



工业和信息化部普通高等教育“十二五”规划教材立项项目

21世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

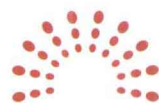
Visual Basic.NET 程序设计教程

An Introduction to Visual Basic.NET
Programming

兰顺碧 主编

阙向红 万奕 杨向东 齐敏 吴霞 王芬 编著

- 以Visual Basic.NET 2010为平台
- 帮助读者快速掌握程序设计的方法
- 实例由简到繁，阐述由浅入深



高校系列

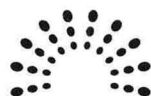
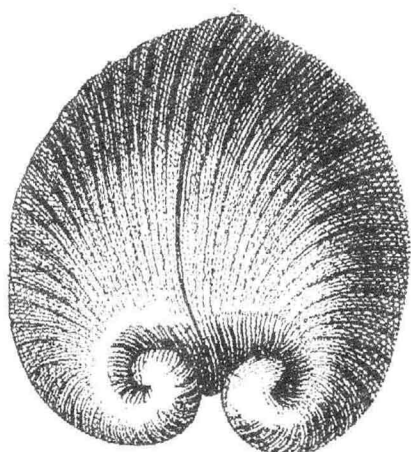
人民邮电出版社
POSTS & TELECOM PRESS

Visual Basic.NET 程序设计教程

An Introduction to Visual Basic.NET
Programming

兰顺碧 主编

阙向红 万奕 杨向东 齐敏 吴霞 王芬 编著



高校系列

人民邮电出版社

北京

图书在版编目 (CIP) 数据

Visual Basic.NET 程序设计教程 / 兰顺碧主编. --
北京: 人民邮电出版社, 2012. 2
21世纪高等学校计算机规划教材
ISBN 978-7-115-27325-3

I. ①V… II. ①兰… III. ①
BASIC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(2011)第274196号

内 容 提 要

本书从最简单的操作、最基本的概念入手, 由简到繁、由浅入深地介绍 Visual Basic.NET 2010 程序设计。本书结合最基本的内容, 通过丰富的实例, 阐述基本编程方法和程序设计技巧, 并详尽地介绍了 VB.NET 语言基础、程序的基本控制结构、数组、过程、用户界面设计、通用对话框和菜单、图形、面向对象的编程、建立类库、文件、访问数据库及调试和错误处理等内容。书中提供的有针对性的实例、精心编排的内容和科学的学习顺序是初学者深入理解“面向对象”思想和从入门到精通的保证。

全书分两部分, 第一部分是 Visual Basic.NET 2010 教程。各章都设计了许多应用实例和综合应用, 编排合理, 概念清晰, 有针对性地提高读者计算机程序设计的能力。第二部分共有 10 个实验, 认真完成每个实验, 就能在较短的时间内基本掌握 Visual Basic.NET 2010 及其应用技术。

本书适合作为高等院校计算机公共课教学及专科生相关课程的教材, 也可供学习 Visual Basic.NET 2010 程序设计的读者参考。

工业和信息化部普通高等教育“十二五”规划教材立项项目

21世纪高等学校计算机规划教材

Visual Basic.NET 程序设计教程

-
- ◆ 主 编 兰顺碧
 - 编 著 阙向红 万 奕 杨向东 齐 敏 吴 霞 王 芬
 - 责任编辑 武恩玉
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
三河市海波印务有限公司印刷
 - ◆ 开本: 787×1092 1/16
印张: 19 2012年2月第1版
字数: 477千字 2012年2月河北第1次印刷

ISBN 978-7-115-27325-3

定价: 36.00 元

读者服务热线: (010)67170985 印装质量热线: (010)67129223

反盗版热线: (010)67171154

广告经营许可证: 京崇工商广字第 0021 号

目 录

第一部分 Visual Basic.NET 2010 教程

第 1 章 概论	1
1.1 Visual Basic.NET 概述	1
1.1.1 第一个 VB.NET 程序	2
1.1.2 基本概念	3
1.2 VB.NET 集成开发环境	7
1.2.1 主窗口	8
1.2.2 菜单栏和工具箱窗口	9
1.2.3 解决方案资源管理器	10
1.2.4 属性窗口	12
1.2.5 窗体设计器与代码编辑器窗口	12
1.3 创建 VB.NET 应用程序	13
1.3.1 VB.NET 应用程序结构与 编码规则	13
1.3.2 创建 VB.NET 应用程序的 步骤与规范	14
1.4 窗体与基本控件	15
1.4.1 窗体	15
1.4.2 基本控件	16
1.5 综合应用	19
习题	21
第 2 章 VB.NET 语言基础	22
2.1 数据类型	22
2.1.1 数值型	23
2.1.2 字符串型	24
2.1.3 日期型	24
2.1.4 逻辑型	24
2.1.5 对象型	25
2.2 常量和变量	25
2.2.1 常量	25
2.2.2 变量	26

