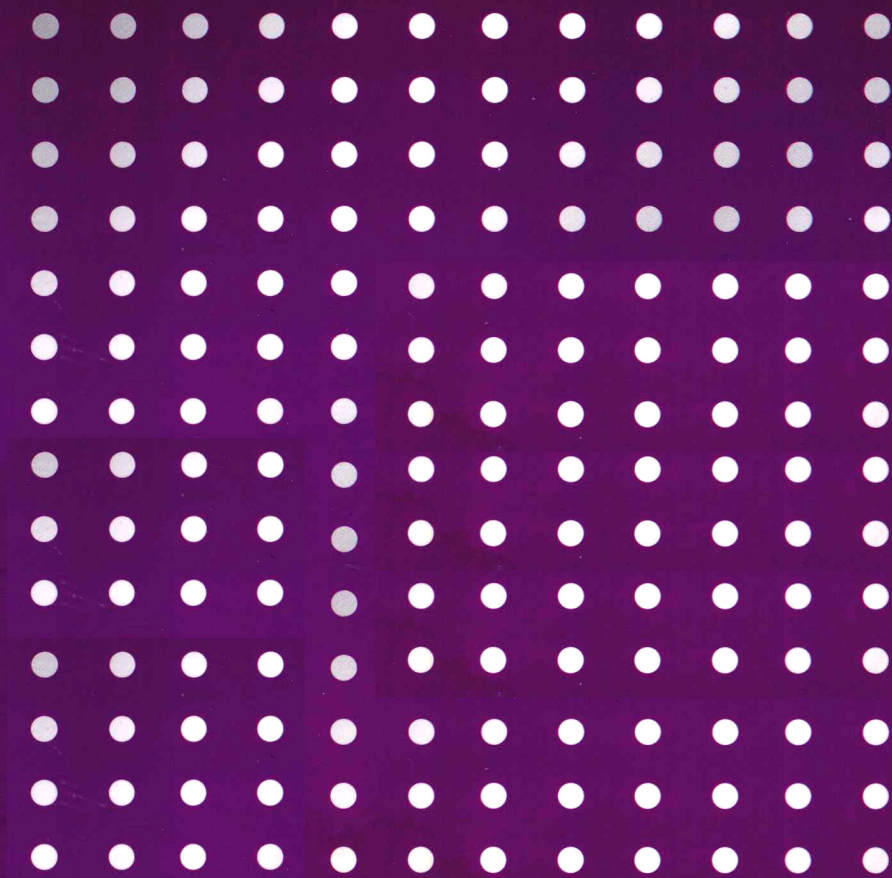


高等院校信息技术规划教材

C#.NET企业级 项目开发教程

马瑞新 主编

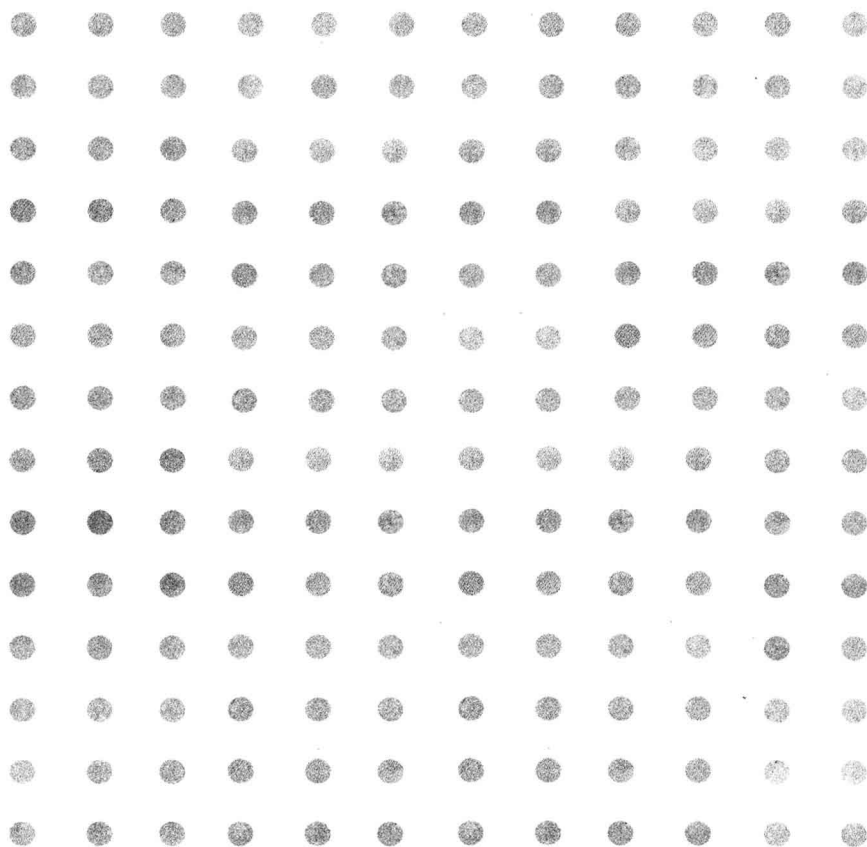


清华大学出版社

高等院校信息技术规划教材

C#.NET企业级 项目开发教程

马瑞新 主编



清华大学出版社
北京

内 容 简 介

本书围绕程序设计语言的基础知识和 C# 语言的强大功能,采用项目导向与案例的方式,通过丰富的实例全面介绍了基于 C# 语言和 .NET 平台进行企业级应用开发的知识、方法和技術。全书共 9 章,将 .NET 框架、C# 语言、ADO.NET 和 ASP.NET 的知识与综合训练实例紧密地融合在一起,突出面向对象技术的教学主线,体现了以用为本、学以致用原则,使学习者逐步形成规范、良好的专业开发人员的基本素养以及分析和解决问题的能力。

