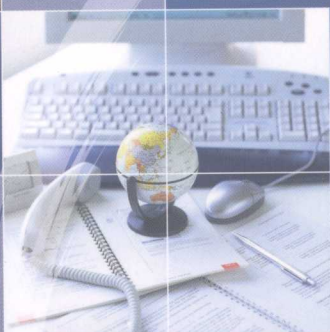




21st CENTURY
实用规划教材

21世纪全国应用型本科计算机案例型规划教材

C# .NET课程设计指导



computer

主 编 龚自霞 高 群
主 审 王立丰



北京大学出版社
PEKING UNIVERSITY PRESS

TP312/2746D

2008

21 世纪全国应用型本科计算机案例型规划教材

C# .NET 课程设计指导

主 编 龚自霞 高 群
副主编 潘志红 沈翠华
主 审 王立丰



北京大学出版社
PEKING UNIVERSITY PRESS

内 容 简 介

本书选择微软公司的 Visual C#作为编程语言,通过项目实例来详细讲解项目开发过程及方法。本书不仅介绍了项目设计的技术方法,还引入了软件开发项目管理的理念,使学生了解项目开发的过程及项目管理流程,为今后的职业生涯做好准备。

本书还配有一张光盘,提供了一个电子教学模板和案例的源代码,以方便教学和使用。

本书可作为高等院校计算机语言类课程的课程设计辅助教材,也适用于各类培训机构作为培训教材及编程爱好者作为提高程序开发能力的指导用书。

图书在版编目(CIP)数据

C#.NET 课程设计指导/龚自霞,高群主编. —北京:北京大学出版社,2008.1

(21 世纪全国应用型本科计算机案例型规划教材)

ISBN 978-7-301-12394-2

I. C… II. ①龚…②高… III. C 语言—程序设计—高等学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2007)第 083338 号

书 名: C#.NET 课程设计指导

著作责任者: 龚自霞 高 群 主编

策 划 编 辑: 孙哲伟 徐 凡

责 任 编 辑: 孙哲伟

标 准 书 号: ISBN 978-7-301-12394-2/TP · 0903

出 版 者: 北京大学出版社

地 址: 北京市海淀区成府路 205 号 100871

网 址: <http://www.pup.cn> <http://www.pup6.com>

电 话: 邮购部 62752015 发行部 62750672 编辑部 62750667 出版部 62754962

电 子 邮 箱: pup_6@163.com

印 刷 者: 世界知识印刷厂

发 行 者: 北京大学出版社

经 销 者: 新华书店

787 毫米×1092 毫米 16 开本 15.25 印张 353 千字

2008 年 1 月第 1 版 2008 年 1 月第 1 次印刷

定 价: 32.00 元(含 1 张光盘)

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究

举报电话: 010-62752024

电子邮箱: fd@pup.pku.edu.cn

前 言

本书是“Visual C#语言程序设计”的课程设计教程。为了配合 Visual C#语言课程的教学,提高学生的动手能力,加强编程技巧的训练,同时为适应软件开发项目管理流程,我们根据多年的教学及项目开发经验,精心设计了三个完整的项目开发案例,从软件项目开发管理的角度,详细介绍了项目开发的具体步骤。由于是具体的项目开发,因此,关键技术也是重要的环节。本书中针对一些项目开发的关键技术将有详细的讲解,同时还给出了一部分重点代码。

Visual C#是 Microsoft .NET 计划的核心编程语言,是微软公司 Visual Studio .NET 集成开发环境中的一员,具有简洁、灵活、安全、面向对象和高度兼容性等特点,兼具 Visual Basic 的高效性和 Visual C++的强大功能,是专门针对.NET 设计的一种语言,在.NET 中有不可替代的作用。它的大多数特点和 Java 一致,使用的符号和 C、C++接近。作为 Visual C#语言的课程设计,我们打破了以往以独立模块为中心的训练方式,取而代之的是用 Visual C#编写的完整的项目作为课程设计和综合实训的题目,结合企业标准的项目开发方法,目的是不仅使学生掌握一个一个实际的编程技巧,而且要掌握模块之间如何衔接,设计与实施如何协调,也就是要用系统的观点来进行项目设计,而不是个人英雄主义,一个人埋头写代码。