2.3 运算符和表达式	28
2.3.1 算术运算符与算术表达式	29
2.3.2 字符串运算符与字符串表达式	30
2.3.3 关系运算符与关系表达式	30
2.3.4 逻辑运算符与逻辑表达式	31
2.3.5 运算符的优先级	32
2.3.6 表达式的书写规则	33
2.4 常用内部函数	33
2.4.1 数学函数	33
2.4.2 字符串函数	34
2.4.3 转换函数	35
2.4.4 日期时间函数	36
2.4.5 Format 格式输出函数	37
2.5 综合应用	38
习题	41

第 3 章 程序的基本控制结构

3.1 VB.NET 结构化程序设计基础	42
3.1.1 基本控制结构	42
3.1.2 程序的结构化	43
3.2 顺序结构	44
3.2.1 赋值语句	44
3.2.2 用户交互函数	46
3.3 选择结构	48
3.3.1 IF 语句	48
3.3.2 Select Case 语句	51
3.4 循环结构程序设计	53
3.4.1 For·Next 循环	53
3.4.2 For Each·Next 循环	55
3.4.3 Do·Loop 循环	56
3.4.4 多重循环	58
习题	61

第 4 章 数组

4.1 数组的概念	66
-----------------	----

4.2 数组的声明和使用.....	67	第 6 章 用户界面设计.....	118
4.2.1 数组的声明.....	67	6.1 RadioButton.....	118
4.2.2 数组的初始化.....	69	6.2 CheckBox.....	119
4.2.3 数组的使用.....	70	6.3 ListBox.....	120
4.3 数组的方法与属性.....	71	6.4 CheckedListBox.....	125
4.4 数组的基本操作.....	72	6.5 ComboBox.....	126
4.4.1 数组的赋值操作.....	72	6.6 GroupBox 和 Panel.....	128
4.4.2 数组的输出.....	73	6.6.1 GroupBox.....	129
4.4.3 数组的转置.....	74	6.6.2 Panel.....	131
4.4.4 求最大值、最小值及位置.....	75	6.7 HScrollBar 和 VScrollBar.....	131
4.4.5 数组元素的倒置.....	76	6.8 ProgressBar.....	133
4.4.6 数组排序.....	77	6.9 PictureBox.....	134
4.5 数组的动态可调性.....	79	6.10 ImageList.....	137
4.5.1 Dim 语句.....	79	6.11 Timer.....	142
4.5.2 ReDim 语句.....	79	6.12 ErrorProvider.....	144
4.5.3 New 子句.....	80	6.13 综合应用.....	145
4.6 数组列表.....	86	习题.....	148
4.7 综合应用.....	91	第 7 章 通用对话框和菜单.....	149
习题.....	94	7.1 通用对话框.....	149
第 5 章 过程.....	96	7.1.1 OpenFileDialog.....	149
5.1 过程的概念.....	97	7.1.2 SaveFileDialog.....	150
5.2 Sub 过程.....	98	7.1.3 FontDialog.....	151
5.2.1 Sub 过程的声明.....	98	7.1.4 ColorDialog.....	152
5.2.2 Sub 过程的调用.....	100	7.2 菜单.....	154
5.3 Function 过程.....	104	7.2.1 创建下拉式菜单.....	155
5.3.1 Function 过程的声明.....	105	7.2.2 创建弹出式菜单.....	156
5.2.2 Function 过程的调用.....	105	7.3 多重窗体.....	156
5.4 参数传递.....	108	7.3.1 添加窗体.....	157
5.4.1 参数传递的机制.....	109	7.3.2 设置启动窗体.....	157
5.4.2 参数传递的顺序.....	109	7.3.3 窗体的实例化与显示.....	157
5.4.3 参数数组.....	110	7.3.4 不同窗体间的数据访问.....	159
5.4.4 可选参数.....	111	7.4 多文档界面 (MDI).....	159
5.5 变量的特性.....	112	7.4.1 多文档界面的常用属性和方法.....	159
5.5.1 生存期.....	112	7.4.2 创建 MDI 父窗体.....	160
5.5.2 可访问性.....	113	7.4.3 创建 MDI 子窗体.....	161
5.5.3 范围.....	114	7.4.4 排列子窗体.....	162
5.6 综合应用.....	114	7.4.5 键盘与鼠标事件.....	162
习题.....	117		

