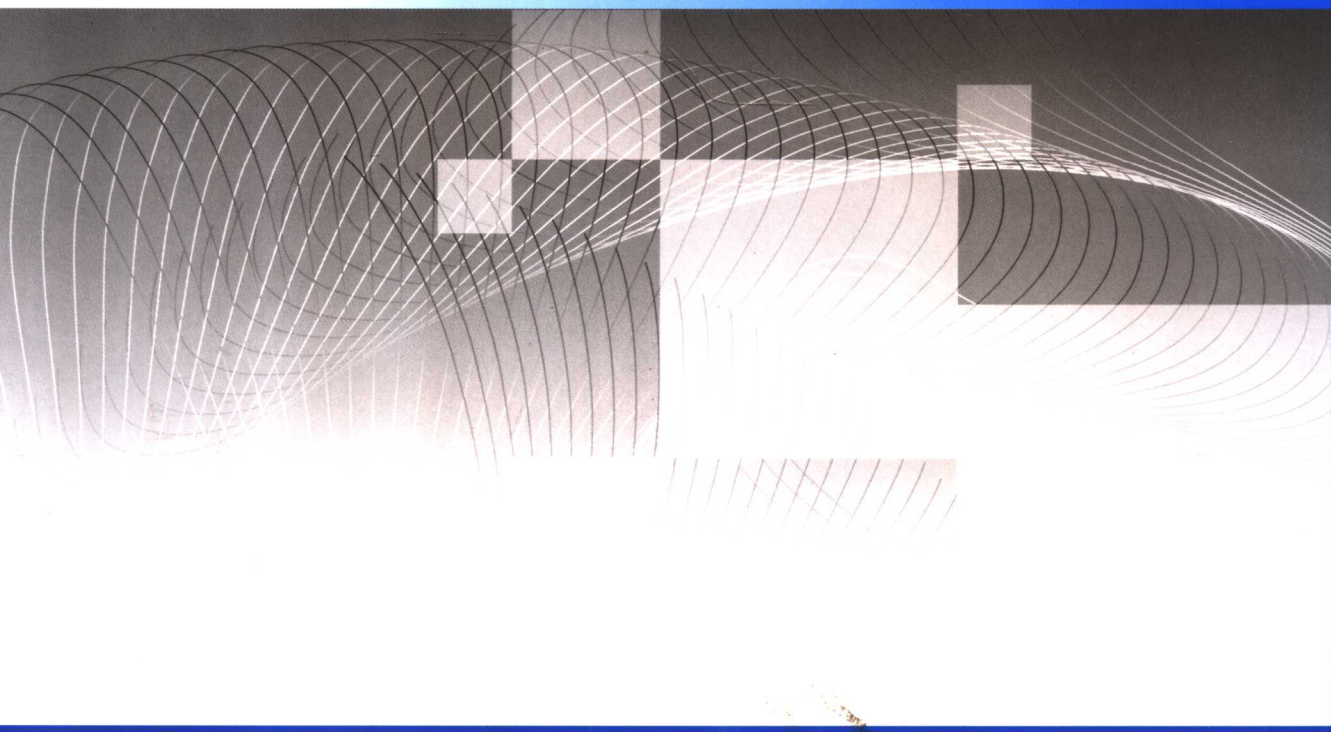


Microsoft®

微软院校课程系列教材

Visual C# 2005 程序设计语言

■ 微软公司 著



高等教育出版社
Higher Education Press

微软院校课程系列教材

Visual C# 2005 程序设计语言

微软公司 著

高等教育出版社

本书的著作权归微软公司所有。未经微软公司书面许可,本书的任何部分不得以任何形式或任何手段复制或传播。著作权人保留所有权利。

图书在版编目(CIP)数据

Visual C# 2005 程序设计语言 / 微软公司著. —北京:
高等教育出版社, 2007.2

ISBN 978-7-04-021614-1

I. V… II. 微… III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2007)第 017386 号

策划编辑 尹 洪 责任编辑 郭福生 封面设计 王凌波 责任绘图 朱 静
版式设计 张 岚 责任校对 胡晓琪 责任印制 陈伟光

出 版 高等教育出版社
社 址 北京市西城区德外大街 4 号
邮政编码 100011
总 机 010-58581000

购书热线 010-58581118
免费咨询 800-810-0598
网 址 <http://www.hep.edu.cn>
<http://www.hep.com.cn>

印 刷 北京民族印刷厂

开 本 787×1092 1/16
印 张 15.5
字 数 370 000

版 次 2007 年 2 月第 1 版
印 次 2007 年 2 月第 1 次印刷
定 价 59.00 元(含光盘)

版权所有 侵权必究
物料号 21614-00

编审委员 刘志鹏 林 波 郑祖宪 王 林 田本和
组织策划 林 波 蒋 斌 尹 洪 Jim Dilanni 徐志献
李朝晖 周雨阳 冯英
技术编审 蒋 斌 张 充 虞谷晔 王 伟 韩 俊
张广军 洪国芬

