Studio 后, 在图 1.1 所示的“新建项目”窗口, 选择 Visual Basic 语言的“控制台应用程序”。再输入项目的名称和存放位置。

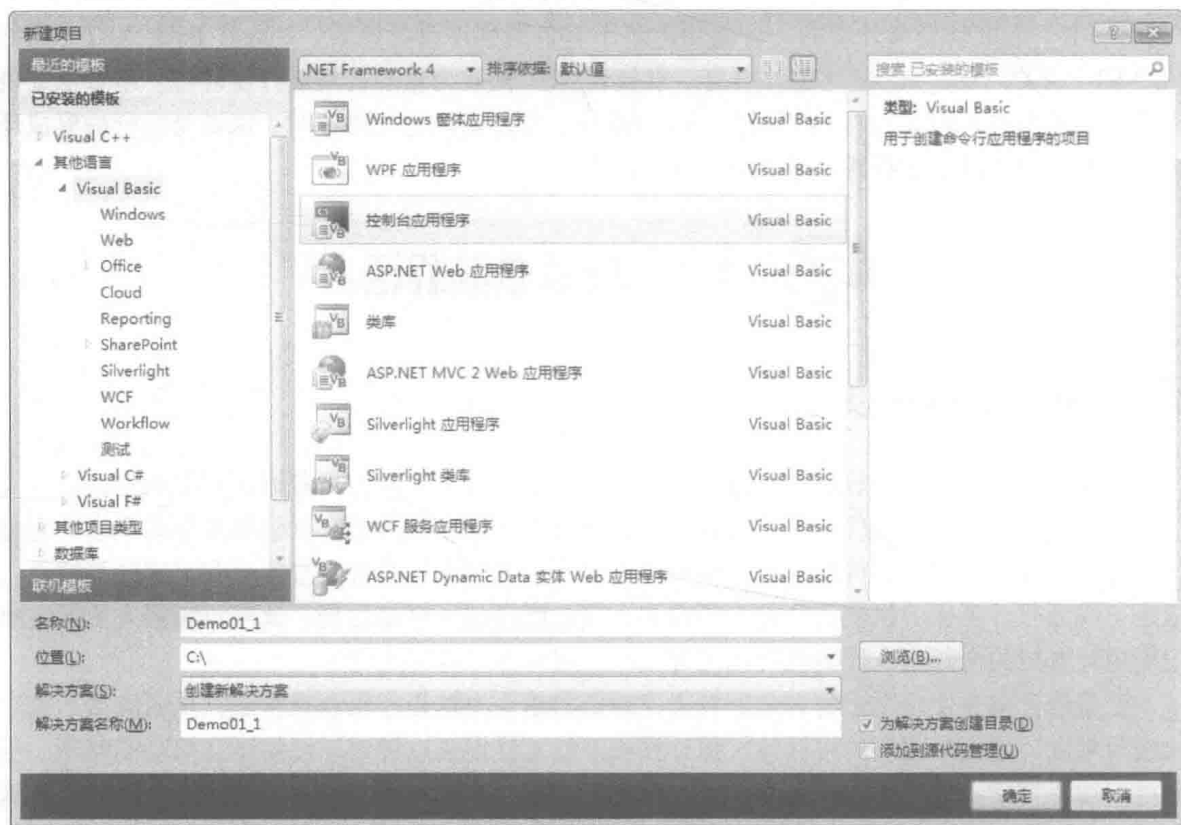


图 1.1 “项目”对话框

单击“确定”按钮后, Visual Basic.NET 会在选择的位置创建以项目名称命名的文件夹。项目有关的所有文件将会存放在该文件夹下。Visual Basic.NET 开发环境会自动创建 Module1.VB 模块文件。并且在模块内创建 Sub Main()...End Sub 的主过程, 如图 1.2 所示。该过程是 Visual Basic.NET 控制台应用程序的入口, 程序运行时会逐行执行 Main 过程中的代码。虽然 Windows 窗体应用程序的 Main 过程是 Visual Basic 编译器自动生成的, 无须用户显式创建, 但了解 Main 过程的作用有利于理解程序运行的机制与原理。