本书属于案例开发系列,以学生成绩管理系统的设计实现为基线,贯穿全书介绍 Visual C#的相关高级实现技术和理论,通过实例将知识点与开发实战紧密结合,从而达到学以致用目的。通读本书,不仅可以全面掌握 Visual C#开发知识,而且可以了解更多 Visual C# 应用技巧。研究本书中的大案例还有助于读者体会到使用 Visual C#开发大型应用程序时需要注意哪些问题,帮助读者建立软件开发中的大局观。

我们选择的项目为学生成绩管理系统的设计实现,整个课程设计的任务就是用系统的观点来完成此项目,因此,管理信息系统的系统分析就成了课程设计的分析,系统设计就成了课程设计的内容,系统实施就成了课程设计的实施过程。也就是说系统设计对应课程设计。

本书第 1 章介绍了标准课程设计的总体要求,阐述了课程设计的目的、对象,设计要求及设计计划的指定。第 2 章重点介绍了案例“学生成绩管理系统”按照软件开发的过程要求,详细阐述了从系统分析到系统实现的全过程。第 3 章介绍了第二个案例“餐馆管理系统”,给出了系统分析、概要设计和详细设计的过程,读者可根据所提供的资料进行课程设计。第 4 章给出了第三个案例,采用面向对象的分析方法设计和分析了“科教研管理信息系统”。第 5 章到第 7 章给出了三个习题,分别是“图书销售服务系统”和“电子邮件客户端程序”,以及采用结构化分析和设计的“学生评价教师系统”。

本书第 1 章和第 3 章由北京联合大学龚自霞和潘志红编写,第 2 章、第 6 章、第 7 章由龚自霞编写,第 4 章由中国农业大学沈翠华编写,第 5 章由山东理工大学高群编写。全书由百育炎华教育咨询有限公司王立丰主审,并由王立丰制作配套光盘。

由于编写仓促,加之编者水平有限,书中疏漏之处在所难免,敬请读者批评指正。

编 者

2007 年 12 月

目 录

第1章 课程设计详细说明	1
1.1 课程设计目的	1
1.2 课程设计对象	1
1.3 课程设计内容说明	1
1.4 课程设计要求	2
1.4.1 文档要求	2
1.4.2 开发要求	3
1.5 课程设计计划	16
1.5.1 项目计划	16
1.5.2 配置计划	20
1.5.3 开发计划	24
第2章 案例一：学生成绩管理系统	25
2.1 “学生成绩管理系统”的系统分析	25
2.2 “学生成绩管理系统”的概要设计	31
2.3 “学生成绩管理系统”的详细设计与系统实施	39
第3章 案例二：餐馆管理系统	133
3.1 “餐馆管理系统”的系统分析	133
3.2 “餐馆管理系统”的概要设计	137
3.3 “餐馆管理系统”的详细设计与系统实施	142
第4章 案例三：科教研管理系统	166
4.1 面向对象的分析	167
4.2 科教研管理信息系统设计与实施	176
第5章 习题一：图书销售服务系统	190
5.1 系统分析	190
5.2 系统概要设计	193
5.3 系统详细设计	196
第6章 习题二：学生评价教师系统	208
6.1 系统分析	208
6.2 系统概要设计	213
6.3 系统详细设计	215

第 7 章 习题三：电子邮件客户端程序	224
7.1 系统分析	224
7.2 系统概要设计	225
7.3 系统详细设计	227

第1章 课程设计详细说明

1.1 课程设计目的

课程设计是课程教学中的一项重要内容，是完成教学计划、达到教学目标的重要环节，是教学计划中综合性较强的实践教学环节，它对帮助学生全面牢固地掌握课堂教学内容、培养学生的实践和实际动手能力、提高学生的综合素质具有很重要的意义。

