

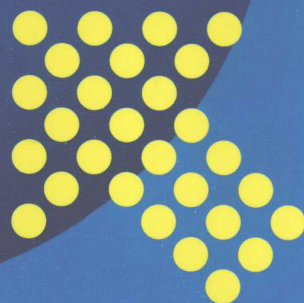
21世纪高等学校规划教材



Visual C#.NET YUANLI YU SHIWU

Visual C#.NET 原理与实务

夏敏捷 主编



中国电力出版社
<http://jc.cepp.com.cn>

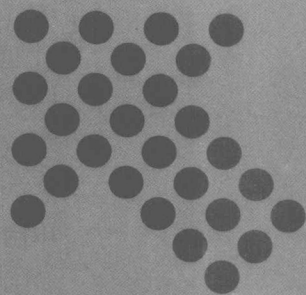
21世纪高等学校规划教材



Visual C#.NET YUANLI YU SHIWU

Visual C#.NET 原理与实务

主 编 夏敏捷
主 审 陈海兵



中国电力出版社

<http://jc.cepp.com.cn>

内 容 提 要

本书为 21 世纪高等学校规划教材。本书最大的特色在于以大家熟悉的“类联众游戏大厅的网络象棋”开发案例为主线，贯穿整本书。原理篇部分以案例的形式向读者展示了 Visual C#.NET 的编程精髓，将 Visual C#.NET 编程中的语法、技巧等方面的知识融会其中，从浅到深，从易到难。书中内容实例丰富、讲解清晰、力避代码复杂冗长。简短的案例特别有助于初学者仿效理解、把握问题的精髓，能够帮助读者快速对 Visual C#.NET 有整体的认识。综合应用案例篇通过“类联众游戏大厅的网络象棋”综合性案例，配以详细的代码说明和图片解释，使读者在掌握原理篇中提供的技术或技巧后，使用这些技术尝试实现更为复杂的功能，在实践中掌握和领会 Visual C#.NET 编程的基本技巧和思想精髓。

本书可作为高等院校相关课程的教材使用，也适合初学者和具有一定编程经验的 Visual C#.NET 用户使用，还可供广大计算机工作者、软件开发者和游戏编程爱好者参考。

图书在版编目 (CIP) 数据

Visual C#.NET 原理与实务 / 夏敏捷主编. —北京：中国电力出版社，2010.7

21 世纪高等学校规划教材

ISBN 978-7-5123-0544-1

I. ①V… II. ①夏… III. ①C 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 (2010) 第 111491 号

中国电力出版社出版、发行

(北京三里河路 6 号 100044 <http://jc.cepp.com.cn>)

北京市同江印刷厂印刷

各地新华书店经售

*

2010 年 8 月第一版 2010 年 8 月北京第一次印刷

787 毫米×1092 毫米 16 开本 25.5 印张 625 千字

定价 40.80 元

敬告读者

本书封面贴有防伪标签，加热后中心图案消失
本书如有印装质量问题，我社发行部负责退换

版 权 专 有 翻 印 必 究

前言

DotNet (.NET) 是微软未来的技术发展方向,其强大的技术优势已经渐渐为人们所推崇,并且在全世界也掀起了学习 DotNet 技术的高潮,掌握该技术,无疑为目前激烈的就业竞争获得了有力武器。作为微软 DotNet 框架下的核心技术之一, Visual C#.NET 经过几年的发展,已经成为主流的开发语言。

本书作者长期从事 Visual C#.NET 教学与应用开发,在长期的工作学习中,积累了丰富的经验和教训,能够了解在学习编程的时候需要什么样的书才能提高 C# 开发能力,即以最少的时间投入得到最多的实际应用知识。

本书最大特色是围绕 Visual C#.NET 综合性案例——“类联众游戏大厅的网络象棋”展开,每一章都最终形成与综合案例相关的所需要的模块。

本书内容包括两篇共 10 章。

第一篇 Visual C#.NET 开发技术原理

第 1 章主要介绍了 .NET 框架和 Visual Studio2008.NET 集成开发环境,同时以案例形式介绍面向对象的基本概念,包括类和对象以及需要重点掌握的面向对象的继承性、多态性思想和具体体现。本章根据类的思想设计出综合性案例设计部分需要的象棋棋子类和棋盘类。

第 2 章主要介绍常用控件,同时向大家展示用 Windows 窗体来编写程序的特点以及技巧。本章利用控件数组设计出综合性案例设计部分需要的象棋游戏界面及联众大厅。

第 3 章介绍了在 Visual C#.NET 语言中,利用 .NET 框架提供的一整套相当丰富的类库绘制各种图形、处理位图图像和视频,从而建立多媒体程序。本章最后利用图形技术完善棋子类和棋盘类。

第 4 章主要介绍了 Visual C#.NET 语言提供的用于文件操作的类,以及如何利用它们实现对文件的存储管理和对文件的读写等各种操作。本章最后设计出基于文件形式的象棋复盘程序。

第 5 章主要介绍利用 .NET 框架类库中提供的应用层类 TCPClient、TCPListener 和 UDPClient 类来实现 Socket 编程的知识。本章最后通过应用层类开发出基于 UDP 的 P to P (俗称 P2P) 网络中国象棋。

第 6 章主要介绍组件的相关概念,以及在 .NET 框架环境下如何编写组件,向组件添加域、属性以及事件;制作自己的组件和控件的一些方法和技巧。本章最后设计出综合性案例所需的游戏房间控件。