本书所有项目均是在 Visual Studio 2010 上开发的,无论是编程新手还是具有 C++/Java 基础的程序员,都可以从本书中获取新知识、新方法和新技术。

本书可作为高校应用型本科相关专业課教材,也适合作为企业培训教材,并可供软件开发人员自学参考。

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。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

C#.NET 企业级项目开发教程/马瑞新主编. --北京:清华大学出版社,2012.4

(高等院校信息技术规划教材)

ISBN 978-7-302-27939-6

I. ①C… II. ①马… III. ①C 语言—程序设计—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2012)第 010405 号

责任编辑:白立军 战晓雷

封面设计:傅瑞学

责任校对:时翠兰

责任印制:李红英

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课件下载: <http://www.tup.com.cn>, 010-62795954

印刷者:北京富博印刷有限公司

装订者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:32.25

字 数:741 千字

版 次:2012 年 4 月第 1 版

印 次:2012 年 4 月第 1 次印刷

印 数:1~3000

定 价:49.00 元

产品编号:043647-01

前言

foreword

现代的程序设计语言如雨后春笋一样,不停地在我们面前出现,到底哪种语言更好?什么技术最强?这种问题也频繁出现在各大论坛。其实,“不管白猫还是黑猫,能抓住老鼠的就是好猫。”每种技术都有它的优势,同时也有它的不足,任何一个人都无法掌握全部技术。在企业开发中,我们也不可能只用一种技术打遍天下。“授之以鱼不如授之以渔”,我们要掌握的是解决问题的方法,这就足矣。

笔者曾经在软件公司工作过10年,那么软件公司需要什么样的人才呢?

(1) 外包开发行业快速发展,对“人才”在代码和文档方面的规范性、技能和工具的熟练程度要求越来越高。

(2) Java和.NET技术在市场上平分秋色,都有大量的岗位需求,同时值得庆幸的是两者在应用层面上的技术差异越来越小。

(3) 软件开发企业对开发人员的基本技术素养强调得越来越多,例如面向对象的程序设计思想和代码组织方法、HTML/CSS/JavaScript客户端技术等。

(4) 为了保证质量和工期,企业中大量使用各种框架技术,要求开发人员至少熟悉一种框架技术。

(5) MIS、OA、ERP、CRM、系统集成、物流、进销存、电子政务和网站建设等加大了对软件工程师的需求。

但是企业强调得最多的是关于软件开发人员的职业素质要求:作为一名职业人,守时、耐心、踏实的习惯以及适应能力、学习能力、抗压能力、规范意识、合作意识和沟通意识,都是不可缺少的要素。多年的跟踪数据显示,具备这“3个习惯、3个能力、3个意识”的软件工程师大都在职业上获得了良好的发展,否则技术再强也难有好的发展机会。

我们知道,软件开发领域内所涉及的技术内容是非常多的,不同的行业会使用不同的技术,不同的项目也会使用不同的技术,在技术选择上我们遵循了实用原则和80/20原则。

(1) 实用原则：所选择的技术一定是能够解决实际工作中的实际问题的技术。打个比方，假设我们在学习修车，关于汽车原理、发动机原理和轮胎的橡胶特征我们就不会花精力去学习，因为这些内容在修车过程中很少会使用到，而如何使用各种检测设备和螺丝刀等工具以及不同品牌汽车修理和保养的经验，则是汽车修理人员最迫切需要掌握的技术。因此我们学习的主要内容也集中在如何解决软件开发项目中所涉及的技术工具、技术框架、开发流程和编码调试经验等方面的问题。请记住：“不要去背诵对象有哪些方法和属性，而是要使用这个对象去解决问题”。

(2) 80/20 原则：该原则包含两层意思，第一重点讲解企业 80% 的时间在使用的 20% 的核心技术，而那些 80% 的不常用的非核心技术则不讲，或者只作为介绍或拓展内容；第二花费 80% 的精力才能够学会的 20% 的内容不讲。

总之，以用为本，学以致用，不用不学，学了就用，综合运用才是王者之道。把发动机的各项参数背得再熟也是不会修车的，而快速判断问题发生的原因，使用扳手、螺丝刀快速搞定问题才是正道。修车高手只能通过大量的实践、总结、再实践才能达到快刀斩乱麻地解决问题的境界。同样，编程的学习也要求你“动手、动手、再动手”，要敢于编码、乐于编码、大量编码才能够达到熟练的程度。

