

高职高专21世纪计算机规划教材



Visual Basic.NET 程序设计教程

肖金秀 韩贵来 编著

冶金工业出版社

高职高专 21 世纪计算机规划教材

Visual Basic.NET 程序设计教程

肖金秀 韩贵来 编著

北 京

冶金工业出版社

内 容 简 介

本书详细介绍了 Visual Basic.NET 的安装方法、编程环境、Visual Basic.NET 的程序设计基础、面向对象编程、常用 Windows 窗体控件和高级 Windows 窗体控件、关系数据库基础及 ADO.NET 的使用方法、使用 ASP.NET 开发 Web 窗体、Visual Basic 中的多线程和异常处理等。

本书语言简洁、思路清晰、概念准确、实用性强，既可作为高职高专等学校及社会培训班的教材，也可供学习 Visual Basic.NET 的读者作自学使用。

图书在版编目 (CIP) 数据

Visual Basic.NET 程序设计教程 / 肖金秀, 韩贵来编著.

北京: 冶金工业出版社, 2005.7

ISBN 7-5024-3783-5

I. V... II. ①肖...②韩... 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 065351 号

出版人 曹胜利 (北京沙滩嵩祝院北巷 39 号, 邮编 100009)

责任编辑 程志宏

佛山市新粤中印刷有限公司印刷; 冶金工业出版社发行; 各地新华书店经销

2005 年 7 月第 1 版第 1 次印刷

787mm × 1092mm 1/16; 17 印张; 388 千字; 262 页

25.00 元

冶金工业出版社发行部 电话: (010) 64044283 传真: (010) 64027893

冶金书店 地址: 北京东四西大街 46 号 (100711) 电话: (010) 65289081

(本社图书如有印装质量问题, 本社发行部负责退换)

前 言

一、关于 Visual Basic.NET

Visual Basic.NET 是目前世界上最为流行的编程工具之一，一方面它摆脱了 Windows 复杂的编程模式，使开发人员不必了解 Windows 的内部机制也能设计出与其相媲美的应用程序；另一方面它又是集最新软件应用之大成，如面向对象编程、多媒体控制、数据库操作、OLE 网络通讯等等，无一不被它所包容。与 Visual Basic 语言以前进行的改进不同，微软将新版 Visual Basic 语言命名为 Visual Basic.NET，用意显然是将其置于 .NET 的战略计划之中。

众所周知，Internet 是当今计算机技术研究最为活跃的领域，客户/服务器两层模式仅仅存在了一段时间，就迅速地被浏览器/Web 服务器/数据库模式所代替，Visual Basic.NET 不仅可以满足一般编程和客户/服务器应用开发的需要，更为精彩的是，它还可以像创建普通表单一样创建用户自己的 Web 动态页面。

与以往的 Visual Basic 版本相比，Visual Basic.NET 具有很多优点，比如统一的程序设计模式，在以前的 Visual Basic 语言环境中，如果要求程序完成任务的性质不同，则需要采用不同的设计模式，这样处理的优点是清晰而方便，但却加大了程序设计人员的学习量和学习难度。

在 Visual Basic 6.0 中，面向对象的能力还不是很强，但随着 Visual Basic.NET 的出现，其面向对象的能力大大增强了。Visual Basic.NET 不仅提供了新的面向对象的特性，而且它也改变了在 Visual Basic 6.0 中实现一些特性时所用的方法。

二、本书结构

本书共分为 11 章，由浅入深地介绍了 Visual Basic.NET 的强大功能和使用方法。具体结构安排如下：

第 1 章：Visual Basic.NET 入门。主要介绍了 .NET 开发平台、Visual Basic.NET 的特点和 Visual Basic.NET 中的语言变化、Visual Basic.NET 安装。

第 2 章：Visual Basic.NET 开发环境。主要介绍了 Visual Basic.NET 的起始页、集成开发环境 (IDE) 的各种窗口、Visual Basic.NET 的各种菜单、创建一个简单的 Visual Basic.NET 应用程序。

第 3 章：Visual Basic.NET 程序设计基础。主要介绍了常量、变量、数据类型、数组、运算符、控制结构语句、函数以及 Visual Basic.NET 语法上的新特点。

