

21世纪应用型本科通用教材
上海市教育委员会高校重点教材建设项目



可视化程序设计.NET教程

Visual Basic.NET

向珏良 主编



上海交通大学出版社
SHANGHAI JIAO TONG UNIVERSITY PRESS

21 世纪应用型本科通用教材
上海市教育委员会高校重点教材建设项目

可视化程序设计.NET 教程

Visual Basic.NET

向珏良 主编

上海交通大学出版社

内 容 提 要

本书从“案例”入手,对可视化程序设计进行了详细的介绍。书中由浅入深地介绍了可视化程序设计的集成开发环境、可视化控件用户界面布局、代码编写的基本构架、常用事件和方法、GDI+图形处理和文件系统等,并以大量实例和代码,通过设计分析环节解剖了解题思路 and 关键步骤,便于读者快速上手和理解。全书共分10章,以VB.NET为工具,由用户界面可视化设计、面向对象的代码设计、数组和过程设计、图形和文件系统等模块以及15个专题实验和思考题组成。读者或教师可以根据实际情况灵活组合这些模块和专题实验作为高等学校的程序设计课程教材,也可作为工程技术人员、程序设计爱好者的自学参考书。

图书在版编目(CIP)数据

可视化程序设计. NET 教程/向珏良主编. —上海:
上海交通大学出版社,2013

ISBN 978-7-313-09438-4

I. 可... II. 向... III. 网页制作工具—程序设计—高等学校—教材 IV. TP393.092

中国版本图书馆 CIP 数据核字(2013)第 015775 号

可视化程序设计. NET 教程

向珏良 主编

上海交通大学出版社出版发行

(上海市番禺路 951 号 邮政编码 200030)

电话:64071208 出版人:韩建民

上海华业装璜印刷有限公司 印刷 全国新华书店经销

开本:787mm×1092mm 1/16 印张:15 字数:365千字

2013年1月第1版 2013年1月第1次印刷

印数:1~3030

ISBN 978-7-313-09438-4/TP 定价:34.00元

版权所有 侵权必究

告读者:如发现本书有印装质量问题请与印刷厂质量科联系
联系电话:021-63812710

前 言

本书章节按照高校课程“程序设计基础”大纲进行了周密布局,围绕讲课和专题实验,以“案例”引导入手,以“范例”综合归纳,每章附有专题实验内容和思考题,还配备了“拓展学习”以供学有余力的读者进行深度修炼,由浅入深地带领读者在学习程序设计的原野上自由驰骋。在 Windows 下学会可视化用户界面程序设计方法,深刻领会面向对象的程序设计方法,掌握程序代码构架和事件驱动的设计方法,以及学习 GDI+ 的图形设计方法和文件系统处理方法。

本书以 VB.NET 为开发语言工具,介绍了 Windows 桌面应用程序开发的大量实例,循序渐进地讲解了这些实例的设计分析思路 and 关键步骤。可视化设计借用了大量已经“对象化”的控件轻松地实现拖放式直观布局和静态设计;同时,也可采用代码化、个性化自定义设计对象和方法,解决一些比较专业的问题。经过本书的学习和专题实验,相信读者有能力可以轻松快捷地再去学习 C#.NET 和其他面向对象的程序设计语言。学好本书的程序设计方法以及 Visual Studio 平台,也是进一步学习 Web 开发技术的前奏。

本书共分 10 章以及 15 个配套的专题实验,它们的安排完全根据教学内容和实验环节步步紧扣。第 1 章:介绍 VB.NET 开发环境,快速上手,感悟编程的快乐(实验 1)。第 2 章:运用基本控件的可视化操作和简单事件响应,体验 Windows 风格编程的豪爽(实验 2)。第 3 章:了解 VB.NET 的基本语法和书写规则,万丈高楼还得打好基础,练习简单编码的程序设计方法(实验 3)。第 4 章:掌握程序代码的结构化框架,控制代码流的进程,学会调试和排错,完成具有分支结构、循环结构的多个实验(实验 4 分支、实验 5 计数型循环、实验 6 循环应用)。第 5 章:学会处理有序批量数据的数组方法来进行程序设计,尤其熟练掌握一维数组的基本处理方法和常用的排序、插入、删除、级数求和等算法,完成数组元素遍历、排序、交换、统计、集合、矩阵等实验(实验 7、实验 8)。还可深入学习动态数组、结构类型数组和控件数组等。第 6 章:学会能简化程序代码的过程方法来进行程序设计。分清函数过程和子过程的不同应用场合,弄懂参数传值和传地址的区别,完成过程及应用的实验(实验 9、实验 10)。还可深度学习递归、多线程等复杂编程。第 7 章:运用高级控件,学习丰富的用户界面设计。学会图片框、定时器、动画、对话框、多窗体、菜单等要素的程序设计。完成用户交互的多个实验(实验 11、实验 12)。第 8 章:熟悉面向对象的代码设计方法,深刻理解面向对象模式的理念,学会基本的设计方法(实验 13)。第 9 章:掌握 GDI+ 的绘图方法,学会函数曲线绘制、图案绘制等技能