现代企业要求合作，代码规范是团队开发的基础，没有人看得懂的代码，不管自己觉得写得如何漂亮，终将被称为“垃圾代码”。在往后的学习中要记住：“代码运行通过了不算代码写完了，代码规范了才算。”

有这样一个故事：说一个人能够射箭时百步穿杨，别人恭维他时他自己也趾高气扬。但是有一天他看到一个卖油翁单手拿着一瓢油往一个很细的葫芦嘴里倒油，结果一滴都没洒，他顿时觉得这才是真正的高手，便向前请教，卖油翁说了句很经典的话，“无他，唯手熟尔。”同样，在软件编程中只有通过大量的、反复的动手实践，才能具备熟练、规范的代码和调试能力，具备真正实用的技能，才有可能被称为应聘者中“有项目开发经验的人”，成为一名企业真正需要的“人才”，在日益激烈的职业竞争中拔得头筹。

我的一位朋友每天晚上都要看电视，他一只手（有时是两只手）拿着遥控器，过几分钟就换一个台（他总是相信另一个频道有更精彩的节目），一晚上总要将 40 多个频道轮换好几遍，结果每个电视剧都没有看完整。有人曾对我说，除去愚蠢之外，浮躁将是阻碍我们进步的最大障碍。当然学习不同于看电视，但是，如果你总是能发现比自己正在学习的还要好的语言，那么我相信，你已经开始拿起遥控器。很早以前我就知道，如果选择的路线正确的话，那么只要坚持下来，就一定能到达目的地，如果你将 C# 作为你现在的选择，那么请你一定要坚持下去。C# 是一个概念清晰、内涵丰富的语言，虽然融合了多种语言的特点，但是在学习 C# 之前，那些语言的知识都不是必需的。不过学习此种语言一定要集中精力将语言的各个基本特性学好，不但要理解其含义，知道其用法，还应多想语言中为何要包括这个特性，并弄清楚它们与其他特性的关系。无论学习何种语言，开始的时候都应该只专注于语法本身，然后再来深入理解面向对象技术，最后，在各种应用中学习 .NET 的类库并且逐渐去领会语言的本质。当在实际工作中使用得多了，自然就会慢慢理解一些更高级的知识，再进一步发展，就开始研究 .NET 源码，自己开始动手为 .NET 平台添砖加瓦。这个时候，在江湖上你就已经是一位大师级的人物了。

学习是一个慢慢体会的过程,同时也是一个逐渐发现的过程,一个人能够学习、积累、总结和思考,这将是幸福的。

本书由马瑞新主编,参与本书编写的还有韩鑫、姜国海、刘宇。同时也感谢大连理工大学软件学院的全体同学,是他们热情的支持和鼓励让我有信心完成此书的编写,每当我快要放弃时,总是他们给我力量,我特将此书作为送给他们的礼物,希望我们一起在软件开发上更上一层楼。

由于作者水平有限,加之时间仓促,书中的错误之处在所难免,敬请专家、读者不吝赐教,予以指正。

马瑞新

2011年10月

目录

Contents

第 1 章 .NET 框架	1
1.1 .NET 概念	1
1.2 .NET 平台	4
第 2 章 C# 语言	7
2.1 数据类型	7
2.2 类的组成	11
2.2.1 创建对象	12
2.2.2 属性	13
2.2.3 类继承	14
2.2.4 多态	14
2.2.5 命名空间	17
2.2.6 重载	18
2.2.7 接口	21
2.2.8 委托	24
2.2.9 多路广播委托	25
2.3 异常处理	26
2.4 集合	29
2.4.1 数组	29
2.4.2 集合对象	33
2.5 泛型	38
2.6 文件、流、序列化	45
2.6.1 基本文件的 I/O	45
2.6.2 驱动器、文件夹和文件	50
2.6.3 文件的创建、删除、移动和查找	53
2.6.4 向文件读写文本和数据	55
2.6.5 序列化	57



第 3 章 综合训练 1	61
3.1 ATM 机存取款	61
3.2 电脑信息管理	74
3.3 汽车租赁系统	87
第 4 章 ADO.NET	99
4.1 ADO.NET 简介	99
4.2 ADO.NET 两种访问数据库模式	102
4.3 ADO.NET 连接模式	103
4.4 ADO.NET 非连接模式	118
4.5 数据绑定	119
4.6 DataSet	126
第 5 章 综合训练 2	132
5.1 员工信息管理系统	132
5.2 酒店客房管理系统 1	158
5.3 火车里程时刻查询系统	202
第 6 章 ASP.NET	210
6.1 ASP.NET 简介	210
6.2 ASP.NET 内置对象	217
6.3 Web 服务器控件	227
6.4 数据绑定	244
6.5 验证控件	247
6.6 母版页和站点导航	251
第 7 章 综合训练 3	255
7.1 网页计数器	255
7.2 北极星博客	268
7.3 酒店客房管理系统 2	311
第 8 章 深入 ASP.NET	353
8.1 成员资格和角色管理	353
8.2 个性化用户配置	369

