

高等院校计算机教育系列教材

Visual Basic .NET 程序设计教程

魏 峥 主 编
王 军 王德亮 副主编



清华大学出版社

内 容 简 介

Visual Basic .NET 是一门面向对象的程序设计语言,是面向对象程序设计教学的主干语言之一。本书从教学实践出发,对 Visual Basic .NET 进行了全面阐述。全书共分 11 章,包括 Visual Basic .NET 概述、Visual Basic .NET 程序设计基础、流程控制、复合数据类型、过程、程序调试和异常处理、Windows 窗体和控件、界面设计、面向对象的程序设计、数据库及应用、Web 应用程序开发等内容。

本书内容详实、概念准确、编排合理。可作为高等院校程序设计课程的教材,也可作为广大希望掌握 Visual Basic .NET 编程的程序设计人员的参考书。

版权所有,翻印必究。举报电话:010-62782989 13501256678 13801310933

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

本书防伪标签采用特殊防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

图书在版编目(CIP)数据

Visual Basic .NET 程序设计教程/魏峥主编;王军,王德亮副主编. —北京:清华大学出版社,2005.10
(高等院校计算机教育系列教材)
ISBN 7-302-11898-1

I. V… II. ①魏… ②王… ③王… III. BASIC 语言—程序设计—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2005)第 109606 号

出 版 者:清华大学出版社 地 址:北京清华大学学研大厦
http://www.tup.com.cn 邮 编:100084
社 总 机:010-62770175 客 户 服 务:010-62776969

组稿编辑:黄 飞

文稿编辑:闫光龙

排版人员:房利萍

印 刷 者:北京国马印刷厂

装 订 者:北京市密云县京文制本装订厂

发 行 者:新华书店总店北京发行所

开 本:185×260 印张:23 字数:546 千字

版 次:2005 年 10 月第 1 版 2005 年 10 月第 1 次印刷

书 号:ISBN 7-302-11898-1/TP·7727

印 数:1~5000

定 价:30.00 元

前 言

Visual Basic .NET 是一门面向对象的程序设计语言，是面向对象程序设计教学的主干语言之一。本书从教学实践的角度对 Visual Basic .NET 进行了全面阐述。全书共分 11 章，每章均配有上机指导和习题，通过大量实例，深入浅出、系统全面地介绍了 Visual Basic .NET 程序设计。

第 1 章介绍了 .NET 基本概念、Visual Studio 集成开发环境；第 2 章至第 5 章介绍了 Visual Basic .NET 程序设计基础、顺序结构、选择结构、循环结构、复合数据类型、过程和函数等基础知识；第 6 章介绍了程序调试和异常处理；第 7、8 章介绍了 Windows 应用程序界面设计；第 9 章介绍了面向对象的程序设计方法；第 10 章介绍了数据库技术，第 11 章介绍了简单的 Web 应用程序开发。

本书避免了普通工具书中知识点与实例脱离的现象，将重要的知识点嵌入具体实例中。全书提供大量实例，使学生可以循序渐进、边学边用，轻松掌握 Visual Basic .NET 程序设计。本书可作为高等院校程序设计课程教材，也可作为想要掌握 Visual Basic .NET 程序设计人员的参考书。

本书第 1~6 章由王军编写，第 7~9 章由魏峥编写，第 10~11 章由王德亮编写。全书由魏峥统稿。由于时间仓促，水平所限，书中难免存在一些疏漏和不足，恳请读者和专家批评指正。同时欢迎读者与作者联系，共同探讨技术问题，不断提高 Visual Basic .NET 程序设计水平。

编 者

目 录

第1章 Visual Basic .NET 概述 1

1.1 .NET 基本概念..... 1	
1.1.1 .NET 开发平台..... 1	
1.1.2 .NET 框架..... 2	
1.1.3 .NET 框架目标..... 2	
1.2 面向对象程序设计的基本概念..... 3	
1.3 Visual Studio .NET 集成开发环境..... 6	
1.3.1 Visual Studio .NET 集成开发环境的启动..... 6	
1.3.2 Visual Basic .NET 集成开发环境..... 7	
1.3.3 创建简单的应用程序..... 10	
1.4 创建 Visual Basic .NET 控制台程序..... 12	
1.4.1 控制台输入输出方法..... 12	
1.4.2 创建控制台应用程序..... 13	
1.5 联机帮助系统..... 14	
1.5.1 动态帮助..... 14	
1.5.2 帮助菜单..... 15	
1.5.3 互联网资源..... 15	
1.6 上机指导..... 15	
1.6.1 Visual Basic .NET 的 启动和退出..... 15	
1.6.2 设置窗体属性..... 16	
1.6.3 设置窗体事件..... 16	
1.6.4 控制台应用程序的 建立和执行..... 16	
1.7 习题..... 17	