7.5 综合应用.....	165	9.9.1 Framework 命名空间和 常用的类.....	203
习题.....	167	9.9.2 命名空间.....	204
第 8 章 图形	168	9.9.3 VB.NET Imports 语句.....	205
8.1 GDI+绘图基础知识.....	168	9.9.4 创建自己的命名空间.....	206
8.2 GDI+绘制图形的基本方法.....	169	习题.....	208
8.3 创建画笔、笔刷和字体绘图工具.....	170	第 10 章 建立类库	209
8.3.1 创建画笔工具.....	170	10.1 类库.....	209
8.3.2 创建笔刷工具.....	170	10.1.1 什么是类库.....	209
8.3.3 创建字体工具.....	171	10.1.2 创建类库.....	210
8.4 绘制图形.....	172	10.1.3 在应用程序中使用类库.....	211
8.5 综合应用.....	175	10.2 程序集.....	212
第 9 章 面向对象的编程	177	10.3 使用强名.....	213
9.1 类和对象.....	177	10.3.1 为程序集签名.....	213
9.1.1 对象.....	177	10.3.2 设置程序集版本.....	215
9.1.2 属性、方法和事件.....	177	10.4 注册程序集.....	215
9.1.3 类的概念.....	181	第 11 章 文件	217
9.2 构建类、对象、方法和事件.....	181	11.1 文件概述.....	217
9.2.1 构建类和对象.....	181	11.1.1 文件类型.....	217
9.2.2 构建方法和事件.....	184	11.1.2 文件访问方式.....	218
9.3 构造函数和析构函数.....	185	11.2 run-time 函数.....	218
9.3.1 构造函数.....	185	11.2.1 run-time 文件函数介绍.....	218
9.3.2 析构函数.....	187	11.2.2 使用 run-time 函数读写文件.....	219
9.4 继承.....	188	11.2.3 实例.....	220
9.4.1 继承的概念.....	188	11.3 System.IO 模型.....	221
9.4.2 MyBase、MyClass 和 Me.....	190	11.3.1 File 类.....	221
9.4.3 派生类的构造函数和析构函数.....	190	11.3.2 Directory 类.....	223
9.4.4 重写.....	192	11.3.3 FileStream 类.....	224
9.5 封装性.....	193	11.3.4 StreamReader 类和 StreamWriter 类.....	225
9.6 重载.....	194	11.3.5 BinaryReader 类和 BinaryWriter 类.....	227
9.6.1 方法的重载.....	195	11.4 文件系统对象.....	227
9.6.2 构造函数重载.....	195	11.5 综合应用.....	229
9.6.3 运算符重载.....	196	习题.....	231
9.7 接口.....	199	第 12 章 访问数据库	232
9.8 多态.....	201	12.1 数据库的概念.....	232
9.8.1 基于继承的多态性.....	201		
9.8.2 基于接口的多态.....	202		
9.9 命名空间.....	203		

12.1.1 Microsoft Access 对象	233	13.2 使用调试工具调试程序	254
12.1.2 表	233	13.2.1 设置和删除断点	254
12.1.3 查询	234	13.2.2 跟踪程序的执行	255
12.2 SQL 中的 SELECT 语句	235	13.3 异常处理	255
12.2.1 SELECT 语句格式	235	13.3.1 结构化异常处理	256
12.2.2 SELECT 列表语句	235	13.3.2 非结构化异常处理	257
12.2.3 FROM 子句	236	13.4 综合应用	259
12.2.4 WHERE 子句	236		
12.2.5 ORDER BY 子句	240		
12.3 Access 中的查询	240	第二部分 实验	
12.3.1 利用设计视图创建查询	241	实验 A VB.NET 环境与可视化	
12.3.2 使用 SQL 语句创建查询	243	编程基础	262
12.4 数据访问组件	243	实验 B 语言基础和顺序	
12.4.1 DataSet 组件	243	程序设计	267
12.4.2 DataGridView 组件	244	实验 C 程序的选择结构	270
12.4.3 BindingSource 组件	244	实验 D 程序的循环结构	272
12.4.4 BindingNavigator 组件	245	实验 E 数组	275
12.4.5 TableAdapter 组件	245	实验 F 过程	279
12.5 数据绑定	245	实验 G 用户界面设计	283
12.6 综合应用	246	实验 H 面向对象程序设计	290
习题	251	实验 I 文件	292
第 13 章 调试和错误处理	252	实验 J 数据库应用基础	294
13.1 主要错误类型	252		
13.1.1 语法错误	252		
13.1.2 运行期间错误	253		
13.1.3 语义或逻辑错误	254		