第 4 章：面向对象编程。主要介绍了面向对象的基本特性、面向对象的基本概念、类与标准模块、面向对象的实现、继承、接口以及对象的生命周期。

第 5 章：常用 Windows 窗体对象。主要介绍了 Windows 窗体、控件、文本控件、按钮控件、对话框控件、时间控件和图片框控件。

第 6 章：高级 Windows 窗体设计。主要介绍了 Windows 窗体设计的界面样式、菜单

控件、多文档界面程序（MDI）以及其他高级窗体控件。

第 7 章：关系数据库基础。主要介绍了关系数据库基础知识、SQL 语言基础、SQL 常用函数、SQL 中的运算符、建立 SQL 查询以及在 Visual Basic.NET 中建立查询。

第 8 章：ADO.NET 数据访问基础。主要介绍 ADO.NET 对象、数据存储基础、使用表单数据绑定、数据连接与访问、XML 扩展支持、使用 Access 和 SQL Server 建立数据库。

第 9 章：使用 ASP.NET 开发 Web 窗体。主要介绍了 Web 应用程序基础知识、服务器控件、Web 服务。

第 10 章：Visual Basic.NET 中的多线程。主要介绍了进程和线程、多线程的使用、通过多线程传送数据、线程的同步、等待句柄、线程计时器、取消事务、线程的优先级、线程池以及死锁。

第 11 章：Visual Basic.NET 中的异常处理。主要介绍了错误类型、Visual Basic.NET 中程序的编译与调试、Visual Basic.NET 中的错误处理语句结构、错误处理过程以及 Visual Basic 6.0 和 Visual Basic.NET 中的错误处理机制比较。

三、本书特点

本书语言简洁、思路清晰、概念准确，可以让读者在最短时间内掌握 Visual Basic.NET 的使用技巧，并成为 Visual Basic.NET 高手。本书不仅介绍了 Visual Basic.NET 的基础知识，还汇集了 Visual Basic.NET 中一些鲜为人知的方法和技巧，同时包括很多新思想、新知识。

四、本书适用对象

本书是学习 Visual Basic.NET 编程知识的初、中级读者良好的进阶教程，同时也可以作为高职高专学校和培训班的计算机专业教材。

由于作者的水平有限，加上时间仓促，书中错误和不足之处在所难免，希望读者和同行给予批评和指正。

虽然经过严格的审核、精细的编辑，本书在质量上有了一定的保障，但我们的目标是力求尽善尽美，欢迎广大读者和专家对我们的工作提出宝贵建议，联系方法如下：

电子邮件：service@cnbook.net

网址：www.cnbook.net

本书的源代码、电子教案和习题的参考答案可在该网站免费下载。此外，该网站还有一些其他相关书籍的介绍，可以方便读者选购参考。

编 者
2005 年 5 月

目 录

第 1 章 Visual Basic.NET 入门	1	2.2.10 命令窗口	25
1.1 .NET 开发平台	1	2.3 Visual Basic.NET 的各种菜单	25
1.1.1 .NET 的概念	1	2.3.1 文件菜单	25
1.1.2 .NET 框架简介	2	2.3.2 编辑菜单	26
1.1.3 “软件作为服务”的思想	4	2.3.3 视图菜单	27
1.2 Visual Basic.NET 简介	4	2.3.4 项目菜单	28
1.2.1 Visual Basic.NET 的特点	4	2.3.5 生成菜单	28
1.2.2 Visual Basic.NET 中的语言变化	5	2.3.6 调试菜单	29
1.3 Visual Basic.NET 的安装	6	2.3.7 工具菜单	29
1.3.1 安装环境要求	7	2.3.8 窗口菜单	30
1.3.2 安装步骤	7	2.3.9 帮助菜单	31
小结	12	2.4 创建一个简单的 Visual Basic.NET	
综合练习一	12	应用程序	31
一、选择题	12	2.4.1 创建项目	31
二、填空题	13	2.4.2 创建用户界面	32
三、思考题	13	2.4.3 设置窗体和控件属性	32
四、上机操作	13	2.4.4 添加代码	33
第 2 章 Visual Basic.NET 开发环境	14	2.4.5 编译	34
2.1 Visual Basic.NET 的起始页	14	2.4.6 调试和运行	34
2.1.1 项目 (S)	14	小结	35
2.1.2 联机资源 (N)	15	综合练习二	35
2.1.3 我的配置文件 (Y)	16	一、选择题	35
2.2 集成开发环境 (IDE) 的各种窗口	16	二、填空题	36
2.2.1 服务器资源管理器	17	三、思考题	36
2.2.2 工具箱	18	四、上机操作	36
2.2.3 解决方案资源管理器	20	第 3 章 Visual Basic.NET 程序设计基础	37
2.2.4 类视图	21	3.1 常量	37
2.2.5 属性窗口	21	3.2 变量	37
2.2.6 编辑器窗口	21	3.2.1 变量的声明	37
2.2.7 任务列表	22	3.2.2 变量的命名规则	38
2.2.8 对象浏览器	23	3.2.3 匈牙利命名法	38
2.2.9 宏资源管理器	24	3.2.4 变量的作用域	39