第2章 Visual Basic .NET 程序设计基础..... 19

2.1 Visual Basic .NET 程序的书写规则..... 19	
2.1.1 关键字与标识符..... 19	
2.1.2 程序语句..... 20	
2.1.3 程序的书写规则..... 20	

2.1.4 命令格式中的符号约定..... 21	
2.2 基本数据类型..... 21	
2.2.1 数值数据类型..... 21	
2.2.2 字符数据类型..... 22	
2.2.3 Boolean 数据类型..... 22	
2.2.4 Byte 数据类型..... 22	
2.2.5 Date 数据类型..... 23	
2.2.6 Decimal 数据..... 23	
2.2.7 Object 数据类型..... 23	
2.3 常量与变量..... 24	
2.3.1 常量..... 24	
2.3.2 变量..... 25	
2.3.3 Option Explicit 语句..... 26	
2.4 运算符和表达式..... 27	
2.4.1 算术运算符和算术表达式..... 27	
2.4.2 关系运算符..... 28	
2.4.3 逻辑运算符..... 29	
2.4.4 字符运算符..... 30	
2.5 常用系统函数..... 30	
2.5.1 数学函数..... 30	
2.5.2 字符处理函数..... 31	
2.5.3 类型转化函数..... 32	
2.5.4 随机数函数..... 32	
2.5.5 日期时间函数..... 33	
2.5.6 DateAdd()增减日期函数..... 33	
2.5.7 DateDiff()时间间隔数函数..... 34	
2.6 顺序结构的程序设计..... 35	
2.6.1 赋值语句..... 35	
2.6.2 使用 InputBox 函数..... 37	
2.6.3 使用 MsgBox 函数和 MsgBox 语句..... 38	
2.7 上机指导..... 40	
2.7.1 计算表达式(1)..... 40	
2.7.2 计算表达式(2)..... 41	
2.7.3 写出函数的返回值..... 42	

2.7.4 弧度转化角度.....	43	4.4.1 结构类型与结构变量 的定义.....	80
2.8 习题.....	44	4.4.2 结构变量的初始化 及其引用.....	82
第3章 流程控制	46	4.4.3 结构数组.....	86
3.1 算法概论.....	46	4.5 上机指导.....	88
3.1.1 算法的概念.....	46	4.5.1 数组元素的赋值和引用.....	88
3.1.2 算法的描述.....	47	4.5.2 二维数组元素的 赋值和引用.....	89
3.2 选择结构.....	47	4.5.3 使用结构数组, 改变数组大小.....	92
3.2.1 IF 条件语句.....	47	4.5.4 选择法排序.....	94
3.2.2 Select Case 语句.....	52	4.5.5 实现批量输入.....	95
3.3 循环结构.....	54	4.6 习题.....	96
3.3.1 For...Next 循环语句.....	54	第5章 过程	98
3.3.2 While 循环语句.....	56	5.1 Sub 过程.....	98
3.3.3 Do While...Loop 循环语句.....	58	5.1.1 定义 Sub 过程.....	98
3.3.4 Do...Loop While 循环语句.....	59	5.1.2 建立 Sub 过程的方法.....	99
3.3.5 Do Until...Loop 循环语句.....	60	5.1.3 Sub 子过程的调用.....	100
3.3.6 Do...Loop Until 循环语句.....	62	5.2 函数过程的定义与调用.....	101
3.3.7 多重循环.....	63	5.3 参数传递.....	102
3.4 上机指导.....	64	5.3.1 按值传递参数.....	102
3.4.1 计算分段函数.....	64	5.3.2 按址传递参数.....	103
3.4.2 判断成绩等级.....	66	5.3.3 参数数组.....	104
3.4.3 求一元二次方程的根.....	66	5.3.4 可选参数.....	105
3.4.4 递推法.....	68	5.3.5 数组参数.....	106
3.4.5 穷举法.....	70	5.4 变量作用域.....	110
3.5 习题.....	70	5.4.1 声明局部变量.....	110
第4章 复合数据类型	72	5.4.2 静态变量.....	111
4.1 数组.....	72	5.4.3 声明模块变量.....	112
4.1.1 一维数组的声明.....	72	5.4.4 声明命名空间变量.....	113
4.1.2 二维数组的声明.....	73	5.5 过程的嵌套调用与递归调用.....	113
4.2 数组的基本操作.....	73	5.5.1 过程的嵌套调用.....	113
4.2.1 数组的初始化值.....	73	5.5.2 过程的递归调用.....	114
4.2.2 数组元素的引用.....	74	5.6 上机指导.....	115
4.2.3 复制数组.....	75	5.6.1 编写了过程 Sub Min 和 函数 Function Min.....	115
4.2.4 改变数组大小和求数组某维 上界函数 UBound()函数.....	76	5.6.2 使用参数数组.....	117
4.2.5 For Each 语句.....	78		
4.3 创建枚举.....	79		
4.4 结构.....	80		