(实验 14)。第 10 章:学会文件的传统处理方法和文件流处理方法,掌握顺序文件的读写(实验 15)。

全书按 64 学时布局,教学与实验 1:1 分配。当然,也可以根据实际情况灵活选择模块进行组合布局。讲课和实验中要注重问题分析的重要性,参考运行界面,读懂代码以及代码边的大量注释。

本书第 1 章和第 9 章由向珏良编写,第 2~4 章由周晶编写,第 5 章由王泽杰编写,第 6 章由张晓梅编写,第 7~8 章由刘惠彬编写,第 10 章由胡浩民编写。在编写过程中还得到了伍建青、赵毅、赵卫东、王明衍、陈强、潘勇、黄容、洪菁、胡建鹏、郁尚奇、章颖芳和实验室老师的帮助。最后还要感谢王裕明教授、汤彬高级实验师对本书提出了许多建议。同时,还要感谢上海交通大学出版社的编辑们为此书进行了精心的策划和仔细的编辑。由于作者水平有限,书中存在的疏漏和问题,恳请专家和读者给予指正和建议。

作 者

2012 年 11 月于上海工程技术大学

目 录

第 1 章 熟悉 VB.NET 的可视化编程环境	1	第 2 章 运用基本控件进行可视化程序设计	21
1.1 新手上路	1	2.1 认识基本控件	21
1.1.1 轻松完成第一个程序	1	2.1.1 控件与可视化对象	21
1.1.2 回头看看已经学会了什么	3	2.1.2 程序设计中对象的三要素	22
1.1.3 VB.NET 的特点	4	2.2 基本控件	23
1.2 熟悉程序设计的集成环境	4	2.2.1 窗体	23
1.2.1 建立 VB.NET 解决方案	5	2.2.2 标签	27
1.2.2 集成环境窗口的标题栏、 菜单栏和工具栏	6	2.2.3 命令按钮	28
1.2.3 对象的可视化布局窗口	6	2.2.4 文本框	29
1.2.4 代码编程窗口	8	2.2.5 复选框	31
1.2.5 解决方案资源管理器窗口	8	2.2.6 单选按钮与分组框 Group Box	31
1.2.6 其他窗口	8	2.3 拓展学习	33
1.2.7 窗口的布置	9	2.3.1 更多属性、事件和方法的 学习	33
1.2.8 打开已经存在的项目	9	2.3.2 共享事件的处理	35
1.3 学会 VB.NET 可视化程序设计的方法	10	2.4 基本控件的程序设计实验	36
1.3.1 程序设计三步曲	10	2.4.1 实验范例	36
1.3.2 VB.NET 项目树结构与编码 规则	11	2.4.2 实验二 运用基本控件的程序 设计	40
1.4 拓展学习	14	思考题	41
1.4.1 了解 .NET 框架	14	第 3 章 学习 VB.NET 编程的基础 知识	43
1.4.2 学会无处不在的帮助	15	3.1 数据类型	43
1.5 熟悉 VB.NET 集成环境的程序 设计实验	17	3.1.1 基本数据类型	43
1.5.1 实验范例	17	3.1.2 标识符	45
1.5.2 实验一 熟悉 VB.NET 集成 环境的程序设计	18	3.2 变量与常量	45
思考题	20	3.2.1 常量	46