根据“设计分析”的四个步骤, 在 Main 过程中输入图 1.3 所示的四行代码。(单引号'后的中文表示“注释语句”, 只起到解释说明的作用, 编译器不会编译运行, 所以无须输入。)

代码中 Dim x, y As Double 表示: 声明 x 和 y 是两个可以带小数的浮点型变量; Console 表示

控制台应用程序的标准输入/输出流。Console.ReadLine 方法从标准输入流（即键盘）读取数据，Console.WriteLine 方法将数据写到标准输出流（即显示器），并加以行结束的“回车符”。

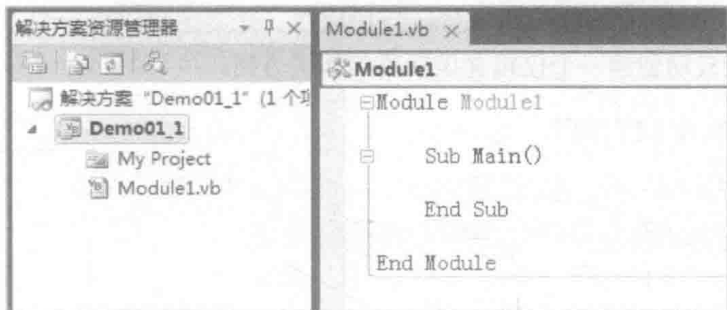


图 1.2 模块中的 Main 过程

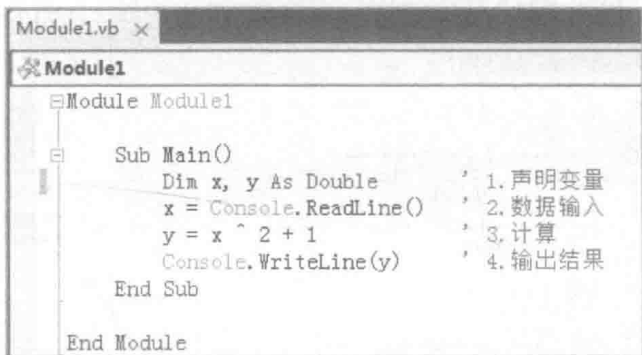


图 1.3 例 1.1 的程序代码

单击工具栏中的“启动调试”按钮或按 F5 键即可运行程序，效果如图 1.4 所示。



图 1.4 例 1.1 运行效果



1.2 Visual Basic.NET 集成开发环境

例题 1.1 的控制台应用程序结构清晰，易于理解。但在 Windows 操作系统中，更多地会使用界面友好的“窗体应用程序”。以下通过创建求解例题 1.1 的“Windows 窗体应用程序”为例加以说明。

Visual Studio.NET 集成环境（IDE）提供了 .NET 框架下的 Visual Basic、C#、C++、J# 开发语言平台。该 IDE 既满足传统的桌面应用程序开发，也适合 Web 应用程序、嵌入式移动设备程序的开发，还提供了许多中间件、类库的开发平台。因此，熟悉 Visual Studio.NET 集成环境是学习 Visual Basic.NET 编程的基础。



1.2.1 建立 Visual Basic.NET 解决方案

要创建一个 Visual Basic.NET 的 Windows 应用程序，先要创建一个 Visual Basic.NET 项目。Visual Studio.NET 会自动创建一个仅包含该项目的解决方案。

1. 创建 Visual Basic.NET 项目

进入 Visual Studio.NET 后，有两种创建 Visual Basic.NET 项目的方法：

启动页：在“最近的项目”栏中，单击“创建”标签行的“项目”。

菜单方式：执行“文件”→“新建”→“项目”命令。

2. 选择程序模板

进入图 1.5 所示的“新建项目”对话框，在“项目类型”中选择 Visual Basic 下的“Windows”；“模板”中选择“Windows 窗体应用程序”。



图 1.5 “新建项目”对话框

3. 拟定项目名称

默认项目名称为“WindowsApplication1”。在“名称”处可以修改成用户想命名的项目名（如“例题 1.1 函数求值”）。单击“确定”按钮，进入图 1.6 所示的 Visual Basic.NET 集成环境的编程界面。