“Visual C#语言课程设计”是“Visual C#语言程序设计”课程的一个重要的实践性教学环节，其目的是使我们：

(1) 进一步巩固和加深对“Visual C#语言程序设计”课程基本知识的理解和掌握，了解 Visual C#语言在项目开发中的应用。

(2) 综合运用“Visual C#语言程序设计”课程和“软件工程”理论，来分析和解决课程设计问题，进行课程设计的训练。

(3) 学习程序设计开发的一般方法，了解和掌握信息系统项目开发的过程及方式，培养正确的设计思想和分析问题、解决问题的能力，特别是项目设计能力。

(4) 通过对标准化、规范化文档的掌握并查阅有关技术资料等，培养项目设计开发能力，同时提倡团队精神。

通过本次实践活动，使学生能够熟练地使用 Visual C#语言，并能按照系统工程化的方法开发一般的管理信息系统项目。

1.2 课程设计对象

本课程设计适用于已学习了“Visual C#语言程序设计”的学生，包括在校的计算机专业和信息专业及相关专业学生，非计算机专业的工科学生或部分管理类学生。本书也可以作为软件培训班的教材或辅导材料，对于程序爱好者及初涉项目管理的人员也有一定的帮助。

1.3 课程设计内容说明

Visual C#的课程设计应在指导教师的帮助下完成，其具体步骤本节将详加介绍。

1. 选题

选题可分为指导教师选题和学生自己选题两种。教师选题可选择统一的题目，学生选题则应通过指导教师批准后方可进行。

1) 选题内容

选题要符合本课程的教学要求，要注意选题的完整性，要能进行分析建模、设计、编

程、复审、测试等一系列工作，并能以规范的文档形式表现出来。

2) 选题要求

(1) 注意选题内容的先进性、综合性、实践性，应适合实践教学和启发创新，选题内容不应过于简单，难度要适中。

(2) 结合企事业单位应用的实际情况进行选题。

(3) 题目成果应具有相对完整的功能。

2. 拟出具体的设计方案

学生应在指导教师的指导下着手进行项目的总体方案总结与论证。学生可根据自己所接受的设计题目设计出具体的实施方案，报指导教师批准后开始实施。

3. 程序的设计与调试

学生在指导教师的指导下应完成所接受题目的项目开发工作，编程和上机调试，最后得出预期的成果。

4. 撰写课程设计总结报告

课程设计总结报告是课程设计工作的整理和总结，主要包括需求分析、总体设计、详细设计、复审、编码、测试等部分，最后写出课程设计的总结报告。

1.4 课程设计要求

1.4.1 文档要求

课程设计总结报告的撰写规范应参照 CMM 模型编写，最终以课程设计实验报告的形式上交归档。

课程设计的总结报告是在完成设计、编程、调试后，对学生归纳技术文档、撰写科学技术总结报告能力的训练，以培养学生严谨的作风和科学的态度。通过撰写课程设计总结报告，不仅可以把分析、设计、安装、调试及技术参考等内容进行全面总结，而且还可以把实践内容提升到理论高度。总结报告按如下内容和顺序用 A4 纸进行打印(撰写)并装订成册。

(1) 统一的封面(含课程设计课题名称、专业、班级、姓名、学号、指导教师等)。

(2) 设计任务和技术要求(由指导教师 in 选题时提供给学生)。

(3) 内容摘要。

(4) 目录。

(5) 课程设计总结报告正文，正文可按章节来撰写，应包含的内容如下所述。

① 项目需求分析(方案的可行性分析、方案的论证等内容)。

② 项目总体设计(系统的总体结构设计等内容)。

③ 项目详细设计(各模块或单元程序的设计、算法原理阐述、完整的程序框图)。

④ 项目复审(对分析、总体设计和详细设计进行详细的复审)。

⑤ 编码(根据某一程序设计语言对设计结果进行编码的程序清单)。

⑥ 项目测试(使用程序调试的方法和技巧排除故障;选用合理的测试用例进行程序的系统测试和数据误差分析等)。

⑦ 总结(本课题核心内容程序清单及使用价值、程序设计的特点和方案的优缺点、改进方法和意见)。

(6) 以统一格式列出主要参考文献。

1.4.2 开发要求

1. 程序开发过程要求

软件系统的开发是按阶段进行的,一般可划分为以下阶段:可行性分析;需求分析;系统设计(概要设计、详细设计);程序开发;编码,单元测试;系统测试;系统维护。

软件开发过程中要明确各阶段的工作目标、实现该目标所必需的工作内容以及达到的标准。只有在上一个阶段的工作完成后,才能开始下一阶段的工作。

1) 可行性分析

明确系统的目的、功能和要求,了解目前所具备的开发环境和条件,论证的内容叙述如下。

- (1) 在技术能力上是否可以支持。
- (2) 在经济上效益如何。
- (3) 在法律上是否符合要求。
- (4) 与部门、企业的经营和发展是否吻合。
- (5) 系统投入运行后的维护有无保障。