5.6.3 函数递归调用.....	118	7.3.5 复选列表框.....	153
5.6.4 顺序查找.....	119	7.3.6 组合框.....	154
5.7 习题.....	120	7.4 其他常用控件.....	159
第6章 程序调试和异常处理.....	121	7.4.1 图片框.....	160
6.1 错误的种类.....	121	7.4.2 进度条.....	160
6.1.1 语法错误.....	121	7.4.3 定时器.....	161
6.1.2 运行时错误.....	122	7.4.4 滚动条.....	161
6.1.3 逻辑错误.....	123	7.5 键盘和鼠标器.....	165
6.2 调试工具.....	124	7.5.1 键盘.....	165
6.2.1 Visual Basic .NET 的 工作模式.....	124	7.5.2 鼠标器.....	171
6.2.2 使用调试工具调试程序.....	125	7.5.3 拖放.....	174
6.2.3 调试窗口.....	126	7.6 上机指导.....	178
6.3 异常处理.....	128	7.6.1 制作一个由命令按钮 组成的简单计算器.....	178
6.3.1 结构化异常处理.....	128	7.6.2 对文本框内的文字 进行各种修饰.....	180
6.3.2 非结构化异常处理.....	130	7.6.3 设计奥运会运动员 报名输入表单设计.....	183
6.4 上机指导.....	132	7.6.4 壁纸欣赏.....	186
6.4.1 非结构化异常处理.....	132	7.7 习题.....	189
6.4.2 结构化异常处理.....	133	第8章 界面设计.....	191
6.5 习题.....	134	8.1 创建菜单.....	191
第7章 Window 窗体与控件.....	135	8.1.1 建立下拉式菜单.....	191
7.1 窗体.....	135	8.1.2 弹出式菜单.....	194
7.1.1 窗体的结构.....	135	8.2 状态栏.....	194
7.1.2 窗体的属性.....	136	8.2.1 创建状态栏.....	195
7.1.3 设置启动窗体.....	138	8.2.2 在运行时设置窗格内容.....	196
7.1.4 常用窗体事件.....	139	8.3 工具栏.....	197
7.1.5 创建应用程序的操作界面.....	140	8.3.1 工具栏的属性.....	197
7.1.6 控件的命名.....	141	8.3.2 创建工具条.....	198
7.2 文本类控件和命令控件.....	141	8.3.3 为工具栏按钮添加图标.....	198
7.2.1 标签.....	141	8.3.4 编写 ButtonClick 事件处理程序.....	200
7.2.2 文本框.....	142	8.4 多窗体程序设计.....	200
7.2.3 命令按钮.....	144	8.4.1 添加窗体.....	200
7.3 选择控件.....	148	8.4.2 与多重窗体程序 设计有关的方法.....	201
7.3.1 单选按钮.....	148	8.5 多文档界面(MDI).....	204
7.3.2 复选框.....	149		
7.3.3 框架.....	149		
7.3.4 列表框.....	152		

8.5.1 与 MDI 有关的属性、方法和事件	204	9.7.2 类的继承	264
8.5.2 MDI 应用程序中的菜单	208	9.8 习题	267
8.6 通用对话框	212	第 10 章 数据库及应用	269
8.6.1 打开文件对话框	212	10.1 数据库概述	269
8.6.2 保存文件对话框	214	10.1.1 创建数据库及事务日志	269
8.6.3 颜色对话框	215	10.1.2 创建表	271
8.6.4 字体对话框	215	10.1.3 添加记录	273
8.7 RichTextBox 控件	216	10.1.4 改变表结构	274
8.7.1 常用格式化属性	216	10.1.5 创建备份设备	274
8.7.2 文件操作方法	217	10.1.6 数据库备份	274
8.8 上机指导	220	10.1.7 数据库恢复	275
8.8.1 计算学习成绩的应用程序	220	10.2 SQL 语言	278
8.8.2 多文档窗体	225	10.2.1 数据查询	278
8.9 习题	234	10.2.2 INSERT 新建数据指令	282
第 9 章 面向对象的程序设计	235	10.2.3 UPDATE 修改数据指令	282
9.1 类和对象的创建	235	10.2.4 DELETE 删除数据指令	283
9.1.1 类的创建	235	10.3 ADO.NET	283
9.1.2 添加实例数据成员	236	10.3.1 ADO.NET 的定义	283
9.1.3 方法	237	10.3.2 ADO.NET 的对象模型	283
9.1.4 属性	237	10.3.3 使用 Connection 对象来连接数据源	284
9.1.5 对象的创建和使用	239	10.3.4 ADO.NET 联机模式的数据存取	288
9.1.6 事件的声明及其激发	241	10.3.5 ADO.NET 脱机模式的数据存取	295
9.2 构造函数	244	10.3.6 利用 ADO.NET 控件编写数据库应用程序	308
9.3 面向对象的高级概念	246	10.4 上机指导	314
9.3.1 方法重载	246	10.4.1 需求分析	314
9.3.2 共享成员	248	10.4.2 概念结构设计	316
9.4 类的继承	250	10.4.3 数据库设计	317
9.4.1 创建基类	251	10.4.4 应用程序设计	318
9.4.2 创建子类	251	10.4.5 运行结果	335
9.4.3 Overrideable 方法与重写方法	254	10.5 习题	335
9.5 类的多态	258	第 11 章 Web 应用程序开发	338
9.5.1 基于继承的多态	258	11.1 Web 应用程序概述	338
9.5.2 基于接口的多态	258	11.1.1 Web 的概念	338
9.6 类将类组织到命名空间	259		
9.7 上机指导	260		
9.7.1 类和对象的创建基础	260		