4. 建立并保存解决方案

建立解决方案后，Visual Basic.NET 会根据新建项目时设置的解决方案“名称”与选择“位

置”自动保存有关的所有数据。若在新建项目时，没有出现解决方案“名称”与“位置”，则可执行“工具”→“选项”命令，在弹出的对话框中选择“项目和解决方案/常规”，勾选“创建时保存新项目”复选框，如图 1.7 所示。建议把第 3、4 步合并，这样在创建项目时就可指定存储路径，并且真正同时物理存储项目和解决方案。

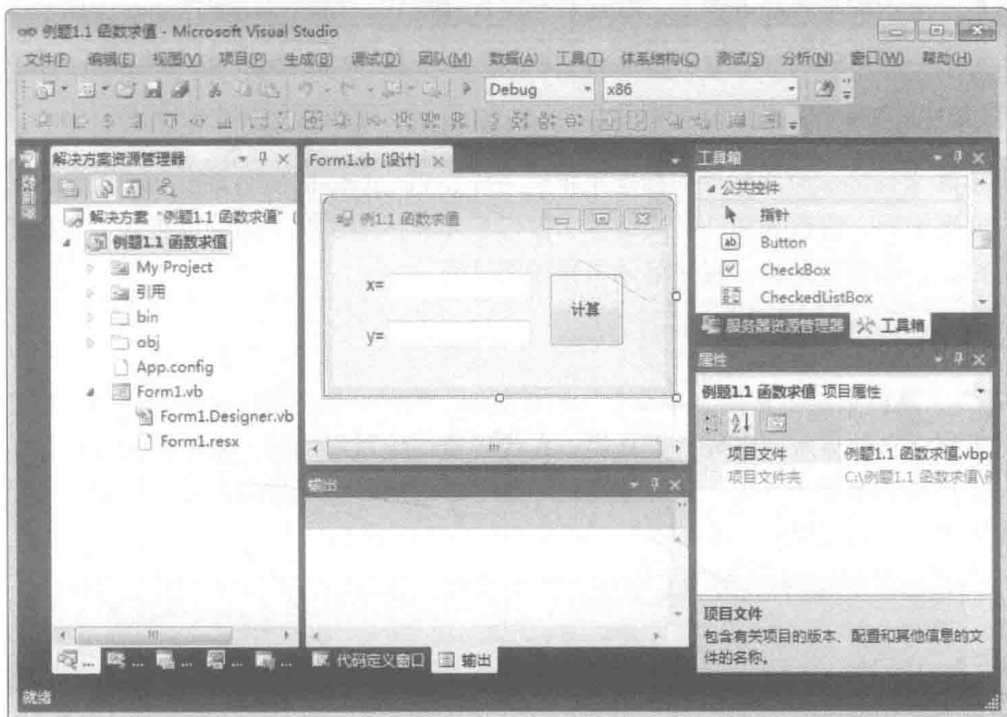


图 1.6 Visual Basic.NET 集成环境与对象可视化布局

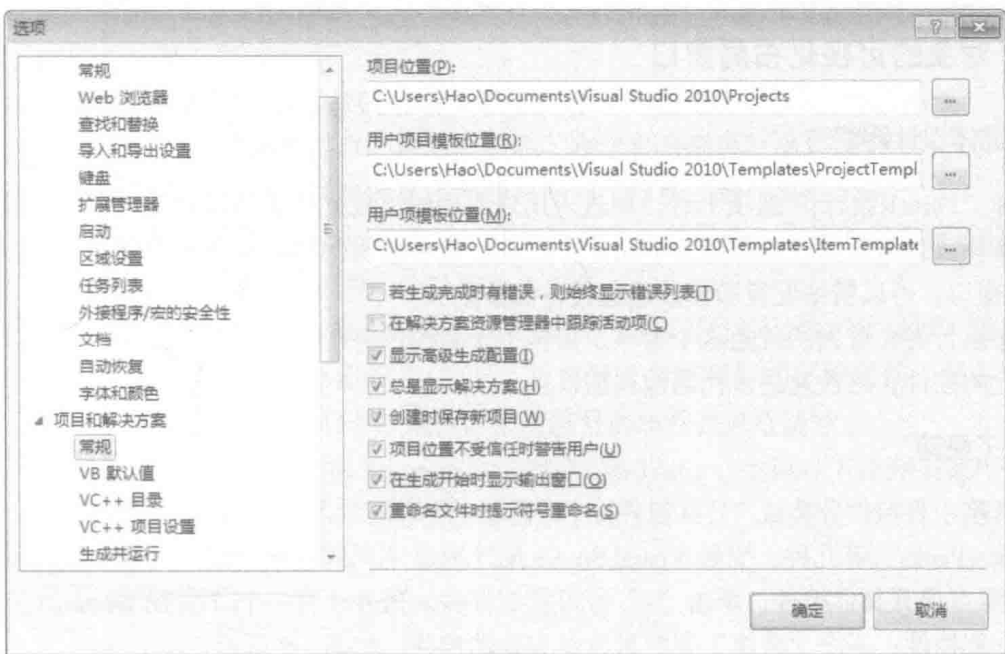


图 1.7 “选项”对话框



1.2.2 集成环境窗口的标题栏、菜单栏和工具栏

1. 标题栏

标题栏上显示的是解决方案名。常规状态是设计模式，如果启动程序运行，则会显示“正在运行”；遇到错误或断点暂停程序时，则会显示“正在调试”。

2. 菜单栏

与大多数 Windows 应用程序的标准菜单栏一样，Visual Basic.NET 集成环境菜单也包含了文件处理、文字编辑、窗口视图、格式编排、窗口布局和帮助系统，同时也包含了与程序紧密相关的菜单栏目，如项目配置、程序编译和程序调试等。