Microsoft Official Curriculum 最终用户许可协议

重要须知——请认真阅读——您一旦打开“许可使用内容”包装的密封或以其他方式使用此处的“许可使用内容”，即表示您同意接受本《协议》各项条款的约束：本 Microsoft Official Curriculum 可能包含 Microsoft 或其供应商提供的软件或其他材料（总称“许可使用内容”），其使用应遵守以下各项 Microsoft 提示条款。每个软件程序都受一份最终用户许可协议（《协议》）的约束，而该《协议》是您（个人或单一实体）（“最终用户”）和 Microsoft Corporation（“Microsoft”）之间就允许使用软件及相关介质或印刷材料、“联机”或电子文档和基于 Internet 的服务达成的一份法律协议。本《协议》的修正条款或补充条款可能随软件一起提供。您一旦安装、复制或或以其他方式使用“许可使用内容”，即表示您同意接受本《协议》各项条款的约束。如果您不同意，请（a）不要打开“许可使用内容”包装的密封；（b）不要使用软件、文档或其他材料，并且（c）退还“许可使用内容”。

“许可使用内容”随带软件的特别提示

许可证的授予。为与本“许可使用内容”一起使用而提供的任何软件（“软件”）都是 Microsoft Corporation 和（或）其供应商享有著作权的作品。“软件”只授予使用许可，而非出售。任何特定“软件”的使用都应遵守以下各《许可协议》中的一份《许可协议》：

（1）一般使用许可。Microsoft 授予最终用户一份有限的、非独家拥有的、免版权费的许可证，许可其在一台由一位单一用户随时使用或访问的单一计算机上安装和使用“软件”的一份副本，并且最终用户：（a）不得修改“软件”，但下文有明确规定时例外；（b）不得发行“软件”或其任何组成部分；（c）不得出借、出租、租赁、出售、分许可、转让“软件”或将“软件”随附的任何印刷材料用于提供商业运营服务；（d）不得在收费的公立或私立课程中使用“软件”；（e）不得对“软件”进行反向工程、反编译或反汇编；尽管有此项限制，但如果适用法律明确允许上述活动并且仅在适用法律明示允许上述活动的范围内，则例外；并且（f）不得转让“软件”的各项权利，除非本《协议》明确规定。

Microsoft 保留一切其他权利。Microsoft 及其供应商保留“软件”的一切产权和所有权，并且不转让或许可使用“软件”或其任何组成部分的任何权利，除非本《协议》具体说明。

（2）替代使用许可。上述规定的一般使用许可将被任何具体“软件”随附或包括的《许可协议》（如果有）的各项条款取代或替代。除非最终用户首先同意《许可协议》的各项条款，否则将无法安装附带或包括该《许可协议》的“软件”。Microsoft 保留一切其他权利。Microsoft 及其供应商保留“软件”的一切产权和所有权，并且不转让或许可使用“软件”或其任何组成部分的任何权利，除非本《协议》明确说明。

（3）样本代码使用许可。如果将特定代码或一个样本应用程序作为“许可使用内容”中包括的实验室练习的部分提供（“样本代码”），则这类“样本代码”以“现有状况”被提供，并且没有任何类型的保证。Microsoft 授予您一份有限的、非独家拥有的、免版权费的许可证，

许可您为了个人使用的目的而安装、使用、修改和复制“样本代码”，条件是您不得：(a) 发行“样本代码”或其任何组成部分；(b) 出借、出租、租赁、出售、分许可或转让“样本代码”；(c) 在收费的公立或私立课程中使用“样本代码”；并且(或者)(d) 转让“样本代码”的任何权利。如果您修改“样本代码”，您应该根据 Microsoft 的请求，自付费用为因您或代表您对“样本代码”做出的任何修改而使 Microsoft 和 Microsoft 的分公司、关联公司、董事、高级主管、员工、代理商和独立供应商面临的任何索赔或诉讼提供辩护，并且您须赔偿 Microsoft 因这类索赔而招致的任何费用、损害赔偿和手续费方面的合理开支（其中包括但不限于律师费和其他专业人士收取的费用），并使其免受任何损害。Microsoft 应：(a) 以书面形式就任何这类索赔或诉讼向您提供合理的及时提示，并且允许您通过 Microsoft 和您双方都接受的律师对这类索赔或诉讼进行答辩和辩护；(b) 在您支付费用的情况下向您提供信息、协助和授权，以帮助您为这类索赔或诉讼进行辩护。您不对 Microsoft 在未经您书面允许的情况下做出的任何和解负责，但您不得以不合理的方式拒绝给予这样的允许。

其他许可限制。安装“软件”仅供最终用户根据适用的《许可协议》使用，并且除非以其他方式在另外一份协议中达成一致意见，否则得不到 Microsoft 或其供应商提供的技术或其他支持服务。法律明确规定：禁止在违反《许可协议》的情况下对“软件”进行任何复制或再发行。明确禁止为进一步复制或再发行软件而将“软件”复制到任何服务器或地点。

美国政府许可使用权利。根据 1995 年 12 月 1 日当天或之后签发的请求而提供给美国政府的所有软件，均根据本协议其他部分规定的商业许可使用权利和限制予以提供。根据 1995 年 12 月 1 日之前签发的请求而提供给美国政府的所有软件，视情况根据 FAR, 48 CFR 52.227-14 (1987 年 6 月) 或 DFAR, 48CFR252.227-7013 (1988 年 10 月) 中规定的“限制权利”予以提供。

免责条款。“软件”仅根据《许可协议》的各项条款对“软件”提供保证（如果提供保证的话）。除非在《许可协议》中提供保证，否则 Microsoft Corporation 和（或）其供应商就“软件”不提供任何的保证和条件，包括适销性、适用性、所有权和不侵权的所有默示保证和条件。