11.1.2 什么是 ASP.NET	339	11.3.3 创建一个 Web Service	344
11.1.3 Web 窗体的概念	340	11.4 上机指导	346
11.2 Web 窗体设计	340	11.4.1 基于 C/S 程序设计基础	346
11.2.1 Web 窗体设计的 一般步骤	340	11.4.2 基于 B/S 程序设计基础	348
11.2.2 Web 窗体设计实例	341	11.5 习题	350
11.3 Web Service 的创建和使用	342	附录 《Visual Basic .NET 程序设计》 课程教学大纲	351
11.3.1 Web Service 概念	342		
11.3.2 Web Service 的技术支持	343		

第1章 Visual Basic .NET 概述

【教学内容】

- .NET 基本概念
- Visual Studio .NET 集成开发环境
- 联机帮助系统

【教学要求】

- 了解: .NET 的定义、.NET 的开发平台和 .NET 的框架。
- 理解: Visual Studio .NET 集成开发环境; 应用程序结构、类、对象、属性、方法和事件等概念。
- 掌握: Visual Studio .NET 的启动和关闭方法; 联机帮助系统。

1.1 .NET 基本概念

Visual Basic .NET 是 Microsoft 公司最新推出的 Visual Studio .NET 可视化应用程序开发工具组中的一个重要成员。Visual Studio .NET 是 Microsoft 公司推出的第一个基于 .NET 框架的可视化应用程序开发工具, 其中包括 Visual Basic .NET、Visual C++ .NET(简称 VC++ .NET)和 Visual C# .NET(简称 VC# .NET)等开发工具。Visual Studio .NET 通过公共语言运行环境, 将 Visual Basic .NET、Visual C++ .NET 和 Visual C# .NET 等应用程序开发工具紧密地集成在一起, 使它们共同使用同一个集成开发环境(Integrated Development Environment, IDE), 并使用同一个基础类库, 这样大大简化了应用程序的开发过程, 为开发人员快速地创建分布式应用程序提供了强有力的支持。

1.1.1 .NET 开发平台

.NET 平台是下一代软件的基础, 这种软件以一种更为统一且个性化的方式来连接完全不同的系统、信息、设备和用户。.NET 平台通过 XML Web 服务使各种不同的技术相互协作。软件开发人员借助 .NET 平台下的工具和技术可以快速有效地构建商业解决方案。这些解决方案可以横跨不同组织间的多个应用程序和客户端设备, 并且使用户能够控制信息的传送方式、传送时间以及传送内容。

.NET 开发平台包括 .NET 框架和 .NET 开发工具等组成部分。.NET 框架(Framework)是整个开发平台的基础, 包括公共语言运行库(Common Language Run, CLR)和框架类库, .NET 开发工具包括 Visual Studio .NET 集成开发环境和 .NET 编程语言。其中, Visual Studio .NET 集成开发环境用来开发和测试应用程序。.NET 编程语言包括 Visual Basic .NET、Visual C++ .NET 和 Visual C# .NET 等用来创建运行在公共语言运行库上的应用程序。.NET 开发平台如图 1.1 所示。

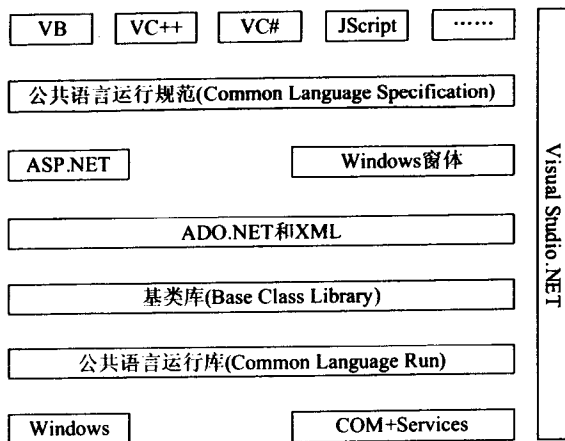


图 1.1 .NET 开发平台

1.1.2 .NET 框架

.NET 框架包括公共语言运行库和 .NET 类库。公共语言运行库是 .NET 的基础，用户可以将公共语言运行库看作一个在执行时管理代码的代理，它提供核心服务(如内存管理、线程管理和远程处理)，而且还强制实施严格的类型安全检查，以确保代码运行的安全性和可靠性。事实上，代码管理的概念是运行库的基本原则。以运行库为目标的代码称为托管代码，不以运行库为目标的代码称为非托管代码。.NET Framework 的另一个主要部件是类库，它是一个综合性的面向对象的可重用类型集合，用户可以使用它开发包含从传统的命令行或图形用户界面(GUI)应用程序到基于 ASP.NET 所提供的创新的应用程序(如 Web 窗体和 XML Web 服务)在内的应用程序。.NET 框架的组成如图 1.2 所示。