第 7 章在 ADO.NET 模型的基础上介绍如何操作数据库,学习本章后读者可以熟练掌握 ADO.NET 中各种对象的操作方法以及常用的 SQL 语句,并能够进行读、写、检索 SQL Server 数据库。还能利用控件通过数据绑定技术显示数据库中的数据。本章最后设计出综合性案例可以使用的登录模块。

第 8 章主要介绍了开发 Web 应用程序的 ASP.NET 工作原理和 ASP.NET 常用控件,在 Web 应用程序中访问数据库等。本章最后创建综合性案例所需的会员注册和留言所需的网络

象棋游戏网站。

第9章介绍.NET框架中与XML相关的命名空间和其中的重要类,并用实例使读者更进一步地了解XML文件的C#读写操作的具体方法。本章最后设计出网络象棋游戏网站会员所需的留言板。

第二篇 Visual C#.NET 综合应用案例

通过第10章的“类联众游戏大厅的网络象棋”的综合性应用案例,详细介绍了系统的开发过程。并按照软件开发方法与流程,从需求分析到编码实施,让读者真正体会到软件项目的开发思想,同时又对上一篇的知识进行了巩固与提高。

本书中列举了大量的程序案例,均提供详细的关键技术分析、设计思路以及具体的解决步骤方案,使读者跳出枯燥的理论学习,在享受程序设计的乐趣中掌握使用Visual C#.NET开发应用程序的方法。而且将本书中的案例加以修改、综合即可应用到程序的开发实践中。本书所有代码可以到程序员网站<http://download.csdn.net>的下载区按书名下载。

需要说明的是:学习编程是一个实践的过程,而不仅仅是看书、看资料的过程,亲自动手编写、调试程序才是至关重要的。通过实际的编程以及积极的思考,读者可以很快地掌握很多的编程技术,而且,在编程中读者会积累许多宝贵的编程经验。在当前的软件开发环境下,这种编程经验对开发者尤其显得不可或缺。

本书由夏敏捷(中原工学院)主持编写,李娟编写第5章,吴婷和马宗梅编写第8、9章,夏敏捷编写第1、2、3、4、6、7和10章。在本书的编写过程中,为确保内容的正确性,我们参阅了很多资料,并且得到了中原工学院计算机系老师和资深Web程序员的支持。在此谨向他们表示衷心的感谢。本书由陈海兵主审。

由于编者水平有限,书中难免有错,敬请广大读者批评指正,在此表示感谢。

夏敏捷

2010年5月

目 录

前言

第一篇 Visual C#.NET 开发技术原理

第 1 章 Visual C#.NET 基础	1
1.1 .NET 框架概述	1
1.2 栈模型的实现	10
1.3 从 Shape 抽象类派生出具体形状类	23
1.4 扑克牌发牌程序	28
1.5 综合性案例设计部分——中国象棋棋子类和棋盘类设计	34
第 2 章 Visual C#.NET 控件及其应用	48
2.1 模拟单项选择题测试	48
2.2 模拟多项选择题测试	51
2.3 小学生算术练习程序	53
2.4 图片自动浏览器	55
2.5 利用滚动条控件调配颜色	58
2.6 TreeView 控件建立学校系部分层列表	60
2.7 ListView 控件显示学生信息	63
2.8 计算器程序设计	65
2.9 扫雷游戏	68
2.10 综合性案例设计部分——控件数组实现象棋游戏界面	74
2.11 综合性案例设计部分——联众大厅界面设计	86
第 3 章 图形图像和多媒体编程	90
3.1 图形图像和多媒体相关类概述	90
3.2 可擦写图形轮廓的实现	91
3.3 拼图游戏设计	98
3.4 特殊形状的窗体界面	103
3.5 制作媒体播放器	106
3.6 七巧板游戏	109
3.7 绘图软件	120
3.8 贪吃蛇游戏	133
3.9 坦克大战游戏	147
3.10 综合性案例设计部分——完善棋子类和棋盘类	160

第 4 章 文件操作	168
4.1 文件操作相关类概述	168
4.2 文件夹浏览器设计	168
4.3 通讯录程序	175
4.4 文件分割合并器	179
4.5 综合性案例设计部分——基于文件形式的象棋复盘程序	185
第 5 章 网络编程	198
5.1 网络通信编程基础	198
5.2 实现网段扫描	204
5.3 基于 TCP 协议的局域网通信程序	210
5.4 综合性案例设计部分——基于 UDP 的 P2P 网络中国象棋	220
第 6 章 组件设计	236
6.1 字符串逆序和排序的简单组件	236
6.2 汉字验证码图片控件	239
6.3 渐变背景椭圆按钮控件	246
6.4 综合性案例设计部分——游戏房间控件	249
第 7 章 数据库编程	257
7.1 ADO.NET 数据库访问技术	257
7.2 按照姓名模糊查询程序	262
7.3 单项选择题的考试系统	266
7.4 电子日记软件开发	271
7.5 学生图像采集存储系统	279
7.6 综合性案例设计部分——会员登录	284
第 8 章 Web 应用程序开发	288
8.1 ASP.NET 基础	288
8.2 统计网站在线人数	292
8.3 创建可以点击的 Web 地图	295
8.4 仿 Windows 界面的选项卡页面	298
8.5 网络硬盘开发	300
8.6 新闻显示处理页面	313
8.7 学生信息网上管理系统	316
8.8 综合性案例设计部分——创建网络象棋游戏网站	322
8.9 综合性案例设计部分——创建象棋游戏网站会员注册页面	329
第 9 章 XML 技术	337
9.1 表格形式显示 XML 文件	337
9.2 使用 ADO.NET 中 DataSet 创建 XML 文件	341
9.3 使用 ADO.NET 中的 DataSet 读取 XML 文件	342
9.4 RSS 新闻阅读器制作	344
9.5 综合性案例设计部分——基于 XML 的象棋游戏网站留言板	353