“许可使用内容”随带文档和（或）其他材料的具体说明

允许从“许可使用内容”（“文档”）中打印文档（如实验室说明等），条件是：(a) 将这类文档用于您的个人培训，并且不得再出版或在任何网络计算机上张贴或以任何介质形式广播这类文档，并且 (b) 不得对任何文档做出任何修改。

明确禁止对任何介质上包含的作为“许可使用内容”组成部分的录像、录音、图形和（或）任何其他材料（“其它材料”）进行任何复制或再发行。

“许可使用内容”的各组成部分均受商业包装法律和其他法律的保护，并且不得全部或部分予以复制或模仿。除非 Microsoft 明示允许，否则不得复制或转发“许可使用内容”中的任何徽标、图形、声音或图像。

无保证。Microsoft 和（或）其供应商不对“许可使用内容”中不论为任何目的而可能包含的文档或其他材料中的信息、音像或任何其他内容是否合适提供任何保证，无论该类文档、信息、音像或任何其他内容是为何目的。所有这类文档和其他材料均以“现有状况”提供，没有

任何类型的保证。Microsoft 和（或）其供应商特此就文档和其他材料不提供任何的保证和条件，包括适销性、适用性、所有权和不侵权的所有默示保证和条件。

有关第三方站点链接的说明

至第三方站点的链接。您可以使用“许可使用内容”链接至第三方站点。第三方站点不由 Microsoft 控制，并且 Microsoft 不对任何第三方站点的内容、第三方站点包含的任何链接或第三方站点的任何更改或更新负责。Microsoft 不对从任何第三方站点收到的网站广播或任何其他形式的传输负责。Microsoft 仅为了您的方便向您提供这些至第三方站点的链接，并且包括任何链接并不暗示 Microsoft 认可相应的第三方站点。

有关全部“许可使用内容”的说明

“许可使用内容”中包括的“软件”、文档和其他材料可能包含不准确的技术内容或印刷错误。可能定期对内容进行修订。Microsoft 可随时在不提供通知的情况下对“许可使用内容”中规定的产品和（或）程序进行改进和（或）更改。

免责条款。除非另行说明，否则本《协议》提及的公司、产品、人物、特性和（或）数据均属虚构，并且无意以任何方式代表任何真实的个人、公司、产品或活动。

保留权利和所有权。Microsoft 保留未在本《协议》中明示授予您的一切权利。“许可使用内容”受著作权和其他知识产权法律及条约的保护。Microsoft 或其供应商拥有“许可使用内容”和其中组件的所有权、著作权和其他知识产权。

同意使用数据。您同意：Microsoft 及其关联公司可以收集和使用作为提供给您的产品支持服务的一部分而收集的与“许可使用内容”相关的技术信息（如果有）。Microsoft 可以将此信息仅用于改进我们的产品或为您提供定制的服务或技术，并且不会以能识别您身份的方式披露此信息。

额外软件/服务。除非我们随下列更新、增补、补充组件或基于 Internet 的服务组件一起提供其他应适用的条款，否则本《协议》适用于 Microsoft 在您获得“许可使用内容”的初始副本之日后可能提供给您的或为您准备的“许可使用内容”的更新、增补、补充组件或基于 Internet 的服务组件。就通过使用“许可使用内容”而提供给您的或为您准备的任何基于 Internet 的服务而言，Microsoft 保留停止这类服务的权利。

出口限制。您承认“软件”受美国出口法律管辖。您同意遵守所有适用于“软件”的国际法和国内法，其中包括美国出口管理条例以及由美国和其他国家（地区）政府颁发的最终用户、最终使用和目的地方面的限制。要了解详情，请访问 <http://www.microsoft.com/exporting/> 网站。

许可使用内容的转让。“许可使用内容”的原始最终用户可以将本《协议》和“许可使用内容”永久性地一次直接转让给另外一位最终用户，条件是该原始用户不得保留“许可使用内容”的任何副本，并且必须转让“许可使用内容”的所有部分（包括全部组件、介质及印刷材料、任何升级版本、各《许可协议》和（如果适用）正版标签）。这种转让不得为非直接转让，如以寄售方式转让。在转让之前，接收“许可使用内容”的最终用户必须同意遵守《协议》的

各项条款。如果“许可使用内容”是一个升级版本，任何转让都必须包括“许可使用内容”的所有先前版本。

终止。如果您未遵守本《协议》的各项条款和条件，在不损害其他权利的情况下，Microsoft 可终止本《协议》。如此类情况发生，您必须销毁“许可使用内容”的所有副本及其全部组成部分。

适用法律。本《协议》受中华人民共和国法律管辖。

责任限制。在适用法律所允许的最大范围内，无论损害赔偿是否在履行合约、出现疏忽或发生其他侵权行为时发生，Microsoft 和（或）其供应商绝不就因“许可使用内容”的任何组成部分或所有组成部分的使用或性能、因提供或未能提供服务、或因可从“许可使用内容”得到的信息而引起的或有关的任何特殊的、间接的、或特定的损害赔偿或任何损害赔偿（包括但不限于因营业中断，因使用、数据或利润的丧失，或因任何其他金钱上的损失而造成的损害赔偿）承担赔偿责任。在任何情况下，Microsoft 的全部责任以及您获得的惟一赔偿将限于为“许可使用内容”实际支付的款额或五美元（U.S.\$5.00）以两者中的较高款额为准；但是，如果您已经签订了一份 Microsoft 服务协议，Microsoft 对这类服务的全部责任将遵守该协议各项条款的规定。由于某些国家和地区不允许排除或限制责任，上述限制条款可能不适用于您。