8.3 XML Web Service	390
8.4 Ajax	396
第 9 章 综合训练 4	406
9.1 MyComics 漫画书网站	406
9.2 网上书店	445

.NET 框架

1.1 .NET 概念

对于 .NET,不同的人可能会对它有不同理解..NET 究竟是什么?人们的回答总是随着时间的推进而改变。最早认为这是一个新的平台,后来认为是 Microsoft 的新战略,但是现在我们这样认为:.NET 是一个概念,是一种构想,或者是微软的一个梦想。只有站在不同的角度,或许才能更深地理解 .NET 的要领。

对于 .NET,微软有如下正式的描述:“.NET 是 Microsoft 用以创建 XML Web 服务(下一代软件)的平台,该平台将信息、设备和人以一种统一的、个性化的方式联系起来。借助于 .NET 平台,可以创建和使用基于 XML 的应用程序、进程和 Web 站点以及服务。它们之间可以按照设计,在任何平台或智能设备上共享和组合信息与功能,以向单位和个人提供定制好的解决方案..NET 是一个全面的产品家族,它建立在行业标准和 Internet 标准之上,提供开发(工具)、管理(服务器)、使用(构造块服务和智能客户端)以及 XML Web 的服务体验(丰富的用户体验)..NET 将成为今天正在使用的 Microsoft 应用程序、工具和服务器的一部分,同时,新产品不断扩展 XML Web 的服务能力,以满足所有业务需求。”这个于美国时间 2000 年 6 月 22 日上午面世的新概念一经发布,就引起全球 IT 界的关注,因为发布它的微软公司一向以改变业界规则而闻名,该公司的另一个出名的地方就是精通“拿来主义”,善于继承别人的思想再经过封装成为自己的产品。在 .NET 的新产品中,有 Delphi 的影子,C# 更宛若 Java 的一个克隆,甚至 .NET 的“网格计算”的概念也曾经在很早以前由 Sun 提出过。但是这些并不能妨碍 .NET 成为最热门的技术词汇,随着 2002 年 .NET 1.0 的发布,到 2003 年的 .NET 1.1,2005 年的 .NET 2.0,2006 年的 .NET 3.0,2007 年的 .NET 3.5 直到 2010 年的 .NET 4.0,微软一步一个台阶,使 .NET 成为人们谈论最多的话题之一。

要真正了解 .NET,就一定要从微软的历史开始。意大利文艺美学家科洛齐有一句名言:“一切历史都是当代史。”也许回忆一下历史会让我们更容易理解微软的 .NET 计划。1975 年,罗伯茨因其发明了第一台微机“牛郎星”大获成功。当时由于他经营的 MITS 公司陷入困境,情急之下才发明了这台微机。《大众电子》曾在宣传中说,“牛郎星”将面向普通用户,采用最简便的 BASIC 语言编程。然而,它的内存只有 4KB,专家们都认为它难以支持 BASIC,罗伯茨手中根本没有这种软件。某一日,罗伯茨突然接到一个