第二篇 Visual C#.NET 综合应用案例

第 10 章	类联众游戏大厅的网络象棋.....	360
10.1	功能需求分析.....	360
10.2	系统设计.....	360
10.3	客户端功能模块的实现.....	365
10.4	服务器端功能模块的实现.....	384
10.5	小结.....	399

第一篇 Visual C#.NET 开发技术原理

第 1 章 Visual C#.NET 基础

.NET 是微软开发的一种面向网络、支持各种用户终端的开发平台。Visual Studio2008.NET 环境是 Microsoft 为实现其 .NET 技术的开发工具,在构建基于 Windows 平台的各种程序时,它为程序员提供了非常出色的开发环境。利用 Visual Studio2008.NET 用户可以非常轻松地创建具有自动伸缩能力的可靠的应用程序和组件。

Visual Studio2008.NET 集成开发环境包含 Visual Basic.NET、Visual C++.NET、Visual C#.NET 和 Visual J#.NET 四种编程语言,并允许它们共享工具,有助于创建混合语言解决方案。

Visual C#.NET 是微软公司针对 .NET 平台推出来的一门新语言,作为 .NET 平台的第一语言,也是微软公司推出的下一代主程序开发语言。Visual C#.NET (简称 C#语言)它几乎集中了所有关于软件开发和软件工程研究的最新成果。如面向对象、类型安全、组件技术、自动内存管理、跨平台异常处理、版本控制、代码安全管理等。它在设计、开发程序界面的时候和以前的某些程序开发语言有所不同。它既有 Visual Basic (本书简称 VB)快速开发的特点,又不乏 C++语言强大的功能。所以 C#很可能成为将来最主要的软件开发语言。

1.1 .NET 框架概述

Visual C#.NET 提出了很多新的功能、概念和观点,掌握 Visual C#.NET 不仅要掌握语法,还需要理解并运用这些新的功能、概念和观点。当然也需要掌握 .NET 框架,理解 CLR (Common Language Runtime, 公共语言运行时)。

.NET 框架 (Framework) 是一组用于帮助开发应用程序的类库集。Visual Studio.NET 开发平台需要此类库集的支持,用 Visual Studio.NET 开发的程序也需要此类库集的支持。.NET 框架的体系结构如图 1-1 所示。

1.1.1 公共语言运行时

1. 公共语言运行时 (CLR)

CLR 是 .NET Framework 的基础,提供诸如内存管理、线程管理和远程处理等核心服务,强制实施严格的类型安全以及确保安全性和可靠性的其他形式的代码的准确性。事实上,代码管理的概念是运行时的基本原则。以运行时为目标的代码称为托管代码,而不以运行时为目标的代码称为非托管代码。

2. 即时 (Just-In-Time) 编译

可以直接将源程序编译为 .EXE 或者 .DLL 文件,但是编译出来的程序代码并不是 CPU 能直接执行的机器代码,而是一种中间语言 (Intermediate Language, IL) 代码,在 IL 代码被调用执行时,CLR 的 Class Loader 会将需要的 IL 代码装入内存,然后再通过即时 (Just-In-Time)

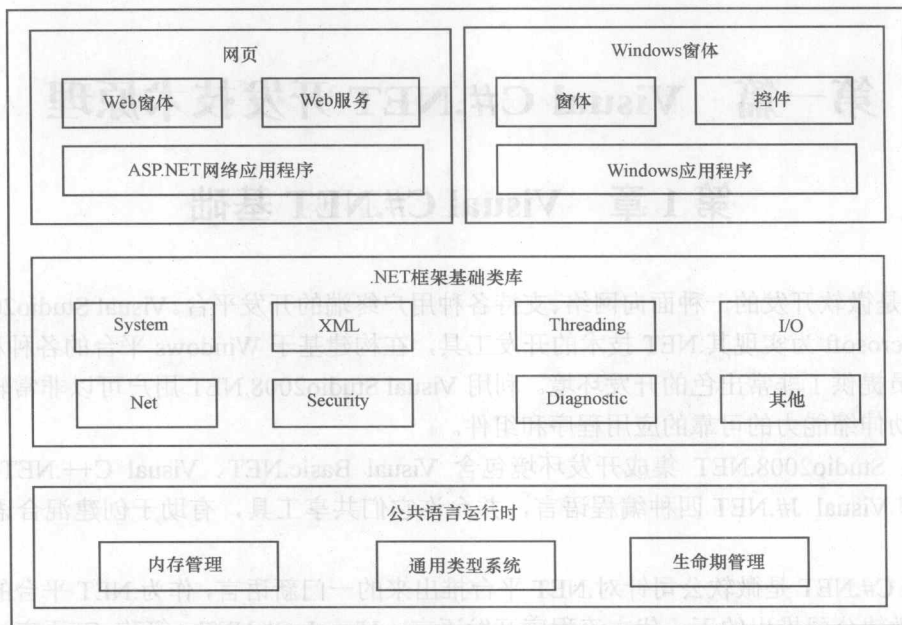


图 1-1 .NET 框架的体系结构

编译方式将其临时编译成所用平台的 CPU 可直接执行的机器代码（二进制文件）。我们看到的程序运行，其实是经过 JIT 编译后的二进制文件在执行。这时可能有很多朋友会问，这种二次编译是否多此一举，具体实现中是否很麻烦，是否会影响程序的运行速度？二次编译的确是影响了程序运行的速度，但它却为实现跨平台带来了可能。其实这种编译过程犹如 Java 中的 JVM（Java 虚拟机）。正是 JVM 才使得 Java 能够开发出跨平台的应用程序。二次编译是 CLR 在 .NET 框架下自动实现 IL 文件到二进制文件转变的，它是自动完成的，并不需要人员的参与，所以它并不会给程序执行带来麻烦。