全部协议；规定可分割性。本《协议》（包括随“许可使用内容”提供的本《协议》的任何补充条款或修正条款）是您与 Microsoft 之间就“许可使用内容”和支持服务（如果有）达成的全部协议，并且取代“许可使用内容”或本《协议》中所包含的任何其他标之所有先前或同时存在的口头或书面的通信、建议和声明。如果任何 Microsoft 支持服务的政策或计划的条款与本《协议》的条款有冲突，以本《协议》的条款为准。如果本《协议》的任何条款被认定为作废、无效、不能执行或非法，其他条款应继续完全有效。

如果您对本《协议》有任何疑问，或者如果您由于某种原因希望与 Microsoft 联系，请使用“许可使用内容”中附带的地址信息与微软（中国）有限公司联系，或在<http://www.microsoft.com> 网站访问 Microsoft。

准则和定义

“许可使用内容”是一种专门设计的培训工具，供 Microsoft Certified Technical Education Center (Microsoft CTEC)、Microsoft Certified Partner (MCP)、Microsoft 认证培训讲师 (MCT)、IT Academy 计划成员和 Microsoft 可能随时以书面形式指定的其他机构使用。“许可使用内容”旨在使 Microsoft 的技术培训渠道能够向计算机专业人士提供系统、支持和开发培训课程。为了取得最佳成果，“许可使用内容”应该由 Microsoft 认证培训讲师 (MCT) 在课堂环境或在线学习环境中讲授。

Microsoft Official Curriculum (MOC)：由 Microsoft 开发的系列课程材料，用于提供 Microsoft 产品和技术的培训和解决方案。

Microsoft 认证培训讲师 (MCT)：具备必要的教学和技术能力并且由 Microsoft 认证为能够通过 Microsoft CTEC 讲授 Microsoft Official Curriculum 的个人。

Microsoft Certified Technical Education Center (Microsoft CTEC)：已经符合 Microsoft 对

指定其为下列场所的资格要求：（a）一处 Microsoft Certified Partner（MCP）营业点、和（b）一处提供 Microsoft CTEC 服务的任何场所。这些培训中心使用 MCT 向学生提供 MOC 课程培训。

Microsoft Certified Partner: 已经符合被指定为 Microsoft Certified Partner 的资格要求的任何场所。

IT Academy 计划成员: 已经符合被指定为 IT Academy 计划成员的资格要求的任何院校。

目 录

| | |
|---|-----------------------------------|
| 第 1 章 概述1 | 2.3.5 窗体和控件..... 36 |
| 1.1 计算机软件与硬件相关知识简介.....1 | 2.3.6 代码视图..... 36 |
| 1.1.1 计算机软件.....1 | 2.3.7 程序分析..... 36 |
| 1.1.2 计算机硬件.....2 | 2.4 小结..... 37 |
| 1.2 以编程方式解决问题的一般方法.....3 | 2.5 实验..... 37 |
| 1.2.1 随堂练习.....4 | 2.6 习题..... 37 |
| 1.2.2 数据结构和算法.....4 | 第 3 章 变量、数据类型和表达式 39 |
| 1.3 编程语言和开发工具.....5 | 3.1 变量..... 39 |
| 1.3.1 编程语言.....5 | 3.1.1 命名变量..... 39 |
| 1.3.2 开发工具.....6 | 3.1.2 随堂练习..... 40 |
| 1.3.3 安装 Visual Studio 2005.....6 | 3.1.3 声明变量..... 40 |
| 1.3.4 创建 C#控制台应用程序.....11 | 3.2 常量..... 40 |
| 1.4 小结.....15 | 3.3 C#数据类型..... 41 |
| 1.5 实验.....15 | 3.3.1 引用类型..... 41 |
| 1.6 习题.....16 | 3.3.2 值类型..... 41 |
| 第 2 章 C#与 Visual Studio 200517 | 3.3.3 泛型..... 45 |
| 2.1 C#.....17 | 3.4 类型转换..... 45 |
| 2.2 Visual Studio 2005 集成开发环境简介.....18 | 3.4.1 隐式类型转换..... 45 |
| 2.2.1 使用起始页.....18 | 3.4.2 显式类型转换..... 49 |
| 2.2.2 随堂练习.....19 | 3.4.3 随堂练习..... 50 |
| 2.2.3 “选项”对话框.....19 | 3.5 运算符和表达式..... 51 |
| 2.2.4 解决方案资源管理器.....21 | 3.5.1 运算符类型..... 51 |
| 2.2.5 类视图.....24 | 3.5.2 运算符的优先级..... 52 |
| 2.2.6 “属性”对话框.....26 | 3.5.3 简单的赋值运算符与赋值表达式..... 53 |
| 2.2.7 工具箱.....28 | 3.5.4 算术运算符与算术表达式..... 53 |
| 2.2.8 智能感知.....28 | 3.5.5 关系运算符与关系表达式..... 54 |
| 2.2.9 使用帮助.....29 | 3.5.6 逻辑运算符与逻辑表达式..... 55 |
| 2.3 可视化开发的初步认识.....32 | 3.5.7 位操作符与位操作表达式..... 56 |
| 2.3.1 编程步骤.....32 | 3.5.8 扩充后的赋值运算符与赋值表达式..... 58 |
| 2.3.2 创建用户界面.....33 | 3.5.9 条件运算符与条件表达式..... 59 |
| 2.3.3 设置属性.....34 | 3.6 小结..... 59 |
| 2.3.4 运行 Windows 应用程序.....36 | 3.7 实验..... 59 |