3.2.5 静态变量	40
3.3 数据类型	41
3.3.1 基本数据类型	41
3.3.2 枚举数据类型	42
3.3.3 自定义数据类型	43
3.3.4 类型转换	43
3.4 数组	44
3.4.1 数组声明	45
3.4.2 数组赋值	45
3.4.3 二维数组和多维数组	45
3.4.4 数组的重新声明	46
3.5 运算符	46
3.5.1 算术运算符	47
3.5.2 连接运算符	47
3.5.3 赋值运算符	47
3.5.4 比较运算符	47
3.5.5 逻辑运算符	48
3.5.6 运算符的优先级	49
3.6 控制结构语句	50
3.6.1 If 语句	50
3.6.2 Select...Case 语句	52
3.6.3 For...Next 语句	53
3.6.4 While...End While 循环	54
3.6.5 Do...Loop 语句	54
3.6.6 跳离语句 Exit	56
3.7 函数	56
3.7.1 无返回值的 Sub 函数	57
3.7.2 有返回值的 Function 函数	57
3.7.3 参数传递	58
3.7.4 常用内部函数	60
3.8 Visual Basic.NET 语法上的新特点	61
3.8.1 变量的声明与设定的改进	62
3.8.2 新增的运算式	63
3.8.3 数组与字符串的改进	64
3.8.4 函数声明及调用的改进	64
3.8.5 关键字的改变	65
小结	66

综合练习三	67
一、选择题	67
二、填空题	67
三、思考题	67
四、上机操作	67

第 4 章 面向对象编程..... 68

4.1 面向对象的基本特性	68
4.1.1 抽象化	68
4.1.2 封装	69
4.1.3 继承	69
4.1.4 多态性	70
4.2 面向对象的基本概念	71
4.2.1 对象	71
4.2.2 域	72
4.2.3 属性	73
4.2.4 方法	74
4.2.5 事件	75
4.3 类与标准模块	78
4.3.1 类	78
4.3.2 模块	79
4.4 面向对象的实现	79
4.4.1 类与对象的关系	79
4.4.2 创建类	80
4.4.3 类关键字	81
4.4.4 创建命名空间	81
4.4.5 创建类库	82
4.4.6 创建方法	83
4.4.7 创建属性	84
4.4.8 重载方法	85
4.5 继承	86
4.5.1 实现基本的继承	86
4.5.2 禁止继承	88
4.5.3 继承的作用域	88
4.5.4 窗体继承的应用	89
4.6 接口	97
4.6.1 接口声明	97

4.6.2 接口的实现	97	5.5.3 ColorDialog 控件	134
4.6.3 多接口	98	5.5.4 PrintDialog 控件	136
4.7 对象的生命周期	99	5.6 时间控件	138
4.7.1 创建对象	99	5.7 图片框控件	139
4.7.2 对象的终止	101	5.7.1 常用图片框属性	139
4.7.3 Finalize 方法	102	5.7.2 使用图片框	139
4.7.4 Dispose 方法	102	小结	140
小结	102	综合练习五	140
综合练习四	103	一、选择题	140
一、选择题	103	二、填空题	140
二、填空题	103	三、思考题	141
三、思考题	103	四、上机操作	141
四、上机操作	103		
第 5 章 常用 Windows 窗体对象	105	第 6 章 高级 Windows 窗体设计	142
5.1 Windows 窗体	105	6.1 界面样式	142
5.1.1 窗体的属性	105	6.2 菜单控件	142
5.1.2 窗体的常用方法和语句	107	6.2.1 MainMenu 控件	142
5.2 控件	108	6.2.2 ContextMenu 控件	144
5.2.1 常用控件的使用方法	109	6.2.3 StatusBar 控件	145
5.2.2 新增的 Anchor 和 Dock		6.2.4 ToolBar 控件	146
控件属性	110	6.3 多文档界面程序 (MDI)	147
5.3 文本控件	111	6.3.1 创建 MDI 父窗体	147
5.3.1 文本框控件	112	6.3.2 创建 MDI 子窗体	148
5.3.2 超链接控件	115	6.3.3 确定活动子窗体	149
5.3.3 剪贴板	115	6.4 其他高级窗体控件	149
5.3.4 RichTextBox 控件	116	6.4.1 TreeView 控件	149
5.3.5 ListBox 控件	118	6.4.2 ListView 控件	151
5.3.6 CheckedListBox 控件	119	6.4.3 滚动条控件	153
5.3.7 ComboBox 控件	119	小结	153
5.4 按钮控件	120	综合练习六	154
5.4.1 Button 控件	120	一、选择题	154
5.4.2 CheckBox 控件	123	二、填空题	154
5.4.3 GroupBox 和 RadioBox 控件	124	三、思考题	154
5.5 对话框控件	127	四、上机操作	154
5.5.1 OpenFileDialog 控件	128		
5.5.2 FontDialog 控件	131	第 7 章 关系数据库基础	155
		7.1 关系数据库简介	155