第一部分 Visual Basic.NET 2010 教程

第 1 章 概 论

本章主要介绍 Visual Basic.NET 的基本概念、集成开发环境、窗体和基本控件、程序结构以及基本开发方法。

1.1 Visual Basic.NET 概述

Visual Basic.NET (简称 VB.NET) 是在 BASIC 语言的基础上发展而来的。VB.NET 与 VB 有很大的不同, VB 是基于事件和对象的思想, VB.NET 是基于 .NET 框架, 这是微软公司为了将新的编译环境与潮流接轨所致。 .NET 是当前程序设计的主流体系之一, 代表了程序设计技术发展的方向。

.NET 的核心是 .NET 框架 (.NET Framework), 它是以一种采用系统虚拟机运行的编程平台, 以通用语言运行库 (Common Language Runtime) 为基础, 支持多种语言 (Visual Basic.NET、Visual C#.NET、Visual J#.NET 等) 的开发。从层次结构来看, .NET 框架又包括 3 个主要组成部分: 公共语言运行时 (CLR: Common Language Runtime)、服务框架 (Services Framework) 和上层的两类应用模板, 即传统的 Windows 应用程序模板 (Win Forms) 和基于 ASP.NET 的面向 Web 的网络应用程序模板 (Web Forms 和 Web Services)。

.NET 框架还提供了一个跨语言的统一编程环境和开发工具, 即 Visual Studio.NET (简称 VS.NET), 它将过程的思想完全用 OOP (Object Oriented Programming, 面向对象程序设计) 的思想取代, 是真正面向对象、功能强大的集成开发工具。VB 面向对象的能力远远不能满足计算机网络的需求, 这也是越大项目越少用到 VB 的原因。VB.NET 新增并加强了许多新的面向对象的特征, 比如继承、重载、多态等, 使开发者可以快速地可视化开发网络应用程序、网络服务、Windows 应用程序和服务器端组件, 可以说是一门全新的面向对象语言。

VB.NET 作为 VS.NET 的重要成员之一, 它继承了 VB 语言简单、易学、易用的优点, 更是云计算时代唯一的开发工具与协作管理平台, 它将为软件开发设计带来深远的影响。

为了掌握程序设计的最新技术, 本书以 Visual Studio 2010 (简称 VS 2010) 下的 VB.NET 为平台进行讲解。

1.1.1 第一个 VB.NET 程序

下面通过一个简单的 VB.NET 应用程序，来了解 VB.NET 的基本概念和特点。

例 1.1 简单的字幕跑马灯演示。一匹小木马在一行字幕“学习使用 VB.NET”的映衬下，在标题为“跑马灯游戏”的窗体上按照自右向左或自左向右两边跑动，且该字幕会在不同的方向上有颜色的变化。在程序运行阶段，木马与字幕的行进方向以及与窗体左边界的距离位置会在窗体上动态显示。任务大致介绍如下。

(1) 设计程序界面与设置控件对象属性。界面由窗体对象 Form1 组成，如图 1-1 所示。在工具箱的“所有 Windows 窗体”组项中，分别选择 5 个 Label 控件、1 个 PictureBox 控件和 1 个 Timer 控件，拖动或双击到窗体相应的位置，并设置各个控件的相应属性。程序运行时的初始界面如图 1-2 所示。