3. 工具栏

“标准”工具栏集结了常用菜单命令，包括复制、粘贴、运行程序等命令按钮。在窗体布局设计时，系统会自动显示“布局”工具栏；在代码窗口编写程序时，系统又会自动显示“文本编辑器”工具栏。通过“视图”→“工具栏”命令，可以个性化显示工具栏配置。

集成环境窗口的标题栏、菜单栏和工具栏如图 1.8 所示。



图 1.8 集成环境窗口的标题栏、菜单栏和工具栏

1.2.3 对象的可视化布局窗口

1. 窗体设计器窗口

选择“Form1[设计]”选项卡时，即进入用户界面的可视化“窗体设计器”窗口。借助于工具箱，窗体设计时就像在画图板上“贴”上控件，也可可视化移动控件和改变其大小；借助于“属性”窗口，可以精细配置和修改这些控件的参数。

程序运行时，看到的就是这个窗体及窗体中布置的所有控件对象。窗体设计时，双击窗体或窗体上的控件，可快速进入代码编程窗口。

2. 工具箱

工具箱中将控件分类成“公共控件”“容器”“菜单和工具栏”“数据”“对话框”“Visual Basic PowerPacks”等几种，安装 Visual Studio.NET 时，不同的选项，控件内容会有稍有不同。单击“+”号可展开具体控件，单击“-”号可收缩控件。另外还有一个“所有 Windows 窗体”栏包含了全部控件；一个“常规”空栏可以保存自选控件，如图 1.9 所示。