可行性讨论的目的是判定软件系统的开发有无价值。分析和讨论的内容可形成“项目开发计划书”,主要包括以下六项。

- (1) 开发的目的及所期待的效果。
- (2) 系统的基本设想,涉及的业务对象和范围。
- (3) 开发进度表,开发组织结构。
- (4) 开发、运行的费用。
- (5) 预期的系统效益。
- (6) 开发过程中可能遇到的问题及注意事项。

可行性研究报告是可行性分析阶段软件文档管理的标准化文档。

2) 系统需求分析

系统需求分析是软件系统开发中最重要的一個阶段,直接决定着系统的开发质量和成败,必须明确用户的要求和应用现场环境的特点,了解系统应具有哪些功能及数据的流程和数据之间的联系。需求分析应有用户参加,到使用现场进行调研学习,软件设计人员应虚心向技术人员和使用人员请教,共同讨论解决需求问题的方法,对调查结果进行分析,明确问题的所在。需求分析的内容编写要形成“需求分析规格说明书”。

软件需求规格说明作为分析结果,它是软件开发、软件验收和管理的依据。因此,必须特别重视,不能有一点错误或不当,否则将来可能要付出很大的代价。

3) 系统设计

可根据系统的规模分成概要设计和详细设计两个阶段。

概要设计包括以下九个方面。

- (1) 划分系统模块。
- (2) 每个模块的功能确定。
- (3) 用户使用界面概要设计。
- (4) 输入、输出数据的概要设计。
- (5) 报表概要设计。
- (6) 数据之间的联系、流程分析。
- (7) 文件和数据库表的逻辑设计。
- (8) 硬件、软件开发平台的确定。
- (9) 有规律数据的规范化及数据唯一性要求。

系统的详细设计是对系统概要设计的进一步具体化,其主要工作有以下四项。

- (1) 文件和数据库的物理设计。
- (2) 输入、输出记录的方案设计。
- (3) 对各子系统的处理方式和处理内容进行细化设计。

(4) 编制程序设计任务书。程序说明书通常包括程序规范、功能说明、程序结构图,通常用 HPIPO(Hierarchy Plus Input Process Output)图来描述。

系统详细设计阶段的规范化文档为软件系详细设计说明书。

4) 程序开发

根据程序设计任务书的要求,用计算机算法语言实现解题的步骤,主要工作包括以下四项。

- (1) 模块的理解和进一步划分。
- (2) 以模块为单位的逻辑设计,也就是模块内的流程图的编制。
- (3) 编写代码,用程序设计语言编制程序。
- (4) 进行模块内功能的测试、单元测试。

程序质量的要求包括以下五个方面。

- (1) 满足要求的确切功能。
- (2) 处理效率高。
- (3) 操作方便,用户界面友好。
- (4) 程序代码的可读性好,函数、变量标识符合规范。
- (5) 扩充性、维护性好。

降低程序的复杂性也是十分重要的。系统的复杂性由模块间的接口数来衡量,一般地讲, n 个模块的接口数的最大值为 $n(n-1)/2$;若是层次结构, n 个模块的接口数的最小值为 $n-1$ 。为使复杂性最小,对模块的划分设计常常采用层次结构。要注意,编制的程序或模块应容易理解、容易修改,模块应相互独立,对某一模块进行修改时,对其他模块的功能应不产生影响,模块间的联系要尽可能少。

5) 系统测试

测试是为了发现程序中的错误,对于设计的软件,出现错误是难免的。系统测试通常由经验丰富的设计人员设计测试方案和测试样品,并写出测试过程的详细报告。系统测试是在单元测试的基础上进行的,包括以下四个方面。

- (1) 测试方案的设计。

- (2) 进行测试。
- (3) 写出测试报告。
- (4) 用户对测试结果进行评价。

除非是测试一个小程序,否则一开始就把整个系统作为一个单独的实体来测试是不现实的。与开发过程类似,测试过程也必须分步骤进行,每个步骤在逻辑上是前一个步骤的继续。大型软件系统通常由若干个子系统组成,每个子系统又由许多模块组成。因此,大型软件系统的测试基本上由下述几个步骤组成。