7.1.1 关系模型	155
7.1.2 关系数据库 DML 的特点	156
7.2 SQL 语言基础	156
7.2.1 SQL 语言特点	156
7.2.2 SQL 数据定义	157
7.2.3 SQL 数据操作	158
7.2.4 SQL 数据控制功能	160
7.3 SQL 常用函数	161
7.3.1 系统函数	161
7.3.2 时间和日期函数	162
7.3.3 字符串函数	162
7.3.4 数学函数	163
7.3.5 集合函数	164
7.3.6 文本和图像函数	164
7.4 SQL 中的运算符	164
7.4.1 算术运算符	164
7.4.2 赋值运算符	165
7.4.3 位运算符	165
7.4.4 比较运算符	165
7.4.5 逻辑运算符	166
7.4.6 字符串运算符	166
7.4.7 一元运算符	166
7.4.8 运算符的优先顺序	167
7.5 建立 SQL 查询	167
7.5.1 SELECT 语句	167
7.5.2 简单的查询	168
7.5.3 使用 FROM 子句	168
7.5.4 使用 WHERE 子句	168
7.5.5 使用 GROUP BY 分组	169
7.5.6 ORDER BY 子句	169
7.5.7 联合查询	170
7.6 在 Visual Basic.NET 中建立查询	170
小结	173
综合练习七	174
一、选择题	174
二、填空题	174
三、思考题	174

四、上机操作	174
--------------	-----

第 8 章 ADO.NET 数据访问基础

8.1 ADO.NET 对象	176
8.1.1 ADO.NET 和 ADO 的差别	176
8.1.2 ADO.NET 的优点	177
8.2 数据存储基础	178
8.2.1 ADO.NET 对象组成	178
8.2.2 Connection 对象	178
8.2.3 Command 对象	180
8.2.4 Recordset 对象	183
8.2.5 DataSet 对象	183
8.2.6 DataReader 对象	183
8.2.7 DataView 对象	184
8.3 使用表单数据绑定	184
8.4 数据连接与访问	185
8.4.1 数据连接	185
8.4.2 读取和更新数据	185
8.4.3 数据操作	186
8.4.4 使用远程数据对象	187
8.5 XML 扩展支持	187
8.6 使用 Access 建立数据库	188
8.6.1 数据库设计	188
8.6.2 创建数据库	189
8.6.3 创建数据表	190
8.6.4 编辑数据	194
8.6.5 设置表之间的关系	195
8.6.6 在 Visual Basic.NET 中 链接数据库	197
8.7 使用 SQL Server 建立数据库	197
8.7.1 创建数据库	197
8.7.2 创建数据表	201
8.7.3 添加和删除数据	202
小结	203
综合练习八	204
一、选择题	204
二、填空题	204