3. 垃圾回收机制

在应用程序开发过程中，内存管理曾经是一件令人痛苦的事情。由于内存管理不科学，应用程序将不断消耗系统资源并最终导致操作系统崩溃。.NET 通过垃圾回收器来管理程序进程中分配的内存，这样，开发人员就不用进行枯燥的工作来保证将所有分配的内存都正确地释放给系统了。

1.1.2 基础类库

类库是一个由 .NET Framework SDK（Software Development Kit，软件开发工具包）中包含的类、接口和值类型组成的库。该库提供对系统功能的访问，是建立 .NET Framework 应用程序、组件和控件的基础。

类库（Class Library）提供了 Internet 和企业级开发所需要的各种功能，完全支持 Web 标准及其应用，而且使用简单，扩充方便。类库分配到不同的命名空间下，System 命名空间是 .NET Framework 中基本类型的根命名空间。

1.1.3 .NET 框架编程模型

.NET 框架编程模型和传统编程模型有所不同，如图 1-2 所示。

传统的编程模型是上层应用直接依附在操作系统之上的。例如在 Windows 中编程，程序运行于 Windows 之上。但在 .NET 平台中，.NET 框架位于操作系统与上层应用之间，上层应用创建于 .NET 框架之上，而不像过去一样直接运行在操作系统之上。正是在操作系统和应用程序之间有了 .NET 框架，才使应用程序的平台无关性成为可能。

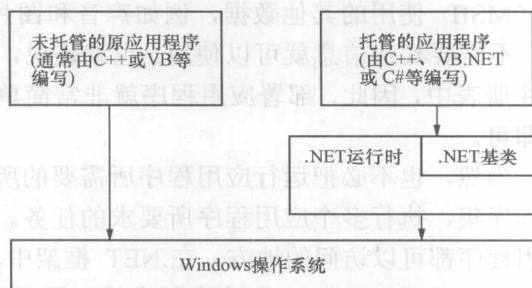


图 1-2 .NET 框架编程模型

.NET 框架是 Microsoft 为开发应用程序创建的一个富有革命性的新环境。这句话最有趣的地方是它的含糊不清，但这是有原因的。注意这句话没有说“在 Windows 操作系统上开发应用程序”。尽管 .NET Framework 发布的第一个版本是运行在 Windows 操作系统上，但是以后将推出可运行在其他操作系统上的版本，这些操作系统包括 FreeBSD、Linux 和 Macintosh，甚至个人数字助理类设备。这就是说 .NET 具有平台独立性，是可移植的，这的确是一个很大的突破。我们知道微软为了保护其 Windows 操作系统的利益，向来在平台独立问题上非常保守，其开发的产品都只能运行在 Windows 环境中，这一直为人们所诟病。

要进一步理解 .NET 框架编程模型，还需要认识两个新的概念：MSIL 和 JIT。下一节将对这两个概念作详细介绍。

1.1.4 .NET 程序的编译与运行

1. MSIL 和 JIT

在编译使用 .NET 框架创建的代码时，不是立即创建操作系统特定的本机代码，而是把代码编译为微软中间语言（Microsoft Intermediate Language, MSIL）代码，这些 MSIL 代码不专用于任何一种操作系统，也不专用于任何一种语言，有些类似于 Java 的字节码。C# 及其他 .NET 语言，如 VB.NET 在编译阶段都编译为这种语言。

因为代码在编译阶段没有直接编译成本机代码，所以在执行应用程序时，必须完成更多的工作，这就是 Just-In-Time (JIT) 编译器的任务。

JIT 把 MSIL 编译为专用于某种操作系统和目标机器结构的本机代码，只有这样，操作系统才能执行应用程序。这里编译器的名称 Just-In-Time，反映了 MSIL 仅在需要时才编译的特性。

过去，常常需要把代码编译为几个应用程序，每个应用程序用于特定的操作系统和 CPU 结构，这通常是一种优化形式（例如，为了让代码在 AMD 芯片上运行得更快），但更多时候是必须的（例如，分别运行在 Windows 和 Linux 操作系统上）。现在就不必要了，顾名思义，JIT 编译器使用 MSIL 代码，而 MSIL 代码是独立于机器、操作系统和 CPU 的。目前有几种 JIT 编译器，每种编译器都用于不同的结构，而且总能找到一个合适的编译器创建所需的本机代码。这样，用户需要做的工作就比较少了。实际上，用户不必考虑与系统相关的细节，只需要把注意力放在代码的功能上就足够了。

2. 程序集

在编译应用程序时，创建的 MSIL 代码存储在一个程序集中，程序集包括可执行的应用程序文件（这些文件可以直接在 Windows 上运行，不需要其他程序，其扩展名为 .exe）和其他应用程序使用的库（其扩展名是 .dll）。

除了包含 MSIL 外,程序集还包含元数据(即程序集中包含的数据信息)和可选的资源(MSIL 使用的其他数据,例如声音和图片文件)。元数据允许程序集是完全自我描述的,不需要其他信息就可以使用。也就是说,不再需要把应用程序所需要的数据添加到系统注册表中,因此,部署应用程序就非常简单的了,只需把文件复制到远程计算机中的目录下即可。