- (1) 模块测试。
- (2) 子系统测试。
- (3) 系统测试。
- (4) 验收测试。

软件测试的方法常用黑盒法和白盒法。

6) 文档资料

文档包括开发过程中的所有技术资料以及用户所需的文档,软件系统的文档一般可分为系统文档和用户文档两类。用户文档主要描述系统功能和使用方法,并不考虑这些功能是怎样实现的;系统文档则描述系统设计、实现和测试等方面的内容。文档是影响软件可维护性、可用性的决定因素。不夸张地讲,系统编程人员的每一张纸片都要保留,所以文档的编制是软件开发过程中的一项重要工作。

系统文档包括:开发软件系统在计划、需求分析、设计、编制、调试、运行等阶段的有关文档。在对软件系统进行修改时,系统文档应同步更新,并注明修改者和修改日期,如有必要应注明修改原因,应切记过时的文档是无用的文档。

用户文档包括以下四个方面。

- (1) 系统功能描述。
 - (2) 安装文档,说明系统安装步骤以及系统的硬件配置方法。
 - (3) 用户使用手册,说明使用软件系统方法和要求,疑难问题解答。
 - (4) 参考手册,描述可以使用的所有系统设施,解释系统出错信息的含义及解决途径。
- #### 7) 系统的运行与维护

系统只有投入运行后,才能进一步对系统检验,发现潜在的问题,为了适应环境的变化和用户要求的改变,可能会对系统的功能、使用界面进行修改。要对每次发现的问题和修改内容建立系统维护文档,并使系统文档资料同步更新。

通过建立代码编写规范,形成开发小组编码约定,提高程序的可靠性、可读性、可修改性、可维护性、一致性,保证程序代码的质量,继承软件开发成果,充分利用资源。提高程序的可继承性,使开发人员之间的工作成果可以共享。

软件编码要遵循的原则如下所述。

- (1) 遵循开发流程,在设计的指导下进行代码编写。
- (2) 代码的编写以实现设计的功能和性能为目标,要求正确完成设计要求的功能,达到设计的性能。
- (3) 程序具有良好的程序结构,提高程序的封装性,减低程序的耦合程度。
- (4) 程序可读性强,易于理解;方便调试和测试,可测试性好。
- (5) 易于使用和维护;具有良好的修改性、扩充性;可重用性强,移植性好。

- (6) 占用资源少, 以低代价完成任务。
- (7) 在不降低程序的可读性的情况下, 尽量提高代码的执行效率。

2. 程序设计代码规范

在开发中保持良好的编码规范是十分重要的。以下 Visual C# 编码规范是一种被证明能明显改善代码可读性, 并有助于代码管理、分类的编码规范。采用这种编码规范, 能避免一些繁长前缀, 便于记忆变量的用途。在课程设计中建议使用以下这种编码规范。

1) 类型级单位的命名

(1) 类。

Visual C# 是完全面向对象的语言。以 Class 声明的类, 都必须以名词或名词短语命名, 体现类的作用。如:

```
class TestClass
```

当类是一个特性(Attribute)时, 以 Attribute 结尾, 当类是一个异常(Exception)时, 以 Exception 结尾: 如:

```
Class ColorSetException
Class CauseExceptionAttribute
```

当类只需有一个对象实例(全局对象, 如 Application 等), 必须以 Class 结尾, 如:

```
Class ScreenClass
Class SystemClass
```

当类只用于作为其他类的基类, 根据情况, 以 Base 结尾:

```
MustInherit Class IndicatorBase
```

如果定义的类是一个窗体, 那么名字的后面必须加后缀 Form, 如果是 Web 窗体, 必须加后缀 Page:

```
Class PrintForm : Inherits Form // Windows 窗体
Class StartPage : Inherits Page // Web 窗体
```

(2) 枚举和结构。

必须以名词或名词短语命名。最好体现枚举或结构的特点, 如:

```
Enum ColorButtons //以复数结尾, 表明这是一个枚举
Struct CustomerInfoRecord //以 Record 结尾, 表明这是一个结构体
```

(3) 委派类型。

委派声明就是定义一个封装特定参数类型和返回值类型的方法体(静态方法或实例方法)的数据类型。如:

普通的委派类型以描述动作的名词命名, 以体现委派类型实例的功能:

```
delegate sub dataseeker (byval seekstring as string)
```