电话,对方声称自己是西雅图市的交通资料公司,可以为“牛郎星”开发出 BASIC 语言,并询问他是否感兴趣。罗伯茨岂止是有兴趣,如果有谁真能做成这件事,那“牛郎星”就不再是件玩具,它将成为一台名副其实的个人计算机。一个星期后,罗伯茨又接到一封正式的信函,信中说他们已经做好了软件。罗伯茨当即按信中提供的联系方式打去电话,可对方答复说那是西雅图的湖边中学。罗伯茨摇摇头,心想这一定是哪个调皮的学生的恶作剧。正准备度假时,这个子虚乌有的公司又打来长途。这一回,罗伯茨不客气了,他恼怒地大声嚷道:“如果你们真有 BASIC 软件,那就劳驾送到 MITS,我保证到机场恭候!”他本想激出对方的真情,哪知电话传来肯定的答复:“完全可以,三周后,请您到机场接机!”三周后,罗伯茨在机场接到的竟是一位乳臭未干的毛孩子。来者说他叫保罗·艾伦,全权代表他的伙伴比尔·盖茨。而此时,比尔·盖茨正在远隔千里的波士顿哈佛大学校园里忐忑不安地等待消息。当这两个不知天高地厚的少年通过电话向罗伯茨拍胸脯保证三周交货的时候,所谓的 BASIC 软件还只是凭空想象而已。好在无论是模拟还是解释程序,对两人来说都不陌生。艾伦的工作必须先行,他不到两周就写完了模拟程序。比尔·盖茨面对的难题更多,他整整写了 8000 行的机器语言程序,又千方百计地精简压缩,设法使它能装进“牛郎星”狭小的“身躯”。然而,越是临近完工,他们越是紧张,毕竟没有一台真正的“牛郎星”,编写的程序能不能在计算机上运行,谁也不敢保证。直到把艾伦送上了飞机,比尔·盖茨的心仍不踏实。显然跟随罗伯茨来到 MITS 的开发试验室的艾伦更加紧张。当他终于第一次看到了真实的“牛郎星”时,立即把带来的程序纸带送进计算机的读带装置,此时,他的心已经提到了喉咙口。时间一分一秒地过去,读带机还在“咔咔”地“吃进”纸带,这几分钟就像是过了好几年。突然,电传打字机轻轻地动了一下,打印纸上印出了 READY,标志着计算机做好了准备。艾伦马上写出了一段模拟阿波罗飞船登月的程序,交给罗伯茨打孔输入,程序果然十分准确地运行并计算出结果。这大概是世界上第一台微机的真正应用。1975 年 2 月,大功告成。艾伦亲赴 MITS 演示,大获成功。这年春天,艾伦进入 MITS,担任软件部经理。念完二年级课程,盖茨也飞往 MITS,加入艾伦所从事的工作。那时他们已有创业的念头,但要等到 BASIC 被广大用户接受才可,此前他们是不会离开罗伯茨的,他们还羽翼未丰。罗伯茨开始摇旗呐喊,如今在他手里有了一件最能吸引大众的法宝——买一部“牛郎星”计算机,可配上一位“织女”——BASIC 软件,这引来各界人士的纷纷解囊,到 1980 年为止,这种软件竟卖出了 100 万套。在 BASIC 软件成功的鼓舞下,比尔·盖茨毅然从哈佛大学退学,于 1975 年 7 月在阿尔伯克基竖起了微软公司的大旗,公司简称 MS,连同艾伦,一共六员大将,专门从事微型计算机的软件开发。比尔·盖茨为公司制定了目标:“每一个家庭每一张桌子上都有一台微型计算机运行着我们的程序!”这个时候,微软还只能算是一个编程语言公司。1978 年,微软公司搬到了太平洋西北岸,并继续为另外一些微型计算机编写程序。从此微软迅速发展起来,1978 年,微软的收益仅为 100 万美元,到 1984 年,微软收益已经突破 2.3 亿美元。但是这时候微软对行业的影响力还远没有现在这么大,直到有一天,IBM 公司找上门来,要求与比尔·盖茨联合开发新的操作系统,于是就产生后来对 IT 业影响深远的 MS DOS。1985 年 6 月,微软和 IBM 达成协议,联合开发 OS/2 操作系统。根据协议 IBM 在自己的计算机上可随意安装 OS/2,几乎分文不取,而微软向

其他计算机厂商收取 OS/2 的使用费用。当时 IBM 在 PC 市场拥有绝对的优势,兼容机份额极低,当时 IBM 负责此项目的洛绎几乎是不假思索地同意了。到了 1989 年,兼容机市场已达到了 80% 的份额,使得微软在操作系统的许可费上短短几年就盈利 20 亿美元。当然双方在 OS/2 的合作没能持续下去。微软从 1981 年就开始开发后来称之为 Windows 的操作系统。Windows 最初的版本与 OS/2 有很大的关系。1985 年,Windows 1.0 的问世,才能使其成为一个产品,而 1995 年,Windows 95 的发布则正式把微软推向计算机业的巅峰,至此微软帝国的雏形出现。随后微软四处出击,向各个领域伸出触角,在文字处理、电子表格、IDE、网络产品等各方面逐渐建立了自己的领先地位,完成了一个小小的程序语言开发公司到操作系统公司再到应用产品集成公司的转变。但是,随着互联网的出现,一切都发生了变化。第一个挑战来自 Netscape,1994 年 10 月 13 日,Marc Andressen 发布了名为 Netscape Navigator 的新一代浏览器,并发布了 HTML 标准。1995 年 Netscape 2.0 发布时,Netscape 已经拥有几百名员工,超过 1000 万的用户和快速增长的产品线,包括代理服务器 Proxy Server、网络开发工具还有商业应用程序 I-Apps,并支持在网络上销售产品或提供服务。1995 年,Netscape 的年收入达到了惊人的 8000 万美元,成为历史上增长最快的软件公司。这给微软很大的刺激,于是当年微软发布了人们期待已久的新操作系统 Windows 95,同时还有 WWW 浏览器 Internet Explorer 1.0。但是此时再行动显然已经慢了一步,很快,微软在 IM 软件、搜索引擎、流媒体和服务器等各方面都受到了不同程度的挑战,尤其是在与 Netscape 的竞争中,不但耗资巨大,而且还受到了使用不正当竞争手段的控告。很明显,微软已经感觉到了未来互联网的力量,比尔·盖茨在 IE 的发布会上将 Internet 的崛起称为“IBM 生产个人计算机 10 年以来最重要的发展”。虽然 Windows 95 取得了极大的成功,但这并不能说明微软已经就此拥有未来,比尔·盖茨显然已经意识到,随着互联网的发展,他们用以征服天下的武器——操作系统的作用将会越来越小,也许人们有一天不再需要安装操作系统来管理计算机和运行各种软件,只需要使用浏览器就可以完成现有的工作了,到那个时候,软件业的“日不落帝国”的称号就有可能落到 Netscape 的头上。于是微软凭借操作系统的优势正式对 Netscape 宣战,虽然付出的代价不小,不过最后总算取得了胜利。而这时 Sun 推出了 Java,提出了一种纯面向对象的跨平台的软件开发模式,并且在业界公开了使用授权,使之很快标准化,各软件大腕纷纷加盟,其中也包括微软。但当时 Java 的技术特性(跨平台)决定了其固有的弱点:运行效率不高,对操作系统平台所具有的特点支持不到位。于是微软针对 Windows 平台对 Java 进行了一些优化,当然这是以损害 Java 的跨平台特性为代价的。此举触怒了 Sun 和 Java 阵营,于是 Sun 通过一场诉讼将微软赶出了 Java 阵营。但此时微软已经为 Java 投入甚多,也意识到 Java 中间的一些技术代表了未来软件的方向。恰巧此时微软已经将 Borland 公司 Delphi 的设计者 Hejlsberg 挖到手了,于是微软干脆抛开 Java,放开手脚沿着 Java 的技术方向自己搞一套。比尔·盖茨放言:“Information on your finger tip,anywhere, anytime, anyway you want”,他认为互联网时代来临了,从而给这个产品起了个气势磅礴的名字.NET,其中.NET 在英文中表示 Start,NET 表示网络,微软这一伟大的计划正是要靠.NET 才能实现。