当然,也不必把运行应用程序所需要的所有信息都安装到一个地方。我们可以编写一些程序集,执行多个应用程序所要求的任务。此时,通常把这些可重用的程序集放在所有应用程序都可以访问的地方。在.NET 框架中,这个地方是“全局程序集高速缓冲存储器”(Global Assembly Cache),它有相应的工具可以把程序集放在高速缓冲存储器中。

3. 托管代码

代码编译为 MSIL,再用 JIT 编译器把它编译为本机代码后,CLR 的任务还没有完全完成。用.NET 框架编写的代码在执行时是托管的,即 CLR 管理着应用程序,其方式是管理内存和处理安全性,以及允许进行跨语言调试等。相反,不在 CLR 控制之下运行的应用程序是非托管的,某些语言如 C++可以用于编写这类应用程序。例如,访问操作系统的低级功能。使用 C#主要编写在托管环境下运行的代码,它们使用 CLR 的托管功能,让.NET 与操作系统进行交互,当然也可以编写在非托管环境下运行的代码,但需要特别标注。

图 1-3 所示的是传统的代码编译与运行过程,图 1-4 所示的是.NET 中代码的编译和运行过程。

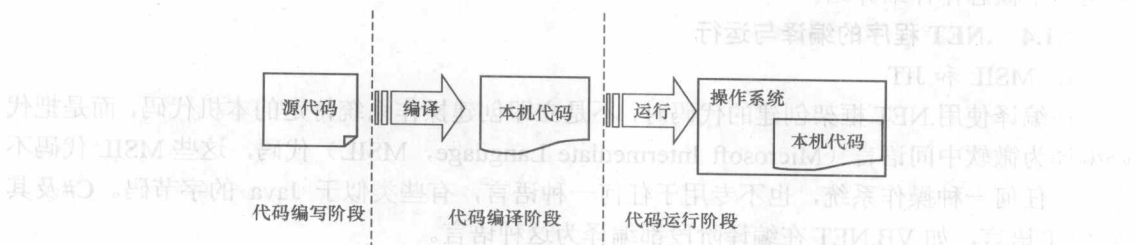


图 1-3 传统的代码编译与运行过程

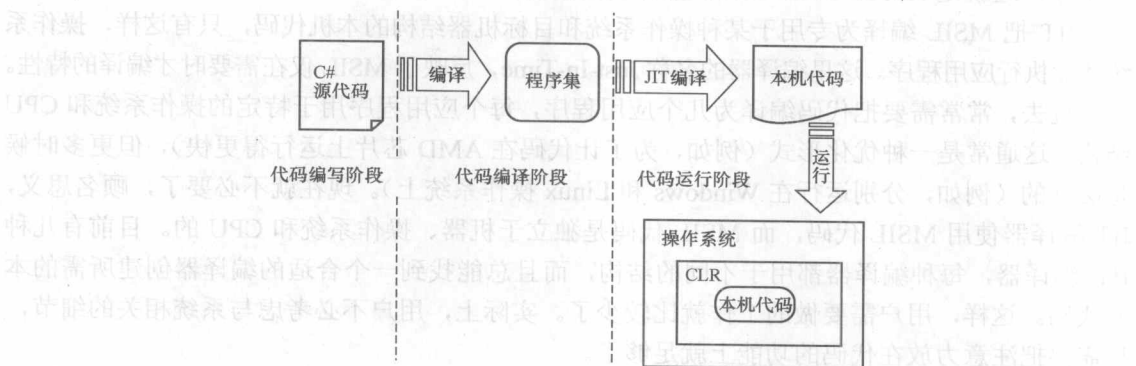


图 1-4 .NET 中的代码编译与运行过程

4. 无用存储单元收集

托管代码最重要的一个功能是无用存储单元收集 (Garbage Collection) (也称垃圾回收),

它可确保应用程序不再使用某些内存时,这些内存就会被完全释放。在.NET 推出以前,这项工作主要由程序员负责,代码中的几个简单错误会把大块内存分配到错误的地方,使这些内存神秘失踪。这通常意味着计算机的速度逐渐减慢,最终导致系统崩溃。

.NET 无用存储单元收集器会频繁检查计算机内存,从中删除不再需要的内容。它可能一秒钟内进行上千次的检查,也可能几秒钟检查一次,或者随时进行检查,但可以肯定的是它进行了检查。

1.1.5 Visual Studio2008.NET 集成开发环境

Microsoft 一直以向开发者提供最有生产力和最佳使用体验的开发工具而闻名。Visual Studio.NET 是 Microsoft 集成开发环境 (IDE) 的一个集大成者,它是一套完整的开发工具集,用于生成 ASP.NET Web 应用程序、XML Web Service、桌面应用程序和移动应用程序。Visual Basic、Visual C++、C#和 Visual J++全都使用相同的集成开发环境 (IDE),利用 IDE 可以共享工具且有助于创建混合语言解决方案。

Visual Studio.NET 极大简化了各类应用程序的开发,极大地提高了开发效率和为开发健壮的应用程序提供了可靠保障。

到目前 Visual Studio.NET 共发布了四个版本,即 Visual Studio2002.NET、Visual Studio2003.NET、Visual Studio2005.NET 和最新发布的 Visual Studio2008.NET。

2002 年,Microsoft 在以前集成开发环境 (IDE) 的基础上,伴随着 .NET Framework 1.0 的发布,正式推出了新一代的 IDE——Visual Studio2002.NET。

2003 年 4 月 23 日,微软公司推出 .NET Framework 1.1 和 Visual Studio2003.NET。这些重量级的产品都是针对 .NET 1.0 的升级版本。