用于事件处理的委派类型, 必须以 eventhandler 结尾, 如:

```
delegate sub datachangedeventhandler (byval sender as object, byval e as
datachangedeventargs)
```

(4) 接口。

接口不一定必须要由 I 作为前缀，但一般都在接口前面加上这个前缀以将它们区别于类。一般用形容词命名，突出表现实现接口的类将具有什么能力：

```
Interface ISortable
```

(5) 模块。

模块不是类型，它的名称一般除了必须以名词命名外，还必须加以后缀 Module：

```
Module SharedFunctionsModule
```

上述所有规则的共同特点是，每个组成名称的词语都必须是以大写字母开始，禁止使用完全大写或小写的名称。

2) 方法和属性的命名

(1) 方法。

无论是函数还是子程序，方法都必须以动词或动词短语命名。无需区分函数和子程序，也无需指明返回类型。

```
sub Open(String CommandString)
function SetCopyNumber(int CopyNumber)
```

参数需要指明 ByVal 还是 ByRef，这一点写起来会让程序变长，但非常必要。如果没有特殊情况，都使用 ByVal。参数的命名方法，参考后面“变量的命名方法”。需要重载的方法，需要加上 Overload 关键字，根据需要编写重载的方法。

(2) 属性。

原则上，字段(Field)是不能公开的，要访问字段的值，一般使用属性。属性以简单明了的名词命名：

```
Property Concentration As Single
Property Customer As CustomerTypes
```

(3) 事件。

事件是出现某种行为的自动通知，是类的成员，只能在事件处理上下文中使用。命名的原则一般是动词或动词的分词。

```
public Event MyEventHandler SomeEvent
event click as clickeventhandler
event colorchanged as colorchangedeventhangler
```

(4) 变量和常数。

常数以表明常数意义的名词命名，一般不区分常数的类型：

```
Const single DefaultConcentr = 0.01
```

在严格要求的代码中，常数以 c_ 开头，例如，c_DefaultConcentration，但最好不要用它，它会带来输入困难。

普通类型的变量，只要有有意义的名字命名即可，不可使用简称和无意义的名称诸如 A、x1 等，下面给出具体例子：

```
int Index;  
double NextMonthExpenditure;  
string CustomerName;
```

不能起太长的名字，应该尽量简洁，如下面的例子：

```
string VariableUsedToStoreSystemInformation; //错误，太复杂了  
string SystemInformation; //正确，简单明了  
string sysInfo; //错误，过于简单
```

特殊情况可以考虑一个字母的变量：

```
boolean g;
```

对于控件，应该指明控件的类型，方法是直接在变量后面加以类名：

```
Button NextPageButton; // 按钮  
MainMenu MyMenu; // 菜单
```

像这样的例子，还有很多，无需规定某种类型的变量的前缀，只需把类型写在前面就行了。

(5) 命名规范

① ADO.NET 命名规范，如表 1-1 所示。

表 1-1 ADO.NET 命名规范

数据类型	数据类型简写	标准命名举例
Connection	con	conNorthwind
Command	cmd	cmdReturnProducts
Parameter	parm	parmProductID
DataAdapter	dad	dadProducts
DataReader	dtr	dtrProducts
DataSet	dst	dstNorthWind
DataTable	dtbl	dtblProduct
DataRow	drow	drowRow98
DataColumn	dcol	dcolProductID
DataRelation	drel	drelMasterDetail
DataView	dvw	dvwFilteredProducts

② WinForm Control 命名规范，如表 1-2 所示。

表 1-2 WinFormControl 命名规范

数据类型	数据类型简写	标准命名举例
Label	Lbl	lblMessage
LinkLabel	Llbl	llblToday
Button	btn	btnSave
TextBox	txt	txtName