| | | | |
|----------------------------------|------------|--------------------------------|------------|
| 3.8 习题 | 59 | 6.1.1 派生类 | 117 |
| 第4章 分支和循环 | 61 | 6.1.2 在派生类中调用基类构造函数 | 119 |
| 4.1 程序的三种结构 | 61 | 6.1.3 密封类 | 121 |
| 4.1.1 顺序结构 | 61 | 6.2 多态性 | 121 |
| 4.1.2 分支结构 | 61 | 6.2.1 编写虚方法 | 121 |
| 4.1.3 循环结构 | 62 | 6.2.2 抽象方法和抽象类 | 125 |
| 4.2 条件语句 | 62 | 6.3 接口 | 127 |
| 4.2.1 if 语句 | 62 | 6.3.1 什么是接口 | 127 |
| 4.2.2 随堂练习 | 64 | 6.3.2 接口的使用方法 | 128 |
| 4.2.3 switch 语句 | 64 | 6.3.3 如何使用实现了某接口的对象 | 130 |
| 4.2.4 随堂练习 | 67 | 6.3.4 如何继承多个接口 | 131 |
| 4.2.5 goto 语句 | 68 | 6.3.5 接口与抽象类的比较 | 133 |
| 4.2.6 在 Visual Studio 2005 中跟踪程序 | 70 | 6.4 小结 | 133 |
| 4.3 循环语句 | 72 | 6.5 实验 | 134 |
| 4.3.1 while 循环 | 72 | 6.6 习题 | 134 |
| 4.3.2 do...while 循环 | 74 | 第7章 程序的生成、调试和异常处理 | 135 |
| 4.3.3 for 循环 | 74 | 7.1 在 Visual Studio 2005 中生成程序 | 135 |
| 4.3.4 foreach 循环 | 77 | 7.2 Visual Studio 2005 的调试功能 | 136 |
| 4.3.5 continue 和 break 在循环中的应用 | 78 | 7.2.1 Visual Studio 2005 调试器 | 136 |
| 4.4 小结 | 80 | 7.2.2 随堂练习 | 138 |
| 4.5 实验 | 80 | 7.3 异常处理 | 139 |
| 4.6 习题 | 80 | 7.3.1 异常类 | 139 |
| 第5章 面向对象 | 82 | 7.3.2 使用异常 | 141 |
| 5.1 理解面向对象 | 82 | 7.3.3 System.Exception 的常用属性 | 144 |
| 5.1.1 用面向对象的一般方法解决问题 | 82 | 7.3.4 默认异常处理 | 145 |
| 5.1.2 用面向对象的程序解决问题 | 84 | 7.3.5 嵌套的 try 块 | 145 |
| 5.2 面向对象的概念 | 85 | 7.3.6 用户定义的异常类 | 146 |
| 5.3 面向对象程序设计 | 86 | 7.4 小结 | 148 |
| 5.3.1 抽象 | 86 | 7.5 实验 | 148 |
| 5.3.2 封装 | 87 | 7.6 习题 | 148 |
| 5.3.3 类 | 87 | 第8章 数组与字符串 | 151 |
| 5.3.4 类成员 | 89 | 8.1 数组的概念 | 151 |
| 5.3.5 委托 | 111 | 8.2 声明和创建数组 | 152 |
| 5.4 小结 | 114 | 8.2.1 声明数组变量 | 152 |
| 5.5 实验 | 114 | 8.2.2 创建数组实例 | 152 |
| 5.6 习题 | 114 | 8.3 初始化数组变量 | 153 |
| 第6章 面向对象的高级应用 | 116 | 8.4 访问单个数组元素 | 154 |
| 6.1 继承性 | 116 | 8.5 随堂练习 | 154 |