从技术上看,完全可以认为.NET 就是微软的 Java,在很多技术特征上(如中间语

言、虚拟机、面向对象)两者如出一辙。当然它们的理念则有所不同,Java 要跨所有的操作系统,而 .NET 主要面向 Windows 操作系统。

每一个大型战略的实施都给众多的商家带来新的商机。PC 的出现产生了目前居第一位的信息产业,而 .NET 的发展和形成,也将给许多开发商和合作伙伴创造新的契机。微软将为领先的合作伙伴提供支持。可以肯定,众多的 .NET 的支持者会从中受益,这些人包括将使其应用扩展到新设备上的现有合作伙伴,逐步扩大网络服务以增加用户服务项目、开发新收入来源的客户,对促进自动化、互动式的网络服务有创意,希望以此促进互联网使用体验的新型合作伙伴。1992 年,微软公司进入中国,那时全中国懂 Windows 的人不足百人,而今天,数以千万计的人们在使用 Windows,最早应用 Windows 技术的中国厂商是新天地公司。目前,有数十万计的基于 Windows 的软件公司、系统集成商和硬件产品供应商。他们都成为目前中国最有影响力的软件厂商,而经微软授权的中文 Windows 软件开发商已达到 8000 多家,开发出众多基于 Windows 的大量应用软件。而我们相信 .NET 技术的推广,不仅将使上述厂商成为第三代互联网平台上的领先厂商,将使更多的信息终端供应商和互联网服务商成为受益者,而且更重要的是,用户将得到更完善的服务。今天,包括 IBM、惠普、三星等一大批微软的合作伙伴,商业界的巨子们及众多的传统产业人士、互联网服务供应商已开始与微软密切合作,共同推广 .NET 技术,搭建第三代互联网平台。在微软公司内部,所有的部门设置都以 .NET 战略的实施开始重组,微软公司所有的资源将为这一伟大的工程服务。比尔·盖茨说过,.NET 是一项以公司的命运为赌注的赌博,微软公司的全部战略也是围绕着 .NET 平台而制定的。许多的战略伙伴愿意与我们同舟共济,所以我们相信这项工程能够取得成功。10 年后,微软有可能不再出售任何一种盒装软件,伴随着 .NET 的发展,微软会成为一家以提供软件技术为基础的服务型公司,我相信这一天终会到来,我相信 .NET 也会为中国的用户带来更大的益处和方便。

1.2 .NET 平台

首先要清楚的是,.NET 平台与 C# 语言不是一回事,对开发人员来讲,.NET 有两个概念,首先,它是 C#、VB、.NET 等程序运行的平台;其次,它为这些语言提供了丰富的类库(称之为基类库)。对一个面向 .NET 平台进行开发的人来说,了解 .NET 平台的整个构成是非常有必要的。图 1-1 说明了 .NET 平台的整体框架结构。

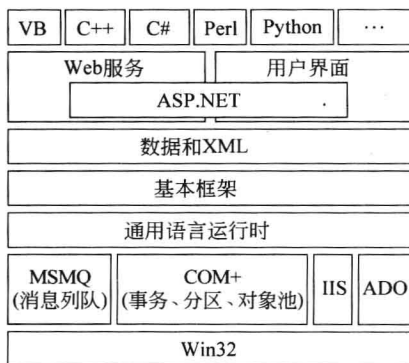


图 1-1 .NET 框架

从图 1-1 中可看到,.NET Framework 是在 Microsoft .NET 平台上进行开发的基础,ASP.NET、Windows Forms 和 ADO.NET 都只不过是 .NET 开发平台的一部分,它们用于 .NET 应用程序的开发及展示。.NET 平台的核心技术为通用语言运行时(Common Language Runtime,CLR)、