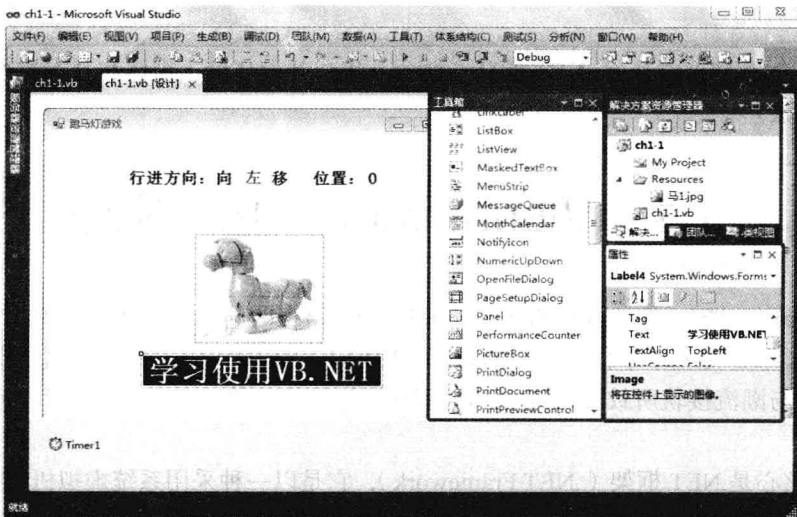


图 1-1 程序设计界面

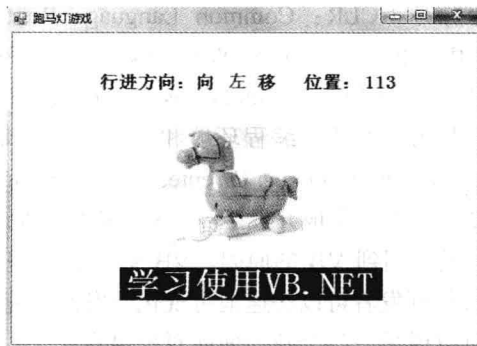


图 1-2 程序运行效果


(2) 编写控件对象的事件过程。在代码窗口中为窗体对象 Form1 编辑 Load(加载)事件过程，为计时器控件对象 Timer1 编辑 Tick()事件过程，各个对象程序代码如下：

```
Public Class Form1
```

```

Dim x As Integer, y As Integer, k As Integer      '走马灯坐标
Dim flag As Boolean = True                      '走马灯初始左移
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) _
Handles MyBase.Load
    x = 150 : y = 150
    Label5.Location = New Point(x - 70, y + 150) '设定文本控件的起始位置
    PictureBox1.Location = New Point(x, y)      '设定图形控件的起始位置
    Label5.AutoSize = True                      '设定文本控件能按数据调整大小
    Timer1.Interval = 10                       '设定定时器周期为 10/1 000=0.01 秒
    Timer1.Enabled = True                      '定时器触发
End Sub
Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) _
Handles Timer1.Tick
    Label4.Text = Label5.Left()
    If flag = True Then                        '左移情况
        x -= 1
        Label5.Location = New Point(x - 70, y + 150)
        PictureBox1.Location = New Point(x, y)
        Label2.Text = "左"
        Label5.BackColor = Color.Blue         '变走马灯字体背景为蓝色
        Label5.ForeColor = Color.FloralWhite '变走马灯字体为白色
        If (Integer.Parse(Label5.Left) <= 0) Then
            flag = False                      '设置右移
        End If
    Else                                       '右移情况
        x += 1
        Label5.Location = New Point(x - 70, y + 150)
        PictureBox1.Location = New Point(x, y)
        Label2.Text = "右"
        Label5.BackColor = Color.FloralWhite '变走马灯字体背景为白色
        Label5.ForeColor = Color.Blue         '变走马灯字体为蓝色
        If (Integer.Parse(Label5.Left) + Integer.Parse(Label5.Width) >= Integer._
Parse(Me.Width)) Then
            flag = True                       '设置左移
        End If
    End If
End Sub
End Class

```

(3) 按“F5”键或单击工具栏中的  图标或在“调试(D)”菜单下单击“启动调试(S)”项, 字幕跑马灯程序开始运行。

1.1.2 基本概念

1. 类和对象

(1) 类

现实世界中, 类(Class)定义了一件事物的抽象特点。在面向对象程序设计中, 类是创建对象实例的模板, 是同种对象的集合与抽象。类包含了所创建对象属性的数据以及对这些数据进行操作的方法定义。

在 VB.NET 中，类可以分为两种：一种由系统预先设定好，用户可以直接使用，如窗体、控件等，即控件类；另一种是由用户自定义的类。

类的定义格式：

```
[访问符] Class 类名
    \类体
End Class
```

其中，“Class”是用于声明类的关键字，必须以“End Class”关键字进行结束。在类体中可以定义类的字段、属性、方法和事件。

在“解决方案资源管理器”窗口中，右键单击项目，在弹出的快捷菜单中选择“添加”→“类”命令，并在弹出的“添加新项”对话框中输入类的名字，即可以创建一个用户自定义的类，如图 1-3 所示。

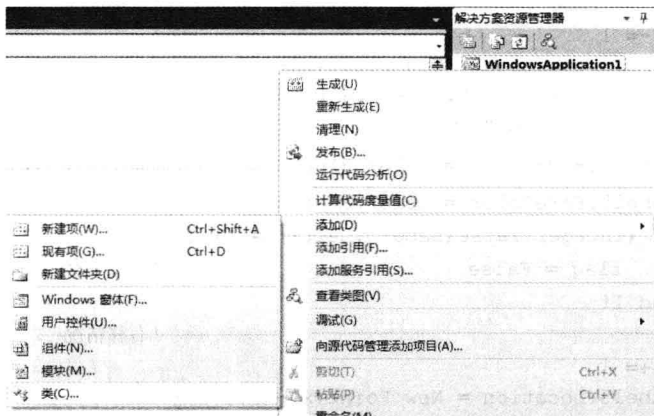


图 1-3 创建一个用户自定义类

(2) 对象

对象 (Object) 是面向对象的程序设计思想的核心概念之一，是类的实例，其定义格式为：

```
Dim 对象名 As 类名 [=New 类名 ()]
```