注意工具箱上的大头针按钮，单击后变成（拔出大头针状态）。此时，当鼠标指针移出工具箱，整个工具箱将会收缩至集成环境的左边（仅显示一个工具箱标签），腾出大片窗体空间

供设计使用。当鼠标指针悬停在左边收缩的工具箱标签上，工具箱会再次展开。根据需要，可以单击按钮，使其变成（按下大头针状态），工具箱又牢牢地钉在集成环境窗体上。

3. “属性”窗口

“属性”窗口也可显示“事件”。“属性”窗口由 4 部分组成（假设在窗体设计器上选定控件 Button1），如图 1.10 所示。

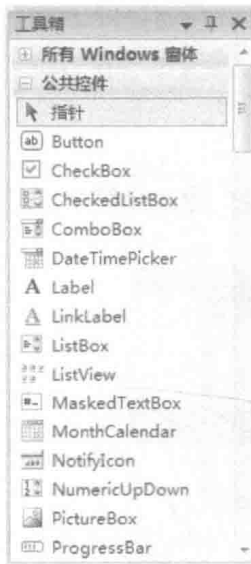


图 1.9 “工具箱”窗口

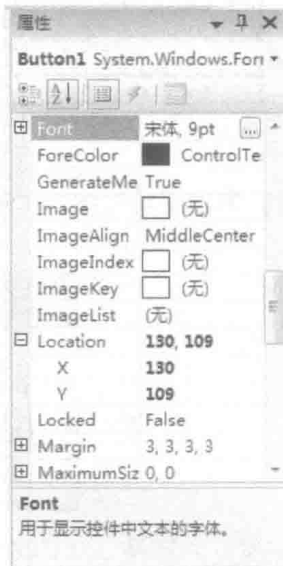


图 1.10 “属性”窗口

(1) “控件名/父类名”下拉列表框：显示当前选定的控件是“Button1”，以及它归属的父类“System.Windows.Forms.Button”。也可以在这个下拉列表框中选择其他控件，改变当前的选择控件，窗体设计器上同步改变。

(2) 命令按钮：显示内容的排列顺序和“属性/事件”控制。

① 按钮：分别将属性内容“按分类顺序”和“字母顺序”排列。

② 按钮：分别控制在属性窗显示当前控件的“属性”和“事件”。

③ 属性或事件内容主体：左边是控件“属性”或“事件”名；右边是其对应的值。

④ 注释文本：当前属性的注释。

如果把窗体设计器称作为可视化设计，则“属性”窗口可称作为数字化设计。前者直观简便，后者精细准确。单击属性名前的“+”号，可以展开当前属性的下级属性；单击属性名前的“-”号，可以收缩当前属性的下级属性。没有符号的属性是末级属性。

属性值可以直接输入数值（如 Location）、文字（如 Text），也可在下拉列表框中选择（如 ForeColor），或者单击按钮打开对话框设置（如 Font）。

1.2.4 代码编程窗口

选择“Form1.vb”选项卡，或者双击控件时，即进入代码编程窗口，如图 1.11 所示。窗口上方左边下拉列表框可以选择控件对象，右边下拉列表框可以选择当前控件对象的事件。在代



码编程窗口中，系统自动构造事件的代码框架，只需要在框架内编写或修改代码即可。代码编写时要充分运用 Visual Studio.NET 的智能感知技术和格式自动对齐技术。

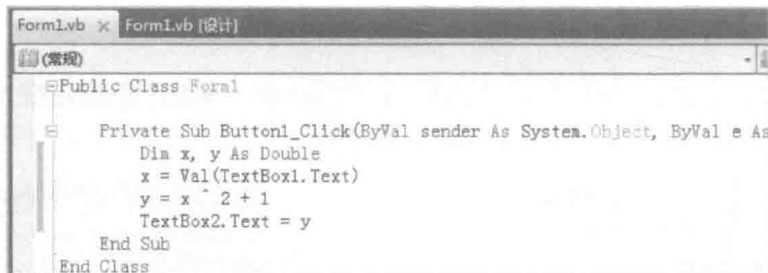


图 1.11 代码编程窗口

输入代码后，单击工具栏中的“启动调试”按钮或按 F5 键运行程序，窗体程序运行效果如图 1.12 所示。



图 1.12 窗体程序运行效果

1.2.5 “解决方案资源管理器”窗口

Windows 应用程序就是一个包含若干个项目的解决方案容器包。图 1.13 所示的“解决方案资源管理器”窗口中可以看到仅包含了 1 个项目。项目名为“例题 1.1 函数求值”，是一棵项目树，不断展开可以看到这棵树下的所有相关文件。将鼠标指针悬停在命令按钮上可知：

- ：显示所有文件（包含隐藏的“引用”、bin、obj）。
- ：刷新项目树下的更新内容。
- ：查看代码，进入“代码编程”窗口。
- ：进入“窗体设计器”窗口。

在资源管理器中，通过右键快捷菜单可以方便地进行复制、删除、粘贴和重命名等操作。



图 1.13 “解决方案资源管理器”窗口