3.2.2 变量	46	4.5.2 异常处理	77
3.3 运算符和表达式	48	4.6 具有控制结构的程序设计实验 ...	80
3.3.1 运算符	48	4.6.1 实验范例	80
3.3.2 表达式	49	4.6.2 实验四 运用分支结构的程序 设计	83
3.4 常用函数	49	4.6.3 实验五 运用计数型循环的 程序设计	84
3.4.1 数学函数	50	4.6.4 实验六 循环应用的程序 设计	85
3.4.2 类型转换函数	50	思考题	86
3.4.3 字符串函数	51	第 5 章 学会数组在有序批量数据中的 应用	88
3.4.4 日期和时间函数	52	5.1 数组的概念	88
3.4.5 随机函数	52	5.2 数组的声明和引用	88
3.5 拓展学习	52	5.2.1 一维数组的声明和初始化 ...	88
3.5.1 命名空间的作用	52	5.2.2 一维数组的基本处理	90
3.5.2 命名空间的使用	54	5.2.3 多维数组	91
3.6 简单编码的程序设计实验	54	5.2.4 动态数组	92
3.6.1 实验范例	54	5.3 结构数据类型与结构数组	94
3.6.2 实验三 简单编码的程序 设计	57	5.3.1 结构数据类型	94
思考题	58	5.3.2 结构数组	96
第 4 章 掌握程序设计的结构控制	60	5.4 数组处理的常用算法	97
4.1 程序结构和代码书写规则	60	5.4.1 排序	97
4.1.1 程序结构	60	5.4.2 插入、删除元素	101
4.1.2 代码书写规则	61	5.5 拓展学习	103
4.2 顺序结构	61	5.5.1 使用 Array 类中的方法处理 数组	103
4.2.1 赋值语句	61	5.5.2 处理控件数组	104
4.2.2 数据输入和数据输出	62	5.6 数组的程序设计实验	105
4.3 选择结构	64	5.6.1 实验范例	105
4.3.1 If 条件语句	65	5.6.2 实验七 数组基本操作的程序 设计	107
4.3.2 Select Case 语句	67	5.6.3 实验八 数组应用的程序 设计	108
4.3.3 选择结构的嵌套	69	思考题	110
4.4 循环结构	69	第 6 章 学会过程在程序结构简化中的 应用	112
4.4.1 计数型循环 For...Next 循环 语句	69	6.1 函数过程的定义和调用	112
4.4.2 条件型循环 Do...Loop 循环 语句	70		
4.4.3 循环结构的三要素	72		
4.4.4 循环结构的嵌套	73		
4.5 拓展学习	76		
4.5.1 条件函数	76		

6.1.1 函数过程对程序的简化·····	112	7.3.4 窗体间的数据传递和引用·····	144
6.1.2 建立函数过程·····	113	7.3.5 多窗体的综合应用·····	144
6.1.3 调用函数过程·····	114	7.4 菜单设计·····	145
6.2 子过程的定义和调用·····	115	7.4.1 下拉菜单·····	145
6.2.1 子过程对程序的简化·····	115	7.4.2 快捷菜单·····	146
6.2.2 建立子过程·····	115	7.5 拓展学习·····	147
6.2.3 调用子过程·····	116	7.5.1 鼠标和键盘·····	147
6.3 参数传递·····	117	7.5.2 更多通用对话框·····	150
6.3.1 形参与实参·····	117	7.6 具有丰富界面的程序设计实验·····	151
6.3.2 传值与传地址·····	117	7.6.1 实验范例·····	151
6.3.3 数组作为参数传递·····	119	7.6.2 实验十一 列表框和动画程序设计·····	155
6.4 变量的作用域·····	120	7.6.3 实验十二 菜单驱动的程序设计·····	157
6.4.1 变量的作用域·····	120	思考题·····	159
6.4.2 静态变量·····	122	第8章 熟悉面向对象的程序设计方法 ·····	160
6.5 拓展学习·····	123	8.1 面向对象的相关概念·····	160
6.5.1 递归·····	123	8.1.1 面向对象技术·····	160
6.5.2 多线程·····	124	8.1.2 类和对象·····	160
6.6 过程的程序设计实验·····	127	8.1.3 类的基本特性·····	161
6.6.1 实验范例·····	127	8.2 创建类和对象·····	162
6.6.2 实验九 自定义函数的程序设计·····	129	8.2.1 定义类·····	162
6.6.3 实验十 子过程应用的程序设计·····	130	8.2.2 定义数据成员·····	163
思考题·····	132	8.2.3 定义属性·····	164
第7章 设计漂亮的用户界面与事件响应 ·····	133	8.2.4 定义方法·····	164
7.1 常用控件·····	133	8.2.5 定义事件·····	165
7.1.1 列表框和组合框·····	133	8.2.6 定义构造函数和析构函数·····	166
7.1.2 滚动条和进度条·····	136	8.2.7 对象及其成员的访问·····	166
7.1.3 图片框·····	137	8.3 继承和派生·····	168
7.1.4 定时器·····	139	8.3.1 继承的概念·····	168
7.2 通用对话框·····	140	8.3.2 继承的实现·····	169
7.2.1 “颜色”对话框·····	140	8.3.3 派生类的构造函数·····	169
7.2.2 “字体”对话框·····	141	8.4 拓展学习·····	170
7.3 多窗体程序设计·····	142	8.4.1 接口·····	170
7.3.1 添加窗体·····	143		
7.3.2 窗体的显示和隐藏·····	143		
7.3.3 设置启动窗体·····	143		