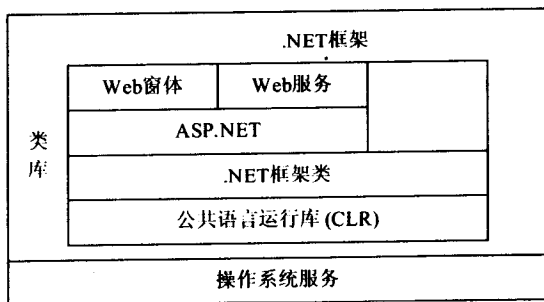


图 1.2 .NET 框架

1.1.3 .NET 框架目标

.NET 框架旨在实现下列目标：

- 提供一个一致的面向对象的编程环境，无论对象代码是在本地存储和执行，还是在本地执行但在 Internet 上分布，或者是在远程执行。
- 提供一个将软件部署和版本控制冲突最小化的代码执行环境。
- 提供一个保证代码(包括由未知的或不完全受信任的第三方创建的代码)安全执行

的代码执行环境。

- 提供一个可消除脚本环境或解释环境的性能问题的代码执行环境。
- 使开发人员的经验在面对类型大不相同的应用程序(如基于 Windows 的应用程序和基于 Web 的应用程序)时保持一致。
- 按照工业标准生成所有通信,以确保基于 .NET 框架的代码可与任何其他代码集成。

1.2 面向对象程序设计的基本概念

面向对象在软件开发领域并不是一个新的概念。从 20 世纪 60 年代出现于挪威的 K.Nygaard 等人开发的程序设计语言 Simula67 开始,对象的概念在越来越多的领域得到了应用和发展。Visual Basic 从 4.0 版本以来,已具备了强大的面向对象的功能,Visual Basic .NET 更是对以前版本的拓展和补充。面向对象技术有如此威力是得益于它的几个特性:属性、方法、继承和多态。所谓对象是包含数据和对数据操作的代码实体,或者说是在传统的数据结构中加入一些被称为成员函数的过程,因而赋予对象以动作。如果把一个人作为对象,那么名字、身高是他的属性,技能就是方法,这种技能可能是他从父辈那里继承来的。有了对象,程序就可以比较方便地反映客观世界。外部程序通过调用对象的方法来使对象工作。

面向对象程序设计从所处理的数据入手,以数据为中心而不是以服务(功能)为中心来描述系统。它把编程问题视为一个数据集合,数据相对于功能而言,具有更强的稳定性。

面向对象程序设计同结构化程序设计相比最大的区别在于:前者首先关心的是所要处理的数据,而后者首先关心的是功能。

面向对象程序设计是一种围绕真实世界的概念来组织模型的程序设计方法,它采用对象来描述问题空间的实体。关于对象这一概念,目前还没有统一的定义。一般的认为,对象是包含现实世界物体特征的抽象实体,它反映了系统为之保存信息和(或)与之交互的能力。它是一些属性及服务的一个封装体,在程序设计领域,可以用“对象=数据+作用于这些数据上的操作”这一公式来表达。

面向对象技术给软件发展带来如下益处:

- 可重用性:从一开始对象的产生就是为了重复利用,完成的对象将在今后的程序开发中被部分或全部地重复利用。
- 可靠性:由于面向对象的应用程序包含了通过测试的标准部分,因此更加可靠。由于大量代码来源于成熟可靠的类库,因而新开发程序的新增代码明显减少,这是程序可靠性提高的一个重要原因。
- 连续性:具有面向对象特点的语言有很大的兼容性。

另外,面向对象语言具有如下基本特征:

- 封装性:对象必须能够对其内部的某些元素进行保护,使它们只能被内部使用,而不受外部干扰。反过来,对象又必须同其他外部元素进行联系,以便对对象进行操作:在 Visual Basic .NET,类有私有的(private)、保护的(protected)和公有的(public)等访问机制。

- 继承性：通过对已有对象进行增加或部分修改的方法建立新的对象，对已有对象可以增加数据和过程，也可以对其中某些过程进行重新定义。最初的类被称为基类，从基类扩展出来的类称为派生类。从已有类派生出新类是为了获得更强的针对性。
- 多态性：多态性在功能上所反映的是，可以编写一个能操作多个类中对象的例程，该例程能以完全一样的方式处理不同类中不同对象。

1. 类和对象

在面向对象编程技术中类是主要的焦点，简单的说，类是一种提供功能的数据类型。对象是代码和数据的组合，可以作为一个单位来处理。对象可以是应用程序的一部分，比如可以是控件或窗体。整个应用程序也是一个对象。Visual Basic .NET 中的每个对象都是用类定义的。用饼干模子和饼干之间的关系作比，就会明白对象和它的类之间的关系。饼干模子是类。它确定了每块饼干的特征，比如大小和形状。用类创建对象，对象就是饼干。类是面向对象程序设计的核心技术，可以理解成一种定义了对象行为和外观的模板；把对象看作是类的原原本本的复制品。