| | | | |
|-------------------------------|------------|-------------------------------------|------------|
| 8.6 遍历数组元素 | 154 | 9.4.10 随堂练习 | 193 |
| 8.7 复制数组 | 156 | 9.4.11 创建和使用状态栏 | 193 |
| 8.8 如何把数组作为方法参数 | 157 | 9.4.12 随堂练习 | 195 |
| 8.9 随堂练习 | 158 | 9.4.13 创建和使用通用对话框 | 195 |
| 8.10 System.Array 类 | 159 | 9.4.14 随堂练习 | 200 |
| 8.10.1 AsReadOnly()方法 | 160 | 9.5 小结 | 201 |
| 8.10.2 Clear()方法 | 160 | 9.6 实验 | 201 |
| 8.10.3 Copy()方法 | 161 | 9.7 习题 | 201 |
| 8.10.4 CreateInstance()方法 | 163 | 第 10 章 创建 Web 应用程序 | 203 |
| 8.10.5 Sort()方法 | 163 | 10.1 HTML 简介 | 203 |
| 8.11 常用字符串处理函数 | 165 | 10.1.1 标记语法和文档结构 | 204 |
| 8.11.1 理解字符串 | 165 | 10.1.2 一个简单的例子 | 205 |
| 8.11.2 字符串常用方法 | 165 | 10.2 ASP.NET | 205 |
| 8.12 小结 | 169 | 10.3 创建 Web 窗体应用程序 | 207 |
| 8.13 实验 | 169 | 10.3.1 安装 IIS | 207 |
| 8.14 习题 | 169 | 10.3.2 创建 ASP.NET 网站 | 207 |
| 第 9 章 基于 Windows 的应用程序 | 171 | 10.3.3 检查本地 IIS 网站的结构 | 208 |
| 9.1 Windows 窗体应用程序概述 | 171 | 10.3.4 添加控件和事件处理程序 | 209 |
| 9.1.1 窗体 | 171 | 10.3.5 生成并运行 Web 窗体页面 | 209 |
| 9.1.2 控件 | 171 | 10.4 ADO.NET | 210 |
| 9.1.3 事件 | 172 | 10.5 XML Web Service | 214 |
| 9.1.4 控件的生存周期 | 172 | 10.5.1 创建一个简单的 XML Web Service | 215 |
| 9.2 Windows 窗体控件介绍 | 172 | 10.5.2 调用 XML Web Service | 216 |
| 9.3 Windows 窗体控件的共同特性 | 174 | 10.6 小结 | 218 |
| 9.4 常用 Windows 窗体控件介绍 | 176 | 10.7 习题 | 218 |
| 9.4.1 创建窗体的主菜单 | 176 | 第 11 章 .NET Framework 2.0 简介 | 219 |
| 9.4.2 随堂练习 | 178 | 11.1 .NET Framework 概述 | 219 |
| 9.4.3 创建和使用工具栏 | 178 | 11.2 .NET Framework 类库 | 220 |
| 9.4.4 随堂练习 | 182 | 11.3 小结 | 221 |
| 9.4.5 使用 Label 类 | 183 | 11.4 习题 | 222 |
| 9.4.6 使用 TextBox 类 | 183 | 词汇表 | 223 |
| 9.4.7 使用按钮类 | 185 | 附录 | 232 |
| 9.4.8 使用 ListBox 类 | 189 | | |
| 9.4.9 使用 ComboBox 类 | 191 | | |

第1章 概述

当今社会，软件无处不在。软件是由软件开发人员开发而成的。软件开发人员根据要实现的功能，编写出可以被计算机执行的指令，从而控制计算机完成相应的功能。

计算机可以直接识别的“语言”（指令）是二进制代码。普通开发人员直接用二进制代码编写软件是困难的。于是，高级编程语言诞生了。高级编程语言的代码近似于人类的语言，可以转换成为二进制代码执行。这样，开发软件就方便多了。

在软件开发过程中，开发人员需要编写大量的代码。同时，他们也会遇到各种导致工作效率下降的问题，例如，输入代码时出现语法错误，程序调试效率低下，等等。集成开发环境（Integrated Development Environment, IDE）的出现缓解了这些问题。它集成了从代码编写到调试、排错等各种开发工具，减轻了开发人员的工作负担，提高了开发效率。

现在的软件已经越来越复杂和庞大，开发工作一般都是在某个软件（也叫做运行环境）的基础上进行开发的。开发出来的软件通过调用下层软件的功能来实现。例如，记事本软件 Notepad 是在 Windows 操作系统上开发出来的。Windows 操作系统就是 Notepad 的运行环境。Notepad 对计算机的操作都是通过调用 Windows 的功能来实现的。

作为一名开发人员，他的开发工作基本上都离不开上面的三个要素：编程语言、集成开发环境和运行环境。具体到本书中就是 C# 编程语言、Visual Studio 集成开发环境和 .NET Framework 运行环境。在接触具体语言、开发环境和运行环境之前，我们先来详细了解一些基础知识。学完本章后你可以：

- 了解最基本的计算机软、硬件知识。
- 了解使用编程方式来解决实际问题的思想和过程。
- 初步了解数据结构和算法的概念。
- 了解编程的概念和常用的编程工具。
- 掌握安装 Visual Studio 2005 的方法。
- 了解简单 C# 程序的基本结构。
- 掌握 C# 的代码格式。

1.1 计算机软件与硬件相关知识简介

1.1.1 计算机软件

计算机软件是指令、数据和文档的集合。它具有以下两个特征：

- 指令按照特定顺序组织，能够使计算机具有信息处理能力。
- 标志一定功能、完成一定任务或产生一定结果。

比如 Windows 中的“记事本”工具，它是由一系列程序指令组成的，可以完成文字处理的任务，能够输入或输出文字数据，还包括一个帮助说明文档。这就是一个软件。

软件分为系统软件和应用软件。

系统软件中最典型的的就是操作系统（Operating System, OS）。操作系统管理、控制各种计算机硬件资源（如内存、处理器和硬盘空间等），把这些资源分配给运行在操作系统上的应用程序，同时还对这些应用程序进行控制和管理。比如，平时我们玩的游戏，程序本身不具备直接操作硬件设备（如显卡和声卡）的能力，它需要调用操作系统提供的接口才能实现。

应用软件运行在操作系统上，实现各种应用功能。它们不能脱离操作系统而独立运行，甚至不能在不同的操作系统上运行，比如 Internet 浏览器、Office 软件套装和 Windows Media Player 多媒体播放软件等，这些软件都实现不同的应用，又基于 Windows 操作系统实现各自的功能，它们都是应用软件。