基类库(Base Class Library,BCL)和 .NET 语言及开发工具 Visual Studio 2010(以后简称 VS 2010)。从图 1-1 中也可以看出,.NET Framework 是架构在 Windows 平台上的一个虚拟的运行环境,你可以想象将最下层的 Windows 换作其他操作系统,例如 Linux,一样可以实现使用符合 CLR 的 .NET 语言来创建应用程序,这其实就是微软 Mono 计划所要实现的功能。所以可以这么认为,理论上 C# 是一种可以跨平台的语言,这很像 Java。C# 另一个比较像 Java 的地方是,它是一种解释性的语言,C# 所编写的程序也是通过 C# 编译器编译成为一种特殊的字节代码(MSIL,微软中间语言),运行时再由特定的编译器(JIT 编译器)编译成为机器代码,以供操作系统执行。不仅是 C# 语言,所有 .NET 语言(包括常用的几十种现代的编程语言)都可以编写面向 CLR 的程序代码,这种代码在 .NET 中称为托管代码(Managed Code),所有的托管代码都直接运行在 CLR 上,具有平台无关性。解释性的语言很安全,并且可以通过它的运行平台为其赋予更多的功能,如自动内存管理和异常处理等。事实上,C# 语言的许多特点都是由 CLR 提供的,如图 1-2 所示。

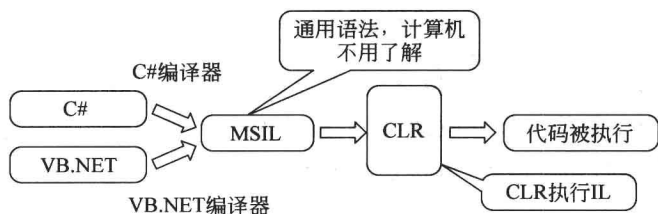


图 1-2 CLR

CLR 位于操作系统之上,并且为承载的应用程序提供虚拟环境。当托管可执行程序时,CLR 加载包括可执行程序的模块,并在其内部执行代码。在多数情况下,给定的方法只在第一次调用时进行 JIT 编译,然后存储在内存中,以便能够无延迟地再次执行。没有被调用的代码不会被 JIT 编译。虽然 JIT 编译影响了性能,但负面影响还是有所减轻,因为在应用程序生存期内方法只被编译一次,而且 Microsoft 的 CLR 小组已经使 JIT 编译器尽可能地快速高效。理论上 JIT 编译的代码胜过普通代码,因为 JIT 编译器可以优化它所运行的主处理器产生的本地代码。如图 1-3 所示的 CLR 结构图说明了这一点。

JIT 编译的代码与解释代码是不同的。在 CLR 的托管环境中,运行代码的好处很多。简单说来,在 JIT 编译器将 CLR 指令转换成本机代码的过程中,CLR 将执行代码验证过程,以确保代码的类型是安全的。实际上,指令是不可能访问没有获得许可的内存的。在托管应用程序中,不会遇到任何与非法指针有关的问题,因为 CLR 在用到非法指针之前就会引发一个异常。不能将一种类型转换为与其不兼容的类型,因为这种操作是不安全的。同时,无法调用格式错误的堆栈结构的方法,CLR 根本不允许其运行。代码安全验证不但可以消除一些影响应用程序的最常见的缺陷,而且使得刻意编写伤害主机系统的恶意代码变得非常困难。如果不需要进行代码安全验证,系统管理员可以关闭此功能,但这种情况非常危险。运行于托管环境中的另一个好处就是托管代码分配的资源

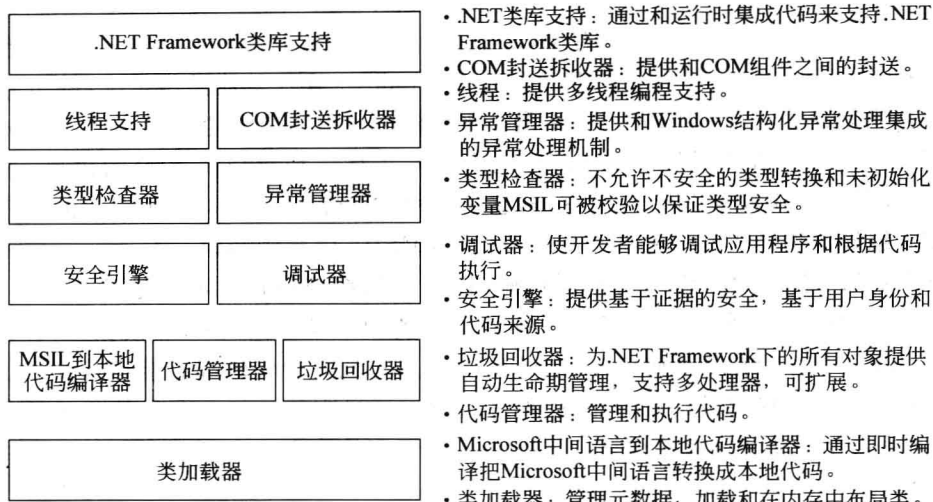


图 1-3 通用语言运行时结构图

可以被系统作为垃圾回收。换句话说，分配内存后，不需要释放它，系统会替用户将它释放。CLR 包含一个完整的垃圾回收器，用来跟踪对代码所创建的对象引用，当其他对象不再引用这些对象时，就可以自动将它们销毁。