8.4.2 多态.....	172	9.5.1 坐标变换与坐标系变换.....	192
8.5 面向对象的程序设计实验.....	174	9.5.2 窗体造型.....	194
8.5.1 实验范例.....	174	9.6 绘制图形的程序设计实验.....	195
8.5.2 实验十三 面向对象的程序 设计.....	175	9.6.1 实验范例.....	195
思考题	177	9.6.2 实验十四 绘制图形的程序 设计.....	199
第9章 掌握图形应用程序设计	178	思考题	202
9.1 在画布上绘制第一个图形.....	178	第10章 学会文件在程序设计中的 使用.....	204
9.1.1 画图:画布、画笔、绘图 方法.....	178	10.1 文件概述	204
9.1.2 书写:画布、画刷与字体、 书写方法.....	179	10.1.1 文件类型	204
9.2 GDI+绘图基础	179	10.1.2 访问文件的方式	206
9.2.1 GDI+绘图的基本类和 画布.....	180	10.1.3 文件操作三步曲	206
9.2.2 GDI+绘图的相关对象	180	10.2 直接文件访问	206
9.2.3 画布的默认坐标系与 变换.....	181	10.2.1 顺序文件访问	206
9.3 在画布上绘图与书写.....	182	10.2.2 随机文件访问	210
9.3.1 画笔与绘图方法.....	182	10.2.3 二进制文件访问	213
9.3.2 画刷与填充方法.....	185	10.3 拓展学习	215
9.3.3 字体与书写方法.....	187	10.3.1 System. IO 访问方式	215
9.4 绘制函数图形.....	188	10.3.2 My. Computer. FileSystem	217
9.4.1 绘制 $y=f(x)$ 函数图形	188	10.4 运用文件的程序设计实验	219
9.4.2 绘制参数方程曲线.....	191	10.4.1 实验范例	219
9.5 拓展学习.....	192	10.4.2 实验十五 运用文件的程序 设计	223
		思考题	227

第 1 章 熟悉 VB.NET 的可视化编程环境

Visual Basic .NET(简称 VB.NET)以微软 .NET 框架为核心,具有 Visual Studio .NET 统一风格的集成开发环境。本章从案例入手,全面展示 VB.NET 的对象可视化布局与设计、事件驱动与代码编写、调试与运行的集成环境,掌握 VB.NET 程序设计的方法和步骤。

1.1 新手上路

新手上路,先试着编写第一个 VB.NET 程序,顺便熟悉一下集成环境中的各种工具、命令、窗口。它们在哪里?长什么样?怎么使用?

1.1.1 轻松完成第一个程序

例题 1.1 在 300×200 的窗体中央放置一个 20×20 的红色小球,窗体下方放置一组方向按钮,右上角动态显示小球当前的位置,程序运行界面如图 1.1 所示。

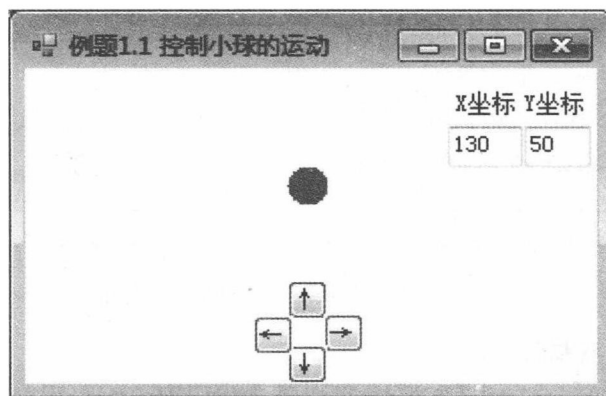


图 1.1 控制小球的运动

1. 对象的可视化布局与设计

进入 VB.NET 集成环境后,自动会在“Form1.vb[设计]”选项卡上建立一个窗体对象 Form1(见图 1.2),借助于左边的“工具箱”分别找到按钮(Button)、标签(Label)、文本框(TextBox)和椭圆图形(OvalShape)的控件(这里的控件也称对象),选中这些控件先后拖曳至窗体 Form1 上。选中某个控件后会出现 8 控制点样式,这时可以可视化地移动该控件进行界面布局,以及改变其大小。然后按表 1.1 在右边“属性窗”精细地数字化进行控件的属性设置。