面向对象的程序设计主要是建立在类和对象的基础上。通常的面向对象程序设计中的类是由程序员自己设计的。而在 Visual Basic .NET 中，类可由系统设计好，也可有程序员自己设计(本部分内容在第 9 章介绍)。

在 Visual Basic .NET 中，【工具箱】对话框上的可视类图标是 Visual Basic .NET 系统设计好的标准控件类。通过将类实例化，可以得到真正的控件对象。也就是当在窗体上画一个控件时，就将类转换为对象，即创建了一个控件对象，简称为控件，如图 1.3 所示。

【工具箱】对话框上的 Button 控件是类，它确定了 Button 的属性、方法和事件。窗体上显示的是 Button 对象，是类的实例化，它继承了 Button 类的特征，也可以根据需要修改各自的属性，例如按钮的大小、标签等；具有为控件设置输入焦点的方法和单击控件时发生的事件。

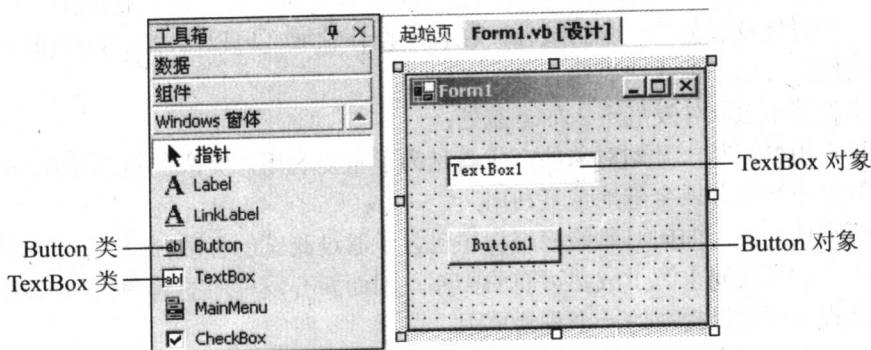


图 1.3 类和对象

2. 属性

属性是对对象特性的描述，Visual Basic .NET 为每一类对象都规定了若干属性，设计中可以改变具体对象的属性值。比如窗体的背景颜色、高度与宽度。

3. 方法

方法指的是控制对象动作行为的方式。它是对象本身内含的函数或过程，它也是一个动作，是一个简单的不必知道细节的无法改变的事件，但不称作事件；同样，方法也不是随意的，一些对象有一些特定的方法。

4. 事件

事件是发生在对象上的动作。事件的发生不是随意的，某些事件仅发生在某些对象上。

5. 属性、方法和事件之间的关系

Visual Basic .NET 对象具有属性、方法和事件。属性是描述对象的数据；方法告诉对象应做的事情；事件是对象所产生的事情，事件发生时可以编写代码进行处理。

Visual Basic .NET 的窗体和控件是具有自己的属性、方法和事件的对象。可以把属性看作一个对象的性质，把方法看作对象的动作，把事件看作对象的响应。

日常生活中的对象，如小孩玩的气球同样具有属性、方法和事件。气球的属性包括可以看到的一些性质，如它的直径和颜色。其他一些属性描述气球的状态(充气的或未充气的)或不可见的性质，如它的寿命。通过定义，所有气球都具有这些属性；这些属性也会因气球的不同而不同。

气球还具有本身所固有的方法和动作。如：充气方法(用氦气充满气球的动作)，放气方法(排出气球中的气体)和上升方法(放手让气球飞走)。所有的气球都具备这些能力。

气球还有预定义的对某些外部事件的响应。例如，气球对刺破它的事件响应是放气，对放手事件的响应是升空。

6. 事件驱动模型

在传统的或“过程化”的应用程序中，应用程序自身控制了执行哪一部分代码和按何种顺序执行代码。从第一行代码执行程序并按应用程序中预定的路径执行，必要时调用过程。

在事件驱动的应用程序中，代码不是按照预定的路径执行，而是在响应不同的事件时执行不同的代码片段。事件可以由用户操作触发、也可以由来自操作系统或其他应用程序的消息触发、甚至由应用程序本身的消息触发。这些事件的顺序决定了代码执行的顺序，因此应用程序每次运行时所经过的代码的路径都是不同的。

因为事件的顺序是无法预测的，所以在代码中必须对执行时的“各种状态”作一定的假设。当作出某些假设时(例如，假设在运行来处理某一输入字段的过程之前，该输入字段必须包含确定的值)，应该组织好应用程序的结构，以确保该假设始终有效(例如，在输入字段中有值之前禁止使用启动该处理过程的命令按钮)。

在执行中代码也可以触发事件。例如，在程序中改变文本框中的文本将引发文本框的 Change 事件。如果 Change 事件中包含有代码，则将导致该代码的执行。如果原来假设该事件仅能由用户的交互操作所触发，则可能会产生意料之外的结果。正因为这一原因，所以在设计应用程序时理解事件驱动模型并牢记在心是非常重要的。