其中，类名是系统预定义或用户自定义的类。可以先实例化一个对象，在程序中为该对象赋值；也可以在实例化的同时，使用 New 关键字调用类的构造函数创建一个实例，为对象赋值，如 [=New 类名()]。

一般来说对象都有自己的特征、行为和发生在该对象上的活动。如某一辆车，这个对象具有颜色、重量、油耗等特征，并具有开动、行驶、载人载物等行为。当外界作用其上时，它产生相应的活动，如上坡加油，下坡刹车等。

在面向对象程序设计中，我们将对象的特征称为字段或属性，对象的行为称为方法，对象的活动称为事件。

2. 对象的字段、属性、方法和事件

(1) 对象的字段、属性

字段 (Fields) 和属性 (Property) 都是类用于保存数据的成员，它们的区别在于，字段只是类的简单变量，是一个对象含有的片断的信息；而属性是对象的特征，当描述一个对象的特征时，就是在描述该对象的属性，对象的每一个属性描述了对象的每一个特定面。属性被描述成名|值

对，即每一个属性都具有属性名和对应的属性值，如：颜色——红色，高度——800 像素。

在 VB.NET 中，每一个控件对象都具有许多属性，例如，控件名称（Name）、标题（Text）、颜色（Color）、字体（Font）等。不同的对象具有各自不同的属性。

对象属性值的设定可以通过两种方式来实现。

- ① 设计阶段：利用属性窗口直接设置对象的属性值。
- ② 运行阶段：利用赋值语句实现，格式为：

对象名.属性名=属性值

在设计阶段，对象的属性都显示在属性窗口中。要通过属性窗口修改对象的属性值，必须先选择要设置属性的对象。选择之后，属性窗口中就列出了该对象的全部属性；选中要修改的属性名，在右边的单元格中直接输入或设置其属性值，如图 1-4 所示。

如果在程序运行过程中，对象的属性值发生了动态的变化，则需要程序代码中修改对象的属性值。例如，一个名为 Button1 的命令按钮，按钮上显示的文本修改为“再次确认”，可以在程序中相应的位置加入代码：

```
Button1.Text = "再次确认"
```

在程序代码中，如果使用同一个对象的多个属性时，可以逐一设置多个属性，也可以使用 With...End With 语句，节约代码的书写。其语法格式如下：

```
With <对象名>
<语句组>
End With
```

例 1.2 需要在程序代码中设置文本框 TextBox1 的属性：宽度（Width）、文本值（Text）、字体颜色（ForeColor）。逐行书写代码如下：

```
TextBox1.Width = 200
TextBox1.Text = "我的文本框"
TextBox1.ForeColor = Color.Blue
```

使用 With...End With 语句可以实现相同的功能。其代码如下：

```
With TextBox1
.Width = 200
.Text = "我的文本框"
.ForeColor = Color.Blue
End With
```

注意：对象的大部分属性是可读|写的，即既可以在设计阶段也可以在程序运行阶段设置。如果属性只能在设计阶段设置，在程序运行阶段不可以改变，这样的属性称为只读属性，如名称（Name）、最大化（MaxButton）、最小化（MinButton）等。

（2）对象的方法

方法（Method）是对象能够执行的动作或功能，每一种对象都有其特定的方法。面向对象的程序设计语言为程序设计人员提供了一种特殊的过程供用户直接调用，这就是方法。它是对象本身内含的函数和过程，用于完成某种特定的功能。

实现方法的具体代码是不可见的。用户在编写代码时，通过方法名调用对应方法的程序代码，



图 1-4 属性面板

对象将按照顺序执行一系列动作。方法的调用格式为：

[对象名.]方法 [实参表]

其中，[对象名.]表示如果是当前窗体，则可以省略。

例如：

```
Me.Hide()           '隐藏当前窗体
MainForm.Show()    '显示名为 MainForm 的窗体
MainForm.TextBox1.Focus() '将窗体 MainForm 上的控件 TextBox1 设为焦点
```

(3) 对象的事件

事件 (Event) 是发生在对象上的活动。同一个事件，作用于不同的对象，会引发不同的反应，产生不同的结果。VB.NET 中的事件是由系统预先定义的、能够为对象和控件所识别的动作，如鼠标单击 (Click)、鼠标双击 (DoubleClick)、获得焦点 (GotFocus)、按下键盘键 (KeyPress) 等。