表 1.1 “控制小球的运动”中控件的主要属性

控件对象	属性	属性值			属性	属性值		
Form1	Text	例题 1.1 控制小球的运动			Size	300, 200		
Button1	Text	↑			Size	20, 20		
Label1	Text	X 坐标			AutoSize	True		
TextBox1	Size	37, 21			Location	210, 30		
OvalShape1	Size	20, 20	BackColor	Red	BackStyle	Opaque	BorderColor	Transparent

Button2~Button4、Label2、TextBox2 可以仿造上述控件完成相应属性设置,如图 1.2 所示。

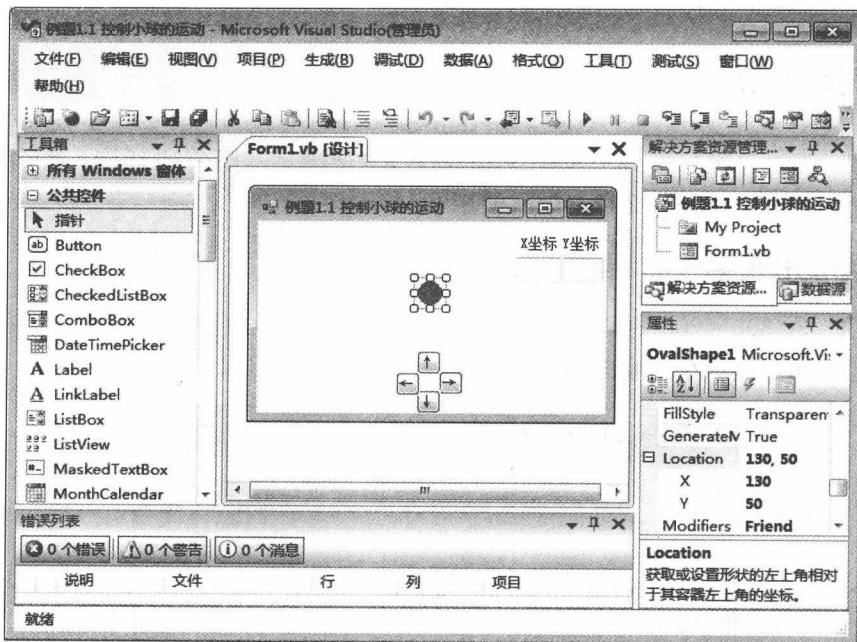



图 1.2 VB.NET 集成环境与对象可视化布局

2. 对象的事件代码编写

双击对象(或单击“Form1.vb”选项卡)可以进入代码窗口编写程序代码,如图 1.3 所示。窗口上方左边下拉框可以选择控件对象(比如 Button1),右边下拉框可以选择该控件对象的事件(比如 Click)。

3. 保存解决方案与调试运行程序

执行“文件/全部保存”菜单命令,系统会弹出保存项目对话框。在指定文件夹下,输入项目名称,系统自动建立与项目同名的解决方案。

完成界面设计和代码编写后,单击工具栏上的  按钮,就可立即运行 Windows 程序了。通过单击方向按钮可以控制小球的运动方向,同时小球的坐标当即刷新。如果发生错误,程序停止运行,首先看清楚出错的原因提示,然后将鼠标悬停在相关变量上,系统会显示该变量的