2005 年,微软发布了 Visual Studio2005.NET,这个版本是面向 .NET 框架 2.0 的。它同时也能开发跨平台的应用程序,如开发使用微软操作系统的手机程序等。

最新发布的 Visual Studio2008.NET 是面向 Windows Vista、Office 2007、Web 2.0 的下一代开发工具,代号“Orcas”,经历了大约 18 个月的开发,是对 Visual Studio2005.NET 一次及时、全面的升级。

Visual Studio2008.NET 整合了对象、关系型数据、XML 的访问方式,语言更加简洁。使用 Visual Studio2008.NET 可以高效开发 Windows 应用。设计器中可以实时反映变更,XAML 中的智能感知功能可以提高开发效率。同时,Visual Studio2008.NET 支持项目模板、调试器和部署程序。Visual Studio2008.NET 可以高效开发 Web 应用,集成了 AJAX1.0,包含 ASP.NET AJAX 项目模板,它还可以高效开发 Office 应用和 Mobile 应用。

Visual Studio2008.NET 集成开发环境(如图 1-5 所示)与以前的 Visual Studio 集成开发环境非常类似,提供了以更方便、更可靠的方式来开发应用程序的工具。

1. Visual Studio2008.NET 的新特性

Microsoft 成功推出了 .NET3.5,同时将 Visual Studio2008 推向历史舞台。Visual Studio2008 在主要功能上与 Visual Studio2005 差别不大,但是在易用性、方便性、应用程序类型上做了不少改进,其主要新特性如下所列。

(1) .NET 框架从 .NET2.0 升级到 .NET3.5,同时还可以根据需要进行不同的 .NET 版本,包括 .NET2.0、.NET3.0、.NET3.5,借此创建不同 .NET 环境的应用程序。

(2) 新增设置迁移功能,可以将已有的 Visual Studio2005 中的参数设置迁移到 Visual

Studio2008 中。

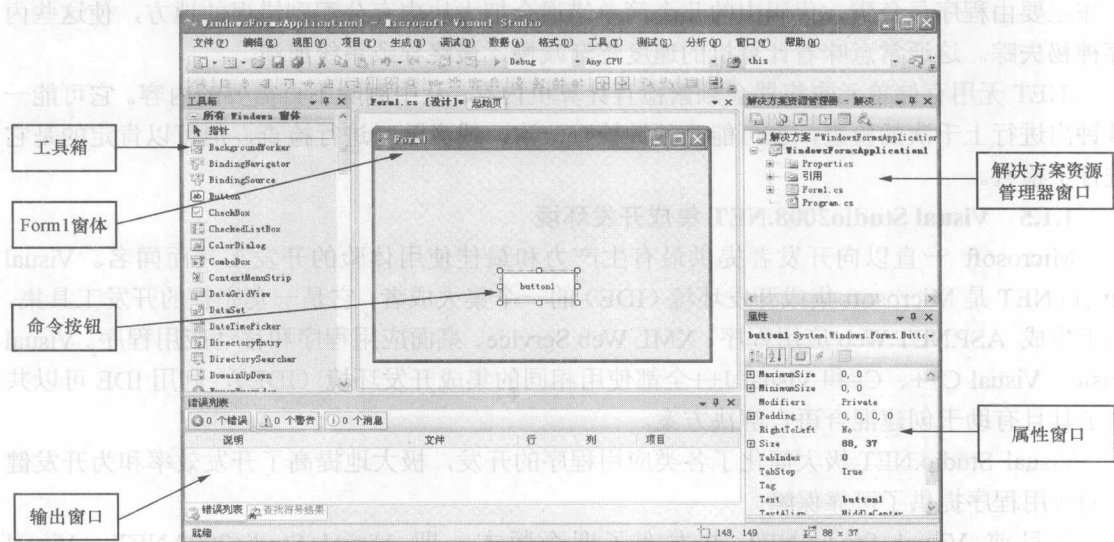


图 1-5 Visual Studio2008.NET 集成开发环境

(3) 新增 WPF、WCF、WWF 应用程序的创建向导。

(4) 增加了 WPF 设计器，方便开发 WPF 应用程序。

(5) 新增语言集成查询 (LINQ)，可以将查询语句与 C# 和 VB.NET 集成，提高了数据查询的开发效率和执行速度。

除了上面这些之外，Visual Studio2008 的改进还体现在报表应用程序项目、AJAX 开发等重要功能上，同时在 IDE 的外观、性能、操作一致性上都有所改进。

2. Visual Studio.NET 简介

(1) 解决方案资源管理器 (Solution Explorer)。使用 Visual Studio.NET 开发的每一个应用程序叫解决方案，每一个解决方案可以包含一个或多个项目。一个项目通常是一个完整的程序模块，同时可以有多个项。“解决方案资源管理器”子窗口显示 Visual Studio.NET 解决方案的树型结构。在“解决方案资源管理器”中可以浏览组成解决方案的所有项目和每个项目中的文件，可以对解决方案的各元素进行组织和编辑。

一个项目中有非常多的文件，但是大部分文件，比如 Properties 文件夹中的文件，不需要开发者进行直接编辑。开发者所要做的是双击解决方案资源管理器中的 Properties 文件夹图标，打开属性编辑器来对项目进行配置。现在并不需要对项目属性作任何修改。

解决方案资源管理器标题栏下方有 工具栏，单击“所有文件” 图标按钮后，可见 bin 文件夹和 obj 文件夹，它们在生成项目执行文件的时候会用到。其中，obj 文件夹中存放的是用来创建最终可执行文件的中间代码 (MSIL)。文件夹 bin 中存放的是“二进制”文件或者应用程序最终的编译版本。要注意的是，把 MSIL 代码称作二进制文件可能会带来误解，因为最终转换为二进制是在运行程序时由 Just in Time Compiler 来完成的。尽管如此，微软仍然把默认的项目编译输出文件夹称作 bin 文件夹。