C# 语言

2.1 数据类型

C# 是微软公司设计的一种编程语言。它松散地基于 C/C++，并且有很多方面和 Java 类似。微软是这样描述 C# 的：“C#（读做 C sharp）是从 C 和 C++ 派生来的一种简单、现代、面向对象和类型安全的编程语言。C# 主要是从 C/C++ 编程语言家族移植过来的，C 和 C++ 的程序员会马上熟悉它。C# 试图兼备 Visual Basic 的快速开发能力和 C++ 的强大灵活的能力。”

C# 是一种相当新的编程语言，它是随 .NET 一起设计出来的全新开发语言，其设计目的就是作为 .NET 的核心语言，它是为了那些愿意牺牲 C++ 的某些原有能力换得更多便利和效率的企业应用开发者而创造的。C# 的重要性体现在以下两个方面。

(1) 它是专门为与微软的 .NET Framework 一起使用而设计的(.NET Framework 是一个功能非常丰富的平台，可开发、部署和执行分布式应用程序)。

(2) 它是一种基于现代面向对象设计方法的语言，在设计它时，微软还吸取了其他类似语言的经验，这些语言是近 20 年来面向对象规则得到广泛应用后才开发出来的。

C# 程序类似英文文章(但语法严格得多)，由相当于英语的变量等标识符、加减乘除等运算符及空格等分隔符组成语句，若干条语句再组成相当于文章段落的方法，而方法和一些语句又组成类，一个类就相当于一篇文章或一本书的一个章节。由多个类型再构成一个命名空间，相当于一本书或一部主题文集。C# 语言的标识符是按照一定规则定义、命名的符号名称。

C# 标识符命名规则如下：

- (1) 只能使用字母(含汉字)、数字和下划线。
- (2) 必须以字母或下划线开头。

例如，x、y、strX、_age、StudentCourseScore、CalculateArea、stu1、stu2、数 1、数 2、变量 2、计算面积等均是符合语法规则(简称“合法”)的标识符。虽然汉字可作标识符，但一般情况下不使用汉字命名的标识符。

无效的标识符如 2stu、x+y、n! 和 price\$ 等。另外请注意 C# 是大小写敏感的语言，一个字母的大写和小写视作不同的符号，因此，area 和 Area 是两个不同的标识符。

标识符的用途是：对程序的各个组成元素，如变量、方法和类等，进行命名标识。



标识符有两种定义和使用范围：

- (1) 系统预定义保留的、含义固定的标识符，即系统标识符，称为关键字。
- (2) 程序设计人员在一定范围自定义的标识符，即用户标识符。

其中，关键字的使用范围很广，但不能使用关键字做用户标识符。

C# 语言的关键字共有 77 个：abstract、as、base、bool、break、byte、case、catch、char、checked、class、const、continue、decimal、default、delegate、do、double、else、enum、event、explicit、extern、false、finally、fixed、float、for、foreach、goto、if、implicit、in、int、interface、internal、is、lock、long、namespace、new、null、object、operator、out、override、params、private、protected、public、readonly、ref、return、sbyte、sealed、short、sizeof、stackalloc、static、string、struct、switch、this、throw、true、try、typeof、uint、ulong、unchecked、unsafe、ushort、using、virtual、void、volatile、while。

关键字是有特定意义的系统预定义保留的标识符，不能在程序中用作自定义标识符，除非它们有一个@前缀。例如，@if 是一个合法的用户标识符，而 if 不是，因为它是关键字。

变量是存放数据的标识符(名称)。变量可看作是容纳数据的一个存储单元，即容器的名称。变量所存放的数据在不同时刻允许变更。变量的值允许变化，但所存放的数据类型一般是固定的，故声明变量(即定义变量)时要指定数据类型。

声明变量的格式：

数据类型 变量列表

例如：

```
double x;           表示声明变量为双精度实数类型
```

声明变量后，就可以对变量进行赋值。如：

```
x=23.5;
x=4.8;              //变量 x 的值从 23.5 改为 4.8
```

变量要先声明后使用，不能使用没经过声明定义的变量。

方法内部的局部变量要先赋值再读取，因为局部变量没有隐含值。

变量属于标识符，起名必须遵循标识符的命名规则。此外，对变量的命名还有如下建议：

- (1) 变量要以小写字母开头。
- (2) 含多个英文单词的标识符，除了第一个单词全部小写之外，其余单词首字母均要大写(称为 camelCase，即骆驼格式记号法)。
- (3) 为了方便程序人员阅读理解，不要创建仅大小写有区别的变量，如不要同时创建 myVariable 和 MyVariable、x 和 X 等，要做到见名知意。

注意：可一次声明多个变量，变量之间用逗号分隔，同时允许对部分或全部变量进行赋值，例如：

```
double x=2.1,y,z=3.4;
```