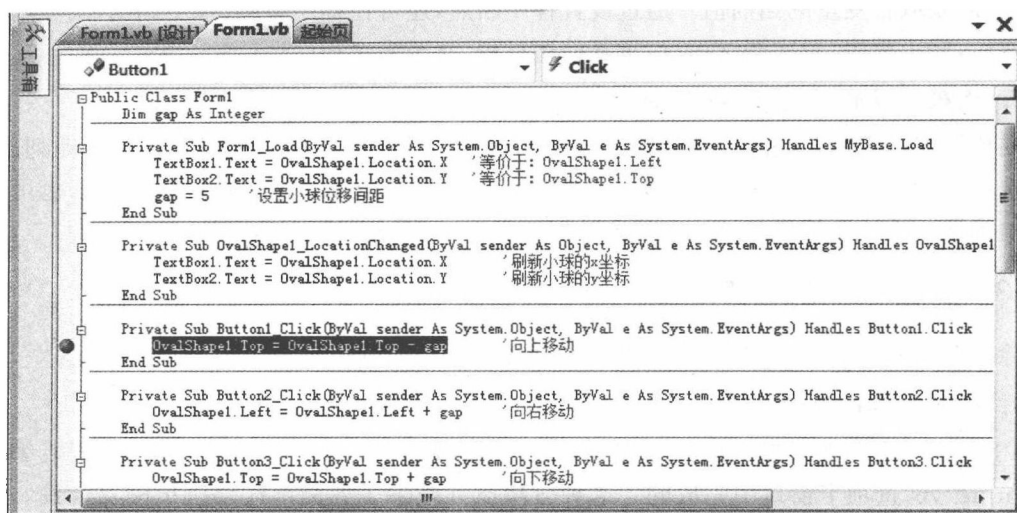


图 1.3 “代码编程窗口”的程序代码

当前值,从而推断引起出错的可能原因。当然,也可以设置断点(在要暂停行的左边栏上单击一下,会显示如图 1.3 的小圆点;再次单击该小圆点,可以取消断点)暂停程序运行,或单步跟踪运行。修正程序后,重新运行程序,直至结果正确。

经过上述 VB.NET 的编程三步曲,祝贺您已经轻松完成了第一个 Windows 下的 VB.NET 程序。

1.1.2 回头看看已经学会了什么

通过例题 1.1 我们已经学会了:

1. 在集成编程环境中进退自如

在 Visual Studio.NET 统一风格的编程集成环境中,可视化设计用户界面、编写代码和调试运行程序。三步联动,一气呵成,轻松编程。


2. 直观的可视化设计方法

VB.NET 的工具箱中,预制了大量各具特色的现成对象集合(也称类)。在用户界面设计时,只要将某个类(也称为控件类)拖曳入窗体,就形成了一个可视的实例对象(以下简称某控件)。选中该控件,就可以可视化地拖曳定位、改变大小。

3. 事件触发与程序响应的联动(事件驱动)

在 Windows 窗体上,对于对象的某种动作(如单击按钮对象触发的 Click 事件),可以引起该事件程序代码的运行(比如 Click 代码段响应)。程序的运行结果取决于事件的先后序列。程序设计人员只需考虑对象对于特定事件的响应,而不必考虑程序状态的全盘控制,简化了程序过程的复杂性。

4. 调试运行和编辑一体化

在程序设计的任何一个阶段,单击  按钮就可运行程序。程序一旦出错,现场状态完全

保留,可以获取各变量的当前值。通过设置程序断点,还可让程序暂停、单步调试,细究出错原因。然后,修正程序,再次调试运行,重复前述过程,直至满意程序运行的结果。

5. 代码编写的智能感知和格式对齐

在代码窗口编写代码时,只要键入对象名的前几个字母,候选对象按字典序自动列出,等待候选;选定对象后键入后续字符(如“.”),还会显示众多候选属性;可以双击选定,也可以键入空格确认。系统还提供了结构化代码的智能对齐功能,使代码层次逻辑非常清晰。

1.1.3 VB.NET 的特点

除了前面我们已经看到的那些轻松编程特点外,VB.NET 还具有以下主要特点:

1. 代码与界面设计分离

VB.NET 的可视化界面设计是在“Form1.vb[设计]”选项卡窗口中进行的,代码编写是在“Form1.vb”选项卡窗口中完成的。这种分离设计减少了界面与代码维护的复杂性。它们虽然分离,设计时互不影响,但是通过对象以及对象的事件会发生响应和关联。

2. 丰富的数据类型和强大的类库

VB.NET 不仅具有丰富的基本数据类型和构造数据类型,还可以自定义具有灵巧的数据处理能力的数据类。依托.NET 框架的支持,VB.NET 可以获取强大的 Windows 类库。

3. 面向对象的事件驱动和结构化程序设计

面向对象的程序设计方法(Object-Oriented Programming,简称 OOP)具有事件驱动的明显特征,封装的对象还具有灵活多样的继承性、多态性和重载。同时,在具体的事件编码过程中,又提供了标准的结构化程序设计框架。

4. 桌面与 Web 程序开发的共同平台

VB.NET 除了具有传统的桌面应用程序开发功能外,同时提供了现今流行的 Web 应用程序开发平台。既具有集表现层、应用层和数据层一体的可视化开发 Web 程序的功能,还具有 B/S 调试运行方式的便利。

5. 支持后台数据库访问

VB.NET 提供了多种访问后台数据库的技术手段。通过数据引擎,可以轻松访问诸如 SQL Server、Oracle、Access 等常见数据库。通过 XML 可以实现跨平台的数据交换和轻型数据交换。