三、思考题	205	10.4.4 ReaderWriter 锁	235
四、上机操作	205	10.5 等待句柄	235
第 9 章 使用 ASP.NET 开发 Web 窗体.....	206	10.5.1 互斥对象	236
9.1 Web 应用程序简介	206	10.5.2 同步事件	236
9.1.1 ASP.NET 与 ASP 的比较	206	10.6 线程计时器	237
9.1.2 Web 窗体和 Windows 窗体 比较	207	10.7 取消事务	238
9.1.3 Web 窗体的特点	208	10.8 线程的优先级	238
9.1.4 Web 窗体的创建	209	10.9 线程池	240
9.2 服务器控件	211	10.9.1 线程池的使用	240
9.2.1 HTML 服务器控件	211	10.9.2 线程池的参数和返回值	242
9.2.2 ASP.NET 服务器控件	213	10.10 死锁	243
9.2.3 HTML 控件与 Web 控件的 比较	218	小结	244
9.3 Web 服务	219	综合练习十	244
9.3.1 Web 服务的基本概念	219	一、选择题	244
9.3.2 Web Service 的体系结构	220	二、填空题	244
9.3.3 Web 服务的开发	225	三、思考题	244
小结	226	四、上机操作	244
综合练习九	226	第 11 章 Visual Basic.NET 中的 异常处理.....	245
一、选择题	226	11.1 错误类型	245
二、填空题	226	11.1.1 语法错误	245
三、思考题	226	11.1.2 执行错误	246
四、上机操作	227	11.1.3 逻辑错误	246
第 10 章 Visual Basic.NET 中的多线程	228	11.2 Visual Basic.NET 中程序的 编译与调试	246
10.1 进程和线程	228	11.2.1 程序编写	246
10.2 多线程的使用	229	11.2.2 程序编译	247
10.2.1 线程的创建	229	11.2.3 设置断点	248
10.2.2 控制线程	231	11.3 Visual Basic.NET 中的错误处理 语句结构	249
10.2.3 线程的生命周期	231	11.4 错误处理过程	251
10.3 通过多线程传送数据	232	11.4.1 添加错误处理	251
10.4 线程同步	233	11.4.2 添加简单的 Try...Catch...End Try 块	251
10.4.1 同步锁 SyncLock	233	11.4.3 确定所发生的事件	251
10.4.2 Join 方法	234	11.4.4 处理特定异常	253
10.4.3 Interlocked 类	234		

11.4.5 使用异常继承层次结构	254	小结	260
11.4.6 使用 Throw 关键字	255	综合练习十一	261
11.4.7 错误处理选项	255	一、选择题	261
11.4.8 使用 Exception 对象构造函数 ..	255	二、填空题	261
11.4.9 无条件运行代码	256	三、思考题	261
11.4.10 创建 Exception 类	257	四、上机操作	261
11.5 Visual Basic 6.0 和 Visual Basic.NET		参考文献	262
中的错误处理机制比较	259		

第 1 章 Visual Basic.NET 入门

本章要点

本章将主要介绍 .NET 和 .NET 框架的概念以及“软件作为服务”的思想，并简要介绍 Visual Basic.NET 的特点及其语言变化；通过本章的学习，应掌握 Visual Studio.NET 开发平台的安装。

1.1 .NET 开发平台

随着 Internet 的快速发展，需要实现异构操作平台和语言间的高度交互性，这就对分布式环境下的软件开发提出了更高的要求。正是在这种背景下，微软推出了 .NET 开发平台，其目的就是为了让网络应用程序可以独立于各种操作平台而实现程序间的有效沟通。微软推出的 Windows 操作平台改变了人们使用计算机的方式，而微软推出的 .NET 平台则将要改变人们使用网络的方式。如果说 Windows 系统标志着微软过去的辉煌，那么 .NET 则寄托着微软未来的希望。

1.1.1 .NET 的概念

.NET 是创建分布式 Web 应用程序的新一代开发平台，它是为了简化在第三代 Internet 网络环境下的分布式应用程序开发，在基于开放的互联网标准和协议的基础上，实现异构语言和平台的交互性。简单地说，.NET 就是可以让应用程序通过网络来相互沟通与分享彼此的资源，而不管该应用程序是由何种编程语言所编写以及在何种操作系统上运行。使用 .NET 创建的应用程序不再是某一台计算机或某些计算机所专有的，而是能够运行于整个 Internet 网络环境下的软件服务程序。这些软件服务可以被 Web 应用程序调用，或者与其他服务捆绑在一起。使用 .NET，服务和应用程序可以在任意数目的计算机上运行，并且应用程序的界面可以在不同类型的设备上显示。

显然，.NET 有以下优点：

1. 跨语言

.NET 支持多种语言的互操作，在一种语言下开发的组件通过面向对象的编程可以在另一组件下重用。这些组件既可以用 Visual Basic、C#、Visual C++ 编写的，也可以是用 Java、Pascal 等编程语言编写的。

2. 跨平台

.NET 是独立于操作系统的，它将各种语言通过使用特定的编译器生成独立于机器的微软中间语言（MSIL），同时产生运行所需要的元数据（Metadata），在代码运行时再使用即时编译器（Just In Time Compiler）生成本地平台的机器代码来执行，从而实现异构平台下对象的互操作。目前，支持 .NET 的平台不仅包括 Windows、Linux 和 UNIX 等操作系统，还包括各种支持 .NET 的智能终端。