在解决方案资源管理器中单击“查看类关系图” 图标按钮后，显示 Visual Studio .NET

当前项目中的类和类型的层次信息。在“类关系视图”中，可以对类的层次结构浏览、组织和编辑。如果双击“类关系视图”中的某一个类名称，将打开该类定义的代码视图，并定位在该类定义的开始处，如果双击类中的某一成员，将打开该类定义的代码视图，并定位在该成员声明处。

(2) 工具箱窗口。“工具箱”用于向 Windows 应用程序或 Web 应用程序添加控件。“工具箱”使用选项卡分类管理其中的控件。根据当前正在使用的设计器或编辑器，“工具箱”中可用的选项卡和控件会有所变化。在 Visual Studio.NET 开发环境左边的工具箱窗口中列出了可以应用到窗体上的控件，如图 1-6 所示。

(3) 窗体窗口。在 IDE 的中部是开发环境的主窗口，用来显示指定的窗体。窗体是一小块屏幕区域，通常为矩形，可用来向用户显示信息并接受用户的输入。设计窗体的用户界面的最简单方法是将控件放在其表面上，如图 1-5 所示。默认情况下，窗体是在设计视图状态下。在解决方案资源管理器中单击“查看代码”图标按钮后，会切换到程序代码编辑窗口；单击“查看设计器”图标按钮后，会切换到窗体设计视图窗口。

(4) 属性 (Properties) 窗口。如图 1-7 所示，在默认情况下位于 Visual Studio.NET 的右下角。与 IDE 的许多其他窗口一样，如果关闭属性窗口，还可以按下 F4 键重新打开该窗口。属性窗口同时采用了两种方式管理属性和方法，即按分类方式和按字母顺序方式。读者可以根据自己的习惯采取不同的方式。面板的下方还有简单的帮助，方便开发人员对控件的属性和方法进行操作和修改。图 1-7 是按字母顺序方式列出的窗体的属性。

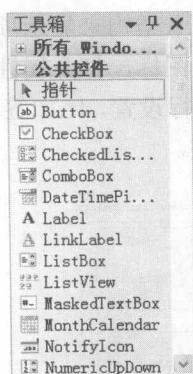


图 1-6 工具箱窗口

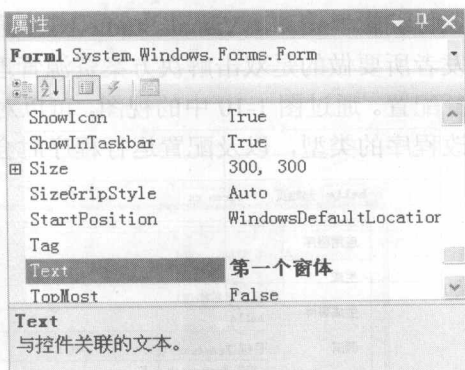


图 1-7 属性窗口

窗体和控件都有自己的属性，属性窗口列出了窗体或控件的属性，可以通过属性窗口对控件的属性值进行修改。

例如，在窗体设计视图中选择窗体 Form1，此时属性窗口就会显示 Form1 的属性，如图 1-7 所示。找到 Text 属性，把默认的 Form1 改为“第一个窗体”。一旦接受属性的改变，新值就会显示为窗体的标题。与在其他环境中通过用户界面编辑的属性隐藏在项目的一些二进制或专用部分不同，.NET 属性是在源文件中定义的。因此，尽管属性窗口看起来类似于其他环境，例如 VB6，但在 Visual Studio.NET 中它要强大得多。

(5) 输出窗口。用于提示项目的生成情况，显示程序运行时产生的信息。这些信息包括编程环境给出的信息，如在编译项目时产生的错误以及在程序中设定要输出的信息等。其外

观如图 1-8 所示。

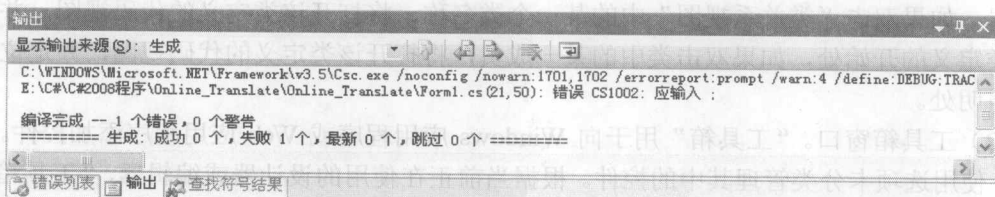


图 1-8 输出窗口

(6) 错误列表窗口。为代码中的错误提供了即时的提示和可能的解决方法。如图 1-9 所示，当某句代码中忘记输入分号作为本句的结束时，错误列表中会显示该错误。

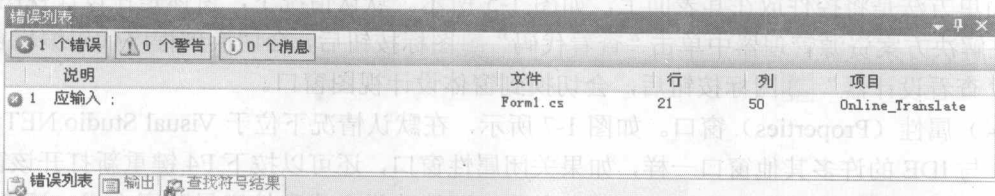


图 1-9 错误列表窗口

3. Visual Studio2008.NET 中的其他窗口

(1) 属性编辑器窗口。Visual Studio.NET 通过一个垂直的选项卡结构来显示项目的设置信息。开发者所要做的是双击解决方案资源管理器中的 Properties 图标，打开属性编辑器来对项目进行配置。通过图 1-10 中的视图，可以对项目的各个方面进行设置，比如签名、安全、发布、修改程序的类型，以及配置运行程序时会用到的外部支持等。