1.3 Visual Studio .NET 集成开发环境

.NET 中所有的开发语言都采用了统一的集成开发环境(Integrated Development Environment, IDE), 使用同一个 IDE 为开发者提供了极大的方便。本节将简单介绍 Visual Studio .NET 集成开发环境的启动和组成。

1.3.1 Visual Studio .NET 集成开发环境的启动

Visual Studio .NET 集成开发环境的启动与一般 Windows 软件一样, 有多种方式, 这里介绍最常用的一种, 其启动方式为:

(1) 选择【开始】|【程序】| Microsoft Visual Studio .NET 2003 | Microsoft Visual Studio .NET 2003 命令, 进入如图 1.4 所示的【起始页】, 它是集成开发环境中默认的 Web 浏览器主页。

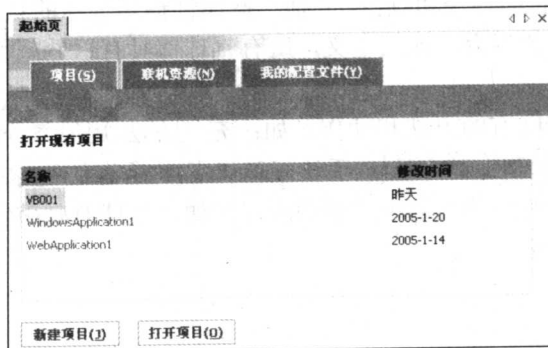


图 1.4 Visual Studio 的【起始页】

(2) 在 Visual Studio .NET 集成开发环境中, 选择【文件】|【新建】|【项目】命令, 弹出【新建项目】对话框, 在【项目类型】中选中【Visual Basic 项目】, 在【模板】中选中【Windows 应用程序】。在【位置】文本框中输入项目保存的位置, 在【名称】文本框中输入项目的名称, 如图 1.5 所示。

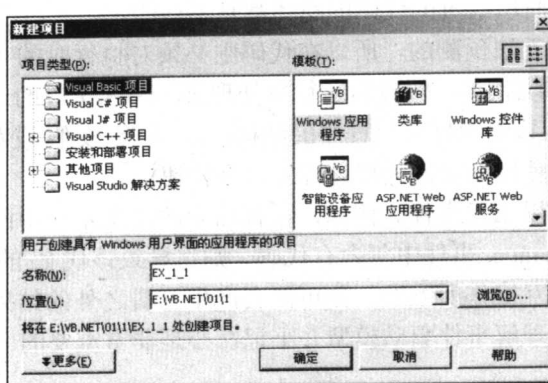


图 1.5 【新建项目】对话框

(3) 单击【确定】按钮，进入 Visual Basic .NET 集成开发环境，如图 1.6 所示。

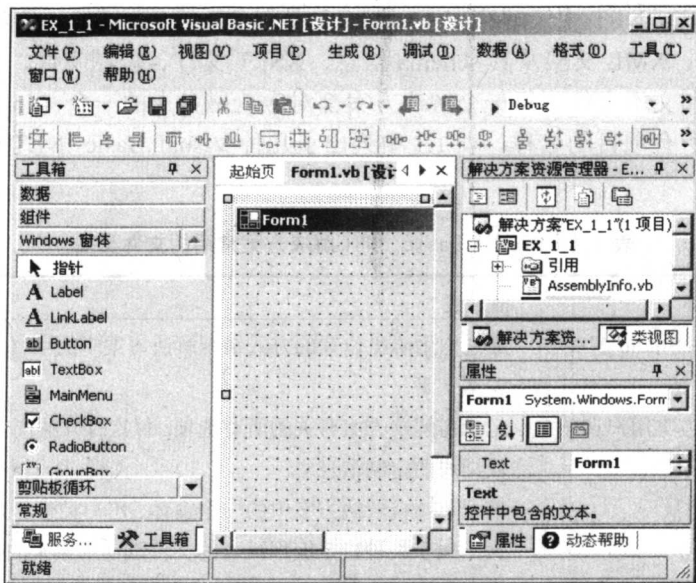


图 1.6 Visual Basic .NET 集成开发环境

1.3.2 Visual Basic .NET 集成开发环境

Visual Basic .NET 集成开发环境除了拥有 Microsoft 应用软件常见的标题栏、菜单栏、工具栏外，还包括 Visual Basic .NET 的几个独立窗格：【解决方案资源管理器】窗格，【属性】窗格，【工具箱】窗格，作为类的【窗体】等，如图 1.6 所示。

 **提示：** 选择【工具】|【选项】命令，出现【选项】对话框，单击【重置窗口布局】按钮，可以恢复默认窗口布局。

1. 设计器

Visual Studio .NET 提供了很多设计器，包括 Window 窗体设计器、Web 窗体设计器、组件设计器、XML 设计器、控件设计器等，Window 窗体设计器只是其中一种，如图 1.7 所示。通常把设计器所在的窗体称为主窗体。Window 窗体设计器窗口简称窗体(Form)，是应用程序最终面向用户的窗口，它对应于应用程序的运行结果。