6. 充分享用自带的帮助系统

通过安装 MSDN,VB.NET 就具有了本地帮助系统,随时可以获得上下文关联帮助。通过 Internet 还可以访问 MSDN 社区,获得大量最新、及时的 VB.NET 信息。帮助系统是学习 VB.NET 的良师益友。

1.2 熟悉程序设计的集成环境

Visual Studio.NET 集成环境(IDE)提供了.NET 框架下的 VB、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 集成环境是入门的必修课。

1.2.1 建立 VB.NET 解决方案

要创建一个 VB.NET 的 Windows 应用程序,先要创建一个 VB.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 VB.NET 项目

进入 Visual Studio.NET 后,有两种创建 VB.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”菜单命令。

2. 选择程序模板

进入如图 1.4 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”。

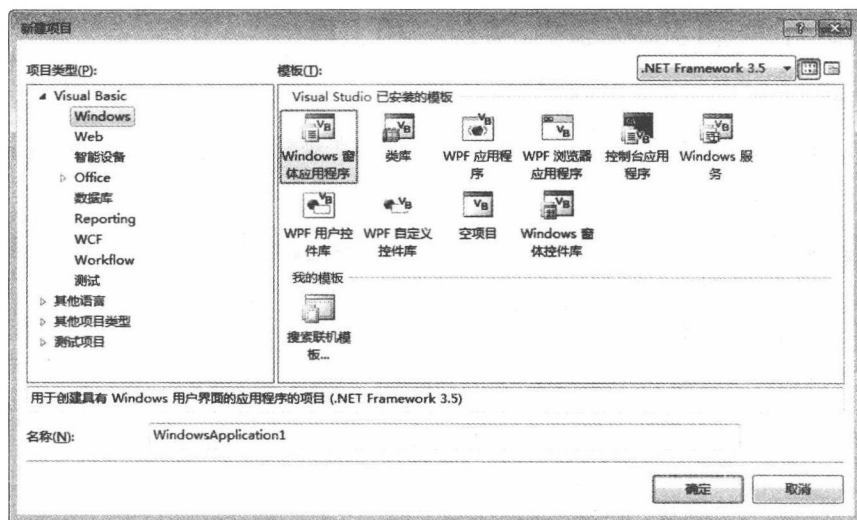


图 1.4 “新建项目”对话框

3. 拟定项目名称

默认项目名为“WindowsApplication1”。在“名称”处可以修改成你想命名的项目名(比如“例题 1.1 控制小球的运动”)。单击“确定”按钮,进入 VB.NET 集成环境的编程状态,类似图 1.2 的样子。

4. 建立并保存解决方案

尽管有了临时项目名,并且已经可以编程,但是该项目并没有被真正存储过。此时,执行“文件/全部保存”菜单命令,系统会弹出“保存项目”对话框,如图 1.5 所示。

项目“名称”显示新建项目时的临时名称(可以再次更改)。在“位置”处,直接输入存储路

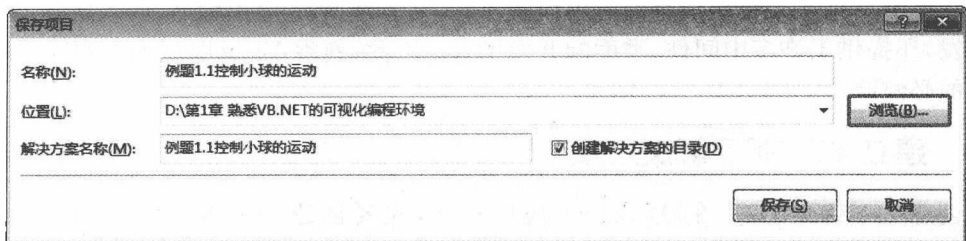



图 1.5 “保存项目”对话框

径,或单击“浏览”按钮,指定存储路径均可。此时,“解决方案名称”会同时更改成项目名称。由于简单的程序一个解决方案仅包含一个项目,因此,我们保持解决方案与项目同名。单击“保存”按钮,就在指定位置保存了“例题 1.1 控制小球的运动”的解决方案和项目。

 **注意:** 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。具体操作:进入 Visual Studio.NET 后,执行“工具/选项”菜单命令,选择“项目和解决方案/常规”,勾选“创建时保存新项目”选项。

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,如图 1.6 所示。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”。