3. 安全

.NET 中的 Runtime 提供了系统的安全服务，以保证未经授权的用户不能访问机器上的

资源, 并且代码也不能完成未授权的活动, 这大大改善了系统的安全性和可靠性, 大幅度减少了应用程序配置时来自“DLL Hell”的挑战。

4. 对开放互联网标准和协议的支持

.NET 通过对 HTTP、XML、SOAP、WSDL 等 Internet 标准和 Web 服务协议的强大支持提供在异构网络环境下的获取远程服务、连接远程设备以及完成程序间的交互。

为什么 .NET 具有如此强大的功能呢? 除了内部结构的革新外, 最重要的一点是将网络上的各种服务器服务的数据以 XML 的格式进行封装, 而在 Web 服务间则以 SOAP 协议让程序通过网络在不同平台间沟通。

1.1.2 .NET 框架简介

.NET 框架是 .NET 的核心组成部分之一, 它给程序员提供一些完整的开发工具, 也提供用于创建、部署和运行分布式网络服务 (Web 服务) 与应用程序的平台, 大家所关心的跨平台能力就是通过 .NET 框架来完成的。若仅从跨平台这方面来看, .NET 框架可类比于 Java 虚拟机 (JVM)。Visual Basic.NET 的一系列强大的功能都是从 .NET 框架中自动获得的。 .NET 框架旨在实现下列目标:

- (1) 提供一个一致的面向对象的编程环境, 而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。
- (2) 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。
- (3) 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。
- (4) 提供一个可消除脚本环境或解释环境的性能影响的代码执行环境。

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。

在整个 .NET 平台中, .NET 框架位于操作系统之上, 为位于它之上的其他部件提供服务。 .NET 框架如图 1-1 所示。

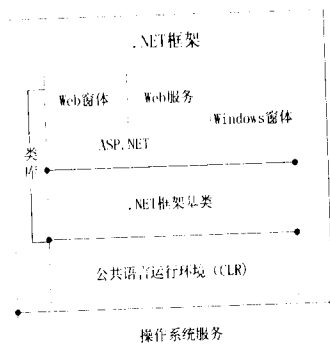


图 1-1 .NET 框架示意图

.NET 框架具有三个主要组件: 公共语言运行环境、.NET 框架类库和 ASP.NET。公共语言运行环境是 .NET 框架的基础, 用于运行和加载应用程序的软件组件, 提供很多服务来简化代码的开发和应用程序的部署, 同时在可靠性和安全性方面也提供大量的服务。可以

将运行环境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以及可确保安全性和可靠性的其他形式的代码准确性。

事实上,代码管理的概念是运行环境的基本原则。以运行环境为目标的代码称为托管代码,而不以运行环境为目标的代码称为非托管代码。.NET 框架可由非托管组件承载,这些组件将公共语言运行环境加载到它们的进程中并启动托管代码的执行,从而创建一个可以同时利用托管和非托管功能的软件环境。.NET 框架不但提供若干个运行库宿主,而且还支持第三方运行库宿主的开发。

公共语言运行环境还是.NET 平台用来实现程序代码跨平台执行的机制。它在进行程序的编译时,将程序代码分成两个部分:微软中间语言(MSIL)和记载着程序执行所需信息的元数据(Metadata)。然后再经即时编译器(JIT Compiler)进行第二次编译,将微软中间代码编译成本地代码,从而使其可在本地机器上运行。

代码编译和执行过程如图 1-2 所示。

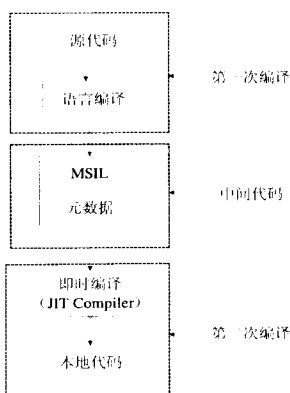


图 1-2 代码编译和执行过程

.NET 框架的另一个主要组件是类库,它是一个与公共语言运行库紧密集成的可重用的类型集合。该类库是面向对象的,并提供开发人员自己的托管代码可从中导出功能的类型。这不但使.NET 框架类型易于使用,而且还减少了学习.NET 框架的新功能所需要的时间。此外,第三方组件可与.NET 框架中的类无缝集成。正如开发人员对面向对象的类库所希望的那样,可以使用.NET 框架类库完成一系列常见编程任务(包括诸如字符串管理、数据收集、数据库连接以及文件访问等任务)。

除这些常见任务之外,类库还支持包括多种专用开发方案的类型。开发多种应用程序,这些应用程序包括控制台应用程序或图形用户界面(GUI)应用程序,也包括基于 ASP.NET 所提供的最新创新的应用程序(如 Web 窗体和 XML Web services)。