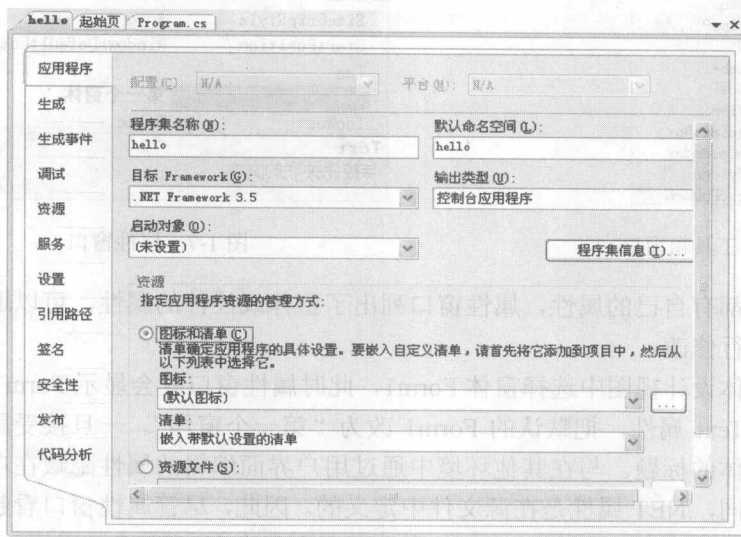


图 1-10 属性编辑器窗口

在图 1-10 中通过单击图中的“应用程序”标签可以更改程序集信息以及为应用程序中的

类设定根命名空间。后面将介绍命名空间的有关概念，这里只需要知道命名空间是可以嵌套的。这样做的好处是可以按照逻辑结构来整理类，方便开发人员减少的同时可能造成的混乱。与 COM 组件的开发类似，创建自己的根命名空间，并把随后开发的类都归入其中是一个值得提倡的做法。当然，这并不是说项目只用到自己的命名空间，实际上，系统命名空间是一切应用程序的基础。

另外在图 1-10 中单击“程序集信息”按钮可以弹出图 1-11 所示的“程序集信息”对话框，并在该对话框中设置文件属性，比如公司名和版本号等。这些信息在构建项目时会被自动写入生成的文件中。过去开发人员必须在 AssemblyInfo.cs 中手工编写 XML 格式的配置信息，现在可以直接在 Visual Studio.NET 中的对话框中进行相关设置。

程序集属性对应的 AssemblyInfo.cs 文件中包含很多设置程序集信息的属性块。每个属性块都含有一个 assembly 标识符。例如：

```
<Assembly: AssemblyTitle("中国象棋")>
<Assembly: AssemblyDescription("")>
<Assembly: AssemblyCompany("中原工学院")>
<Assembly: AssemblyProduct("")>
<Assembly: AssemblyCopyright("夏敏捷")>
<Assembly: AssemblyTrademark("")>
<Assembly: CLSCompliant(True)>
```

通过上面图 1-11 的对话框，开发人员可以直接在开发环境中配置程序集信息而不需要手工编写 XML 文件，极大地提高了效率。

(2) 类视图窗口。“类视图”以树型结构显示 Visual Studio.NET 当前项目中的类和类型的层次信息。在“类视图”中，可以对类的层次结构浏览、组织和编辑。如果双击“类视图”中的某一个类名称，将打开该类定义的代码视图，并定位在该类定义的开始处，如果双击类中的某一成员，将打开该类定义的代码视图，并定位在该成员声明处。

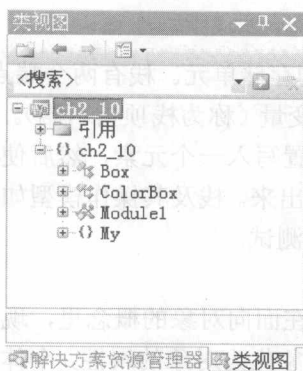


图 1-12 类视图窗口

的命名空间被自动引入，见表 1-1。

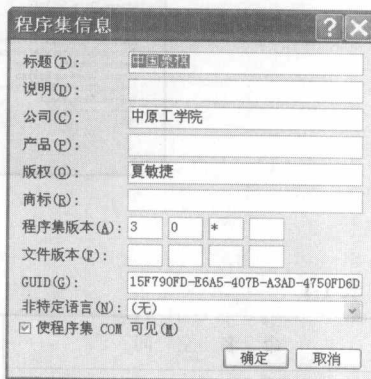


图 1-11 “程序集信息”对话框

在系统默认情况下，类视图窗口是不显示的。可以选择“视图”→“其他窗口”→“类视图”命令进入类视图窗口，如图 1-12 所示。

(3) 引用窗口。在项目中可以添加引用。选择菜单“项目”→“添加引用”命令，弹出图 1-13 所示的“添加引用”对话框。可以选择要引用的 .NET 类库、应用程序，以及 COM 组件。甚至可以引用当前解决方案中定义在其他项目中的类。一个新项目有系统默认的文件，与此类似，每个项目也会有默认的库引用。对于 Windows 窗体应用程序来说，下面的