在我们平时的使用中，最常见的两种应用软件是 Windows 应用软件和 Web 应用软件。

Windows 应用软件就是基于 Windows 平台的应用软件。这类软件通过调用 Windows 平台提供的各种接口来实现自己的功能，有很强的平台依赖性。

Web 应用软件一般是基于浏览器/服务器（Browser/Server, B/S）架构的软件，这类软件的客户端不依赖于特定的平台，只需要浏览器能正常显示 HTML 编写的代码和脚本语言（如 VBScript 和 JavaScript）编写的脚本。大部分的功能都由服务器端提供。例如，在线电子商务网站、BBS 论坛、基于企业内部网的报价单或销售报告的管理和生成工具等。

时下流行的 XML Web Service 也可以算是一种 Web 应用软件，它以 XML 和 SOAP 协议为标准，可以和其他应用程序相结合构建分布式系统。例如，某网站提供了天气信息查询的 XML Web Service。有一个程序员想自己编写一个桌面提醒程序，可以每天提醒用户注意当天的天气。要实现此功能，程序员只要实现桌面提醒的功能即可，而天气信息可以通过调用天气查询的 XML Web Service 获得，不需要自己去开发一套天气信息查询系统。

本书所介绍的 C#语言可以在 .NET Framework 开发平台的基础上，构建上述所说的各种应用程序，包括 Windows 应用软件和 Web 应用软件。

1.1.2 计算机硬件

软件是运行在硬件上的，那么软件又是如何使用各种硬件设备的呢？为此，首先来了解一下计算机硬件的组成。

冯·诺依曼体系结构指出，计算机应该遵循如下原则：

- 把计算机要执行的指令和要处理的数据都采用二进制表示。
- 把要执行的指令和要处理的数据按照顺序编成程序，存储到计算机内部让它自动执行。

冯·诺依曼体系结构是大部分现代计算机运行的理论基础。现代计算机的结构主要由以下几部分组成。

① CPU（Central Processing Unit）——又称中央处理器。其主要功能是进行算术和逻辑运算，内部结构大概可以分为控制单元、算术逻辑单元和存储单元等几个部分。按照其处理信息的字长可以分为 8 位微处理器、16 位微处理器、32 位微处理器以及 64 位微处理器等。

② 存储器——在计算机中用来存放信息的部件。现代计算机中的存储器一般都由内存和

外存组成。内存由半导体元件组成，可分为两类：只读存储器（Read-Only Memory, ROM）和随机存储器（Random Access Memory, RAM）。只能读出 ROM 中的信息，不能向其中写入信息，断电时信息不会丢失，典型应用就是基本输入/输出系统（Basic Input/Output System, BIOS）。RAM 中的信息可读、可写、可修改，只要计算机断电，信息就会丢失。由于 RAM 中的信息在计算机断电时会丢失，因此，应及时地将 RAM 中的信息转移。外存储器主要包括软盘、硬盘和光盘。硬盘是计算机主要的存储设备，容量大、读/写速度快。光盘是一种新的存储器，容量大，相对于硬盘而言其速度较慢，在计算机中必须依赖光盘驱动器（光驱）来读取光盘上的信息。

③ 输入设备——用于向计算机软件输入指令或者要处理的数据。最常见的计算机输入设备包括鼠标和键盘。这也是标准配置。此外还可能有触摸屏、手写板、语音识别器等设备可以接受用户输入。

④ 输出设备——输出设备种类多样，标准的输出设备是显示器。此外可能还包含音箱和打印机等。

在了解了计算机的基本硬件组成以后，你也许要问软件是如何在这样的计算机系统上执行的呢？

首先软件需要存储在一种永久性存储介质上，可能是硬盘或光盘上。在软件要运行的时候，计算机系统会把软件加载到内存中，然后由 CPU 顺序或者并行地执行软件指令。在此过程中，软件可能需要用户从输入设备上输入一些数据进行处理，同时以各种形式在输出设备上把提示和结果信息传递给用户，如使用音箱播放声音、在显示器上显示图像等。

1.2 以编程方式解决问题的一般方法

首先，来看一个例子：在期末考试结束以后，计算一下本班的数学成绩平均分。

如果要以程序来解决这个问题，一般可以遵循如下流程：

① 分析。也就是定义问题。一定要明确问题的实质是什么，即分析什么是最终输出，什么是必要输入，必要输入是如何变成最终输出的。

上面这个问题中，最终输出的是本班的数学成绩平均分。要获得平均分，必须获得两个数据：本班数学成绩的总和以及本班人数。

② 设计。设计出解决问题的方法。根据问题找到正确解决问题的一套逻辑方法，通常被称为算法。

在这个问题中，核心算法为：

$$\text{本班数学成绩平均分} = \text{本班数学成绩的总和} / \text{本班人数}$$

③ 选择用户界面。确定以什么方式接受用户输入。

制作一个应用程序界面。用不同的文本框来输入成绩，显示人数和最终的平均分。用户可以用“输入”按钮确认输入成绩，用“计算”按钮确认成绩输入完毕，同时计算并显示平均分。

| | | |
|-----------------------------------|----------------------|-----------------------------------|
| 数学成绩平均分 | <input type="text"/> | <input type="button" value="计算"/> |
| 输入数学成绩 | <input type="text"/> | 人数 <input type="text"/> |
| <input type="button" value="输入"/> | | |