最后一个重要的组成部件是 ASP.NET。ASP.NET 建立在.NET 框架的编程类之上,它提供了一个 Web 应用程序模型,并且包含了便于生成 ASP Web 应用程序的控件集和结构。ASP.NET 包含封装公共 HTML 用户界面元素(如文本框和下拉菜单)的控件集。但这些控件在 Web 服务器上运行,并以 HTML 的形式将它们的用户界面传送到浏览器。在服务器上,这些控件公开一个面向对象的编程模型,为 Web 开发人员提供了面向对象编程的灵活性。

ASP.NET 还提供结构服务（如会话状态管理和进程回收），进一步减少了开发人员必须编写的代码量，也提高了应用程序的可靠性。

另外，ASP.NET 使用这些概念使开发人员能够以服务的形式交付软件。使用 XML Web services 功能，ASP.NET 开发人员可以编写自己的业务逻辑，并使用 ASP.NET 结构通过 SOAP 交付该服务。

1.1.3 “软件作为服务”的思想

在 Internet 迅速普及的今天，为了避免网络出现信息孤岛的现象，需要解决的一个突出问题就是各个网络应用程序之间的沟通问题，这就使得 Web 服务的概念应运而生。.NET 的设计目标就是让“软件作为服务”，构建在.NET 平台上交付最终用户使用的产品，如 Office.NET 等就深刻体现了微软“软件作为服务”的思想。“软件作为服务”的另一个方面，就是在网络上能够公开可编程接口，并通过这种方式来共享站点的服务。

1.2 Visual Basic.NET 简介

Visual Basic.NET 是微软.NET 体系中的一种快速开发工具，是新一代的 Visual Basic。千万不要认为 Visual Basic.NET 只是简单地在 Visual Basic 6.0 的基础上增加了一些新特性，实际上，它是微软重新打造的新产品。它能比过去的 Visual Basic 版本更容易地编写分布式应用程序，例如：Web 和企业的 n 层系统。Visual Basic.NET 具有两个新的窗体包（Windows 窗体和 Web 窗体），还有用于访问断开连接的数据源的 ADO 新版本——ADO.NET。开发人员使用 Web 窗体和 ADO.NET 可快速开发出灵活的 Web 站点。

Visual Basic.NET 比 Visual Basic 6.0 功能更强大，并且增加了一些只能使用 C++ 类语言才能访问系统资源的功能。为了使 Visual Basic 开发人员从面向对象的设计中获益，简化企业 Web 应用程序的开发，Visual Basic.NET 还针对 Visual Basic 对面向对象程序设计支持欠佳的情况，实现了包括继承在内的所有的面向对象的特性，从而使得 Visual Basic.NET 一跃成为完全的面向对象的语言。利用这些新的语言特性，Visual Basic.NET 将迅速有效地提供开发企业关键性应用程序所需的所有强大功能，而同时保持了使其成为世界上最流行的开发工具的易用性。

另外，Visual Basic.NET 还可与其他 Visual Studio.NET 语言完全集成。程序开发人员不仅可以用不同的编程语言开发应用程序组件，而且可以通过跨语言继承用其他编程语言编写的类。通过使用统一的编译器，可以调试用多个语言编写的应用程序，而不要考虑该程序是在本地计算机还是在远程计算机上运行。

1.2.1 Visual Basic.NET 的特点

Visual Basic.NET 是一种面向对象的编程语言，它不仅继承了 Visual Basic 6.0 的优点，而且还增加了许多新的特性，从而大大提高了程序开发人员的效率。主要特点如下：

1. 可视化设计

Visual Basic.NET 为用户提供了大量的控件对象，如菜单、对话框等，用户只需利用鼠标、键盘把这些控件对象拖动到适当的位置，设置它们的大小、形状和属性等，就可以设计出所需要的应用程序界面。

2. 面向对象编程

尽管从 Visual Basic 4.0 就开始支持面向对象的程序设计，但那时的 Visual Basic 还不能成为面向对象的编程语言。而 Visual Basic.NET 在原来 Visual Basic 封装性的基础上又增加了多态性、继承性等面向对象编程的特性，从而使得 Visual Basic.NET 具备面向对象语言的一切特性，成为完全面向对象的编程语言。

3. 事件驱动编程

Visual Basic.NET 的运行是由事件驱动的，从而使得程序的运行是按照用户的意图或系统事件进行，而不是顺序执行的。比如用鼠标单击一个文件夹，则 Visual Basic.NET 会跳去执行一次打开操作。单击鼠标的过程就可以称为一个事件，由该事件驱动了一次有关打开部分程序的运行，从而完成了一次打开操作。