续表

数据类型	数据类型简写	标准命名举例
MainMenu	mmnu	mmnuFile
CheckBox	chk	chkStock
RadioButton	rbtn	rbtnSelected
GroupBox	gbx	gbxMain
PictureBox	pic	picImage
Panel	pnl	pnlBody
DataGrid	dgrd	dgrdView
ListBox	lst	lstProducts
CheckedListBox	clst	clstChecked
ComboBox	cbo	cboMenu
ListView	lvw	lvwBrowser
TreeView	tvw	tvwType
TabControl	tctl	tctlSelected
DateTimePicker	dtp	dtpStartDate
HScrollBar	hsb	hsbImage
VScrollBar	vsb	vsbImage
Timer	tmr	tmrCount
ImageList	ilst	ilstImage
ToolBar	tlb	tlbManage
StatusBar	stb	stbFootPrint
OpenFileDialog	odlg	odlgFile
SaveFileDialog	sdlg	sdlgSave
FoldBrowserDialog	fbdlg	fgdlgBrowser
FontDialog	fdlg	fdlgFoot
ColorDialog	cdlg	cdlgColor
PrintDialog	pdlg	pdlgPrint

③ WebControl 命名规范，如表 1-3 所示。

表 1-3 WebControl 命名规范

数据类型	数据类型简写	标准命名举例
AdRotator	adrt	Example
Button	btn	btnSubmit
Calendar	cal	calMettingDates
CheckBox	chk	chkBlue
CheckBoxList	chkl	chkIFavColors
CompareValidator	valc	valcValidAge
CustomValidator	valx	valxDBCheck

续表

数据类型	数据类型简写	标准命名举例
DataGrid	dgrd	dgrdTitles
DataList	dlst	dlstTitles
DropDownList	drop	dropCountries
HyperLink	lnk	lnkDetails
Image	img	imgAuntBetty
ImageButton	ibtn	ibtnSubmit
Label	lbl	lblResults
LinkButton	lbtn	lbtnSubmit
ListBox	lst	lstCountries
Panel	pnl	pnlForm2
Placeholder	plh	plhFormContents
RadioButton	rad	radFemale
RadioButtonList	radl	radlGender
RangeValidator	valg	valgAge
RegularExpression	vale	valeEmail Validator
Repeater	rpt	rptQueryResults
RequiredFieldValidator	valr	valrFirstName
Table	tbl	tblCountryCodes
TableCell	tblc	tblcGermany
TableRow	tblr	tblrCountry
TextBox	txt	txtFirstName
ValidationSummary	vals	valsFormErrors
XML	xmlc	xmlcTransformResults

④ 变量的作用域及前缀，如表 1-4 所示。

表 1-4 变量的作用域及前缀

前 缀	说 明	举 例
P	全局变量	pstrName
St	静态变量	ststrName
M	模块或者窗体的局部变量	MstrName
A	数组	AintCount[]

⑤ 变量数据类型的前缀，如表 1-5 所示。

表 1-5 变量数据类型的前缀

C#数据类型	类库数据类型	标准命名举例
Sbyte	System.sbyte	sbte
Short	System.Int16	sht
Int	System.Int32	int
Long	System.Int64	lng
Byte	System.Byte	bte

续表

C#数据类型	类库数据类型	标准命名举例
Ushort	System.UInt16	usht
UInt	System.UInt32	uint
Ulong	System.UInt64	ulng
Float	System.Single	flt
Double	System.Double	dbl
Decimal	System.Decimal	dcl
Bool	System.Boolean	bol
Char	System.Char	chr
Object	System.Object	obj
String	System.String	str
DateTime	System.DateTime	dte
IntPtr	System.IntPtr	intptr

⑥ 自定义对象。

规定应该根据自定义对象的名称来确定该对象类型的前缀，例子如下：

对象：SysSet

前缀：ss

例子：ssSafety

3) 标签

标签就是用于 Goto 跳转的代码标识，由于 Goto 并不推荐使用，所以标签的使用也比较苛刻。标签必须全部大写，中间的空格用下划线_代替，而且应该以 _ 开头，如：

_A_LABEL_EXAMPLE:

如此定义标签是为了与其他代码元素充分区别。

4) 名字空间

通常，一个工程使用一个名字空间，通常不需要用 Namespace 语句，而是在工程选项的“Root Namespace”中指定，使用根名字空间可以使代码更加整齐，容易修改，这一点是 Visual C# 最主要的优点。名字空间的语法是：

公司名.产品名[组件名的复数] 如：

Namespace COM.NET

Namespace COM.File.IO.Files

随便起一个名字空间的名字绝对不是一个好主意，一定要遵守上述规定。

5) 格式化

良好的格式化代码对我们的浏览与维护有相当的好处。

(1) 块。

.NET 提供了 #Region...#End Region 块控制。我们应该根据代码所实现的功能分类并以块组织起来。