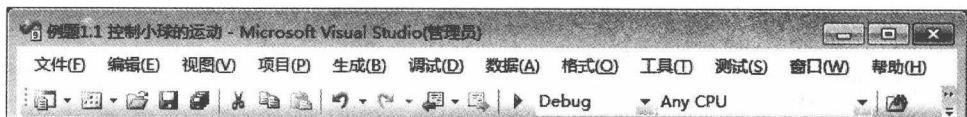


图 1.6 集成环境窗口的标题栏、菜单栏和工具栏

2. 菜单栏

与大多数 Windows 应用程序的标准菜单栏一样,VB.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”菜单的“工具栏”,可以个性化显示工具栏配置。大多数情况下,系统感知的工具栏已经想得很周到了。

1.2.3 对象的可视化布局窗口

1. 窗体设计器窗口

当您选择了“Form1[设计]”选项卡时,就进入了用户界面的可视化“窗体设计器”窗口,如

图 1.2 所示。借助于工具箱,窗体设计时就像在画图板上“贴”上控件,可视化移动控件和改变其大小;借助于属性窗,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里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,可以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”和“Visual Basic PowerPacks”等几种,安装 Visual Studio .NET 时,不同的选项,控件内容会稍有不同。单击“+”栏可展开具体控件,单击“-”栏可收缩控件。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 1.7 所示。

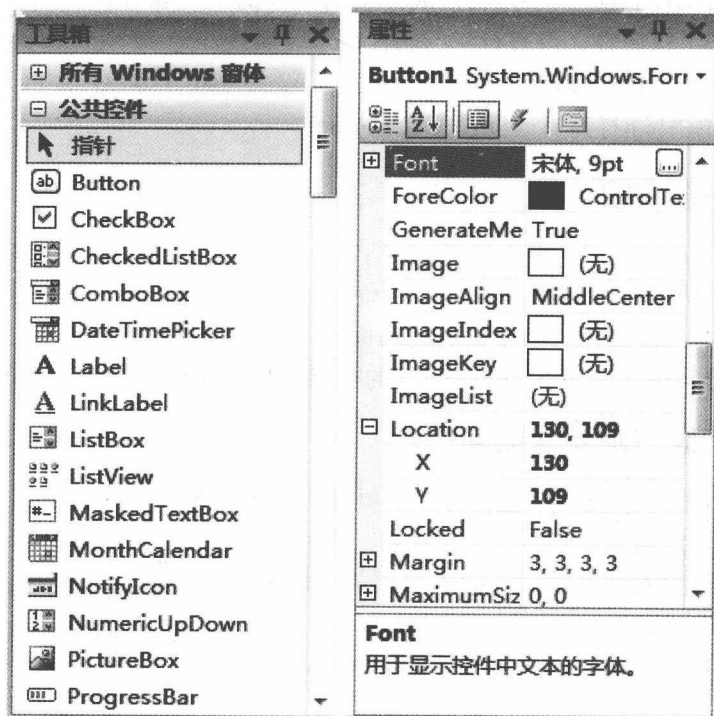


图 1.7 “工具箱”和“属性”窗口

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,如图 1.3 所示),腾出大片窗体空间供设计用。当鼠标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了集成环境窗体上了。



3. 属性窗口

虽然称“属性”窗口,其实也可显示“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 1.7 所示:

(1) “控件名/父类名”下拉框: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: 属性窗内显示内容的排列顺序和“属性/事件”控制。

-  按钮: 分别将属性内容“按分类顺序”和“字母顺序”排列。
-  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为可视化设计,则属性窗可称为数字化设计。前者直观简便,后者精细准确。单击属性名前“+”号,可以展开当前属性的下级属性;单击属性名前“-”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),也可下拉框选择(如 ForeColor),或者单击  打开对话框设置(如 Font)。

1.2.4 代码编程窗口

当您选择了“Form1.vb”选项卡或者双击了控件时,就进入了“代码编程”窗口,如图 1.3 所示。窗口上方左边下拉框可以选择控件对象,右边下拉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,您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。

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 1.8 所示的“解决方案资源管理器”中可以看到仅包含了 1 个项目。项目名为“例题 1.1 控制小球的运动”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。鼠标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 1.8 “解决方案资源管理器”窗口

1.2.6 其他窗口

除了上述几个在窗体设计、代码设计中的相关窗口外,还有“命令”窗口、“输出”窗口、“错误列表”窗口等。如果意外关闭了某些窗口,可以在集成环境菜单栏的“视图”菜单下,重新打开这些窗口。

程序运行发生错误或遇调试断点暂停程序时,系统还会跳出“监视”窗口(见图 1.9)和“调