4. 统一的集成开发环境 (IDE)

Visual Basic.NET 提供了统一的集成开发环境。在该环境下，用户可以自由选择自己喜爱的语言编写各种应用程序。

1.2.2 Visual Basic.NET 中的语言变化

1. 继承

Visual Basic.NET 允许为类定义基类。派生类继承并可以扩展基类的属性和方法，还可以通过重载来覆盖基类的方法和属性。继承是 Visual Basic.NET 中新增的一个特性，它的一个重要目标就是实现代码重用。尽管在以前的 Visual Basic 版本中也支持代码重用，但都是以组合代码的形式创建和重用 COM 组件的。

```
Class Father
    Sub Method()
        .....
    End Sub
End Class
Class Child
    Inherits Father
    .....
End Class
```

2. 多态性和重载

多态性是在一个类中重新定义一个函数的过程，通过使用关键字 Overrides 实现。而重载则是定义同名但使用不同数据类型的属性、方法或过程的能力。

3. 结构化异常处理

Visual Basic.NET 通过使用增强版本的 Try...Catch...Finally 语法支持结构化异常处理。有了 Try...Catch...Finally 语法，开发人员可以嵌套异常处理，也可以用它们编写在正常条件和异常条件下都能运行的执行清洁代码的控制结构。

4. 构造函数和结构函数

构造函数允许创建一个类的新实例的同时向这个实例传递参数。构造函数对于面向对象编程是相当重要的，因为它允许用户定义的构造代码通过实例的创建来传递参数。而析构函数则是用于释放类实例不再使用的系统资源的方法。Visual Basic.NET 通过使用 Sub New 和 Sub Finalize 或 Sub Dispose 方法支持构造函数和析构函数。Finalize 析构函数是只能从其所属类或派生类调用的受保护的方法，而 Dispose 析构函数则是一个公开的方法，

可以在任何时候调用。

5. 新增的数据类型

Visual Basic.NET 中引入了三个新的数据类型——Char、Short 和 Decimal。Char 数据类型是无符号的 16 位整数，用于存放 Unicode 字符，它等效于 .NET 框架中的 System.Char 数据类型。Short 数据类型为 16 位整数，是较早版本中的 Integer 的替代。Decimal 数据类型是 96 位十进制值。

6. 接口

与类不同的是，接口只描述类的属性和方法，而不提供实现。尽管在以前的 Visual Basic 版本中也使用接口，但不能创建接口，而在 Visual Basic.NET 中可用 Interface 关键字声明创建接口，并用 Implements 关键字实现这些接口。

7. 委托

委托是代表自己调用对象方法的对象，可以通过新增的关键字 Delegate 定义委托。每个委托实例都将进一步调用特定对象的某个方法，而对象和方法的选择则是在构造委托实例时决定的。委托可以用于事件处理和自由线程中。

8. 共享成员

共享成员是指那些可以被类的所有实例共享的数据和函数成员。一个共享成员独立地存在于类的每个实例中。在 Visual Basic.NET 中用 Shared 关键字定义共享成员。共享成员可以通过类实例或类名来访问。

9. 多线程

在 Visual Basic.NET 中，可以编写独立执行多个任务的应用程序。通过多线程，开发人员可以生成一个线程来完成一些运行时间长的任务、执行一个复杂的查询或运行一个多部分的计算，而应用程序的其他部分仍继续执行。

10. 命名空间

命名空间通过将类、接口和方法组织出层次结构来避免命名冲突。命名空间实际上是建立在整个 .NET 框架上的一批类，每种 .NET 编程语言都使用这些命名控件，其好处是可以让程序开发人员根据功能快速定位到相关类型。.NET 的框架类库就是通过命名空间把庞大的类按功能组织得有条不紊。

11. 部件

部件代替了类型库，构成了基于 .NET 的应用程序的部署、重用和安全授权的基本单元，它能够描述特定组件或应用程序所需要的所有文件。部件可以以可执行（.exe）文件或动态链接库（.dll）文件的形式出现，用来向公共运行环境提供了解类型实现所需要的信息。

12. 引用

引用允许使用定义于其他部件中的对象。在 Visual Basic.NET 中，引用指向部件，而不是类型库。

1.3 Visual Basic.NET 安装

Visual Basic.NET 作为 Visual Studio.NET 开发平台的一部分，它的安装和配置是包含在整个 Visual Studio.NET 的安装和配置中的。相比于以前的 Visual Studio 6.0，Visual Studio.NET 安装所需的软件、硬件环境高出很多。安装时间也较长，不过安装过程并不复