在对象的各要素中，字段、属性和方法是对象的用户和对象连接的途径，而事件则是对象和程序连接的途径。

3. 事件过程和事件驱动

(1) 事件过程

当事件被触发，也就是在对象上发生了事件后，相应的对象就会对该事件做出处理。处理的步骤就是事件过程。

VB.NET 中事件过程的形式如下：

```
Private Sub 对象名_事件名 [(参数列表)]
    事件过程代码
    .....
End Sub
```

其中，

对象名：Name 属性。

事件名：VB.NET 中预先定义好的能被该对象识别的事件。

事件过程代码：处理该事件的程序。

例 1.3 计算机 D 盘上文件夹 vb 内有图片文件 tulips.jpg，通过下列事件过程加载图片。

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As _ System.EventArgs)
Handles Button1.Click
    Me.Text = "装载图片"
    PictureBox1.SizeMode = PictureBoxSizeMode.StretchImage
    PictureBox1.Image = Image.FromFile("d:\vb\tulips.jpg")
End Sub
```

运行后单击命令按钮，结果如图 1-5 所示。

(2) 事件驱动

VB.NET 中采用了事件驱动的编程机制，程序执行的次序与程序设计者无关，而是取决于用户的具体操作。

程序的执行步骤如下：

- ① 启动应用程序，装载和显示窗体；
- ② 窗体或控件等待事件的发生，直至退出；



图 1-5 例 1.3 的运行效果

③ 事件被触发，执行相应的事件过程，执行完后，返回步骤②；

④ 退出。

事件的触发可以有3种形式，第1种是由用户触发，如 Click、DbClick 等事件；第2种是由系统触发，如 Timer 事件；第3种是由代码间接触发，如加载窗体的 Load 事件等。

程序开发只需要编写相应用户事件的动作代码，且各个事件不一定存在着关联，因而无需关心执行的具体过程。用户对这些事件驱动的顺序决定了代码的执行顺序。

对象、事件和事件过程间的关系如图 1-6 所示。



图 1-6 对象、事件和事件过程之间的关系

1.2 VB.NET 集成开发环境

Visual Studio 产品系列共用一个集成开发环境 (Integrated Development Environment, IDE)，该环境由应用程序设计、编辑、运行、调试、资源管理等多种界面元素组成，是所见即所得的开发工具。

启动后的 Visual Studio 2010 窗口如图 1-7 所示，该窗口可完成新建项目、打开项目等操作。单击“新建项目”选项后，进入如图 1-8 所示“新建项目”对话框，在该窗体中间的列表中列出了创建某种类型的应用程序开发项，表 1.1 列出了部分选项的应用说明。例如，创建一个 Windows 应用程序，双击“Windows 窗体应用程序”即可。

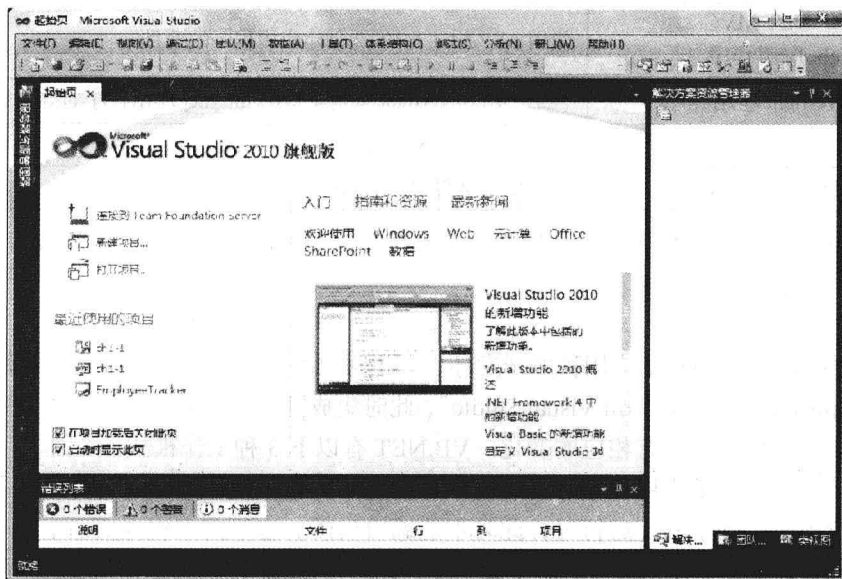










图 1-7 Visual Studio 2010 起始页



图 1-8 新建项目

表 1.1

“新建项目”部分选项

图标	选项	说明
	Windows 窗体应用程序	用于创建具有 Windows 用户界面的应用程序的项目