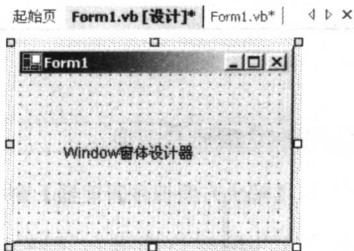


图 1.7 Window 窗体设计器

2. 解决方案资源管理器

在 Visual Studio .NET 中, 解决方案(solution)是创建一个应用程序所需要的一组项目, 包括 HTML 页面、XML 文档及其 schema 信息、XSLT 文件、位图文件、光标文件、图标文件、资源文件和文本文件。每一个解决方案都对应着磁盘上的一个 .sln 文件, 该文件包含了解决方案中所有项目的名称。表 1.1 中列出了几种 Visual Basic .NET 解决方案常用的文件类型。

表 1.1 Visual Basic .NET 解决方案常用的文件类型

扩展名	名称	描述
.sln	Visual Studio 解决方案	通过引用磁盘位置的方式组织解决方案中的项目、项目元素、解决方案元素
.suo	解决方案用户选项	记录与解决方案有关的所有选项。每次打开解决方案时, 它就能读取用户的定制信息。
.vb	VB 项目	表示单项目解决方案中的一个窗体、用户控件、类和模块文件。任何不属于某种特定编程语言的文件都使用原来的扩展名
.vbproj	VB 项目	表示多个项目解决方案中的一个窗体、用户控件、类和模块文件。该扩展名可以用来区分 Visual Basic .NET 文件和其他 .NET 兼容语言的文件(Visual C#使用.csproj 扩展名)
.aspx	Web 项目元素	Web 项目元素包含了和 Web 相关的文件, 例如, .aspx 用来表示 Web 窗体, .asmx 用来表示 XML Web service, .asax 用来表示全局应用程序类, Web 项目仍然使用.vb 作为类和模块文件的扩展名
.asmx		
.asax		

如果集成环境中没有出现【解决方案资源管理器】窗格, 可通过选择【视图】|【解决方案资源管理器】命令来显示该窗格。双击窗格中的某个项目, 可以对其进行编辑, 如图 1.8 所示。

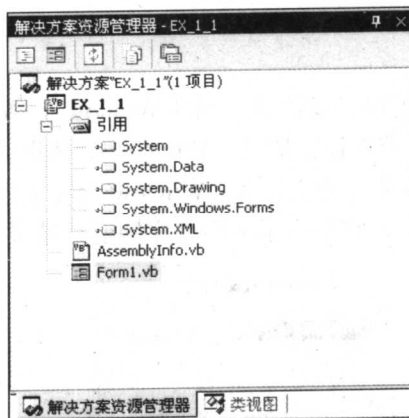


图 1.8 【解决方案资源管理器】窗格

在窗口顶部, 有一系列按钮。这些按钮将根据在【解决方案资源管理器】窗格中所选项目而变化, 其中:

- **【查看代码】** : 显示解决方案资源管理器中具有焦点的文件的代码。
- **【视图设计器】** : 显示解决方案资源管理器中具有焦点的特定文件的设计器。
- **【刷新】** : 刷新解决方案资源管理器。
- **【显示所有文件】** : 显示所有文件, 包括格式文件的任何代码。
- **【属性】** : 显示所选文件的属性。

3. 属性窗口

【属性】窗格如图 1.9 所示。如果集成环境中没有出现该窗格, 可通过选择菜单**【视图】|【属性】**命令来显示该窗格。



图 1.9 **【属性】**窗格

【属性】窗格用于设置窗体或者控件的属性。属性定义了控件的信息, 诸如大小、颜色和位置等。每个控件都有自己的一组属性。

4. 工具箱

工具箱如图 1.10 所示。如果集成环境中没有出现该窗格, 可通过选择菜单**【视图】|【工具箱】**命令来显示。

【工具箱】中包含了可重用的控件(或称组件), 用于自定义应用程序。使用可视化的方法编程时, 程序员可在窗体中拖放控件, 绘制出应用程序界面, 而不用自己去写代码。

控件在**【工具箱】**中是以组的形式出现的。通过使用**【工具箱】**右下部的黑色滚动箭头 , 能浏览所有的控件。组的第一项不是控件, 是鼠标指针, 单击它可以取消控件的选择, 以便重新选择其他控件。

5. 代码窗口

在主设计界面上, 双击 Form1 窗体, 或者在**【解决方案资源管理器】**中选择 Form1, 右击, 从快捷菜单中选择**【查看代码】**命令, 出现如图 1.11 所示内容。

 **说明:** 第一行声明了叫作 Form1 的新类, 在 Visual Basic .NET 中, 可以在任何代码文件中声明类, 其数量可以是任意的。第二行声明了 Form1 是从 System.Windows.Forms.Form 派生而来的类。从这里可以看出, 窗体是可以继承的类,