④ 代码实现。用编程语言把算法转换成程序代码的过程。

用 C# 编写代码实现上面的例子。

⑤ 测试和调试。查找并解决程序中的任何错误。因为很难保证所编写的程序中不会有一些小的失误，那么就要在编程结束后，以各种形式对程序进行测试，并解决所发现的问题。

⑥ 完成程序的文档说明。包括对问题的描述和程序的描述，你是如何解决问题的，用户界面如何使用，等等。

1.2.1 随堂练习

编写一个“数学成绩平均分计算器”的使用说明。说明用户如何输入成绩，并且得到平均分。

1.2.2 数据结构和算法

计算机解决问题的实质就是高速的算术和逻辑运算。对于一个程序来说，其本质就是计算方法和数据，用术语来说就是算法和数据结构。

算法是指在有限步骤内求解某一问题所使用的一组定义明确的规则。简单地说，就是计算机解题的过程。

在实际开发过程中，首先要建立解决问题的算法，然后再用程序来实现算法。比如上题中计算数学平均成绩，首先得出计算平均成绩的算法，然后用程序来实现。

一个算法应该具有以下 5 个重要的特征：

- 有穷性：一种算法必须保证执行有限步骤之后结束。
- 确切性：算法的每一步骤必须有确切的定义。
- 输入：一种算法有 0 个或多个输入，以刻画运算对象的初始情况。
- 输出：一种算法有一个或多个输出，以反映对输入数据进行加工后的结果。
- 可行性：算法原则上能够精确地运行，而且人们用笔和纸做有限次运算后即可完成。

算法是程序的灵魂。算法的优劣直接决定了程序的执行效率和质量。例如，对于计算机下棋程序，算法好的程序，同一时间可以搜索的棋步比算法差的程序高出很多，那么它就会比后者“聪明”很多。

数据结构是计算机存储和组织数据的方式。作为一门学科，数据结构主要研究数据的各种逻辑结构和存储结构以及对数据的各种操作。数据结构用于将真实世界中的各种数据有效地呈现和存储在计算机中。这就好比是在仓库中存放货物，不同的货物有不同的包装和存放方法。包装、存放合理，取用的时候就方便，反之则会导致很多问题。

数据结构和算法是密切相关的。不同的数据结构往往需要不同的算法对其进行操作。

1.3 编程语言和开发工具

1.3.1 编程语言

编程语言通常分为 3 类：即机器语言、汇编语言和高级语言。

1. 机器语言

机器语言是用二进制代码表示的、计算机能直接识别和执行的一组机器指令的集合。它是计算机的设计者通过计算机的硬件结构赋予计算机的操作功能。机器语言具有灵活、直接执行和速度快等特点。

用机器语言编写程序，编程人员首先要熟知所用计算机的全部指令代码和代码的含义。编写程序时，程序员得自己处理每条指令和每一数据的存储分配和输入/输出，还得记住编程过程中每步所使用的存储单元处于何种状态。这是一件十分繁琐的工作，编写程序花费的时间往往是实际运行时间的几十倍或几百倍。而且，编出的程序全是 0 和 1 的指令代码，直观性差，还容易出错。现在，除了计算机生产厂家的专业人员外，绝大多数程序员已经不必学习机器语言了。

2. 汇编语言

为了克服机器语言难读、难编、难记和易出错的缺点，人们就用与代码指令实际含义相近的英文缩写词、字母和数字等符号来取代指令代码（如用 ADD 表示运算符号“+”的机器代码），于是就产生了汇编语言。所以说，汇编语言是一种采用了助记符的、仍然是面向机器的计算机语言。汇编语言采用助记符来编写程序，比用机器语言的二进制代码编程要方便些，在一定程度上简化了编程过程。汇编语言的特点是用符号代替了机器指令代码，而且助记符与指令代码一一对应，基本上保留了机器语言的灵活性。使用汇编语言能面向机器并较好地发挥机器的特性，得到执行效率较高的程序。

由于汇编语言中使用了助记符，用汇编语言编制的程序送入计算机，计算机不能像对待机器语言编写的程序那样直接识别和执行，必须通过预先放入计算机的“汇编程序”的加工和翻译，才能变成能够被计算机识别和处理的二进制代码程序。用汇编语言等非机器语言书写好的符号程序称为源程序，运行时“汇编程序”要将源程序翻译成目标程序。目标程序是机器语言程序，它一旦被安置在内存的预定位置上，就能被计算机的 CPU 处理和执行。

汇编语言像机器指令一样，是硬件操作的控制信息，因而仍然是面向机器的语言，使用起来还是比较繁琐费时，通用性也差。汇编语言是低级语言。但是，使用汇编语言来编写系统软件和过程控制软件，其目标程序占用内存空间少，运行速度快，有着高级语言不可替代的优势。

3. 高级语言

不论是机器语言还是汇编语言，都是面向硬件进行具体操作的，语言对机器的过分依赖，要求使用者必须对硬件结构及其工作原理都十分熟悉，这对非计算机专业人员来说是难以做到的，对于计算机的推广应用也非常不利。计算机事业的发展，促使人们去寻求一些与人类自然语言相接近且能为计算机所接受的语义确定、规则明确、自然直观和通用易学的计算机语言。这种与自然语言相近并为计算机所接受和执行的计算机语言称为高级语言。高级语言是面向用