	WPF 应用程序	Windows Presentation Foundation 客户端应用程序
	控制台应用程序	用于创建控制台应用程序
	ASP.NET Web 应用程序	用于创建具有 Web 用户界面的应用程序的项目
	类库	用于创建 VB 类库的项目 (.dll)
	ASP.NET MVC 2 Web 应用程序	使用 ASP.NET MVC 2 创建应用程序的项目
	Silverlight 应用程序	一个使用 Silverlight 创建丰富的 Internet 应用程序的空项目
	Silverlight 类库	一个用于创建 Silverlight 类库的项目
	WPF 服务应用程序	用于创建 WPF 服务的项目

下面以 Windows 应用程序开发环境为例，介绍与之相关的各种界面元素组成。

1.2.1 主窗口

图 1-9 所示为 Windows 应用程序开发环境，也称为主窗口。窗体顶端的标题栏中的标题为“WindowsApplication1 - Microsoft Visual Studio”，此时集成开发环境处于设计模式状态。进入其他工作模式状态后，标题栏会有相应的变化。VB.NET 有以下 3 种工作模式。

- (1) 设计模式：处于应用程序开发时用户界面设计（包括代码编辑）。
- (2) 运行模式：处于程序运行时，此时不能编辑代码，也不能编辑界面。
- (3) 调试模式：处于程序运行暂时中断状态，此时可以编辑代码，但不能编辑界面。按“F5”键或单击“启动调试”按钮，继续运行程序；单击“停止调试”按钮，停止程序运行。在此模式

下会弹出“自动窗口”和“错误列表”窗口，在“自动窗口”中可编辑值，并立即执行；在“错误列表”窗口中可列出出现的错误信息。

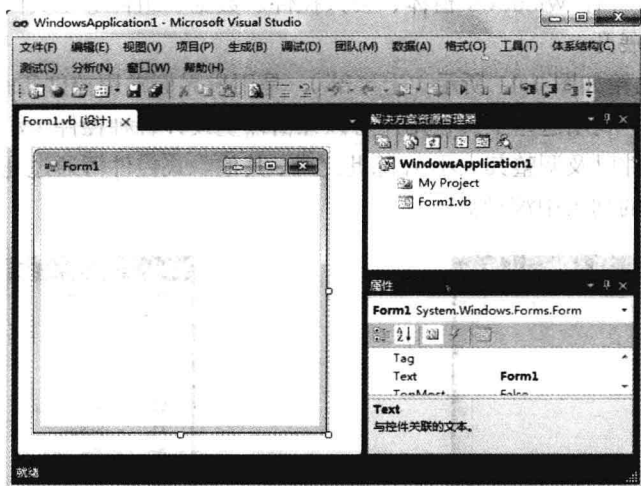


图 1-9 主窗口

1.2.2 菜单栏和工具箱窗口

1. 菜单栏

用户可以通过单击菜单中的菜单项，来实现应用程序所提供的功能。

在“设计”模式下，VS 2010 菜单栏中包含了 14 个下拉菜单，涵盖了 VB.NET 的所有功能，具体名称和主要功能如表 1.2 所示。

表 1.2 各项菜单功能

菜单名称	功能说明
文件 (F)	用于文件、项目及系统相关的各种操作，如新建、打开、保存、退出等
编辑 (E)	用于文件、控件及代码编辑相关的各种操作，如剪切、复制、粘贴、查找等
视图 (V)	管理工作区的各种窗口及工具，如设计器、工具箱、工具栏、属性窗口等
项目 (P)	用于在当前项目中添加各种组件操作，如添加窗体、控件、组件、类、模块等
生成 (B)	用于生成和重新生成以及发布项目等操作
调试 (D)	用于代码的调试与设置，如设置断点、逐语句、逐过程、异常等调试
团队 (M)	用于项目之间进行协同开发，如自动检测、虚拟部署等
数据 (A)	用于对数据库中数据源的相关操作，如添加和显示数据源、添加查询等
格式 (O)	用于与格式相关的操作，如对齐、水平间距、垂直间距等
工具 (T)	提供各种工具的使用，如连接到数据库、服务器及设置集成开发环境选项等
体系结构 (C)	包含建模和可视化工具中引入的新图类型，如 UML 图、DGML 等
测试 (S)	用于完成与项目测试相关的操作，如新建测试、创建新测试列表等
分析 (N)	用于性能分析，如启动性能向导、比较性能报告和探查器等
窗口 (W)	用于设置窗口布局的各种操作，显示当前打开的窗口列表等
帮助 (H)	包含获取帮助信息的相关命令，如查找帮助、管理帮助设置、MSDN 论坛等