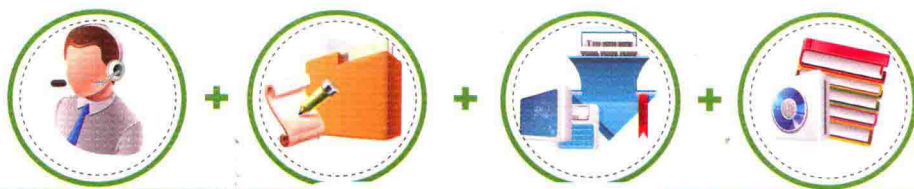


C#课程设计

案例精编（第2版）

杨 恒 主编



- ◆ 以基础理论—实用技术—实训为主线
- ◆ 按照教与学的实际需要取材谋篇
- ◆ 精心设置了“小型案例实训”，旨在培养学生的实践能力
- ◆ 配备丰富的免费教学资源——微视频电子教案、习题答案、案例素材

全国高等院校应用型创新规划教材·计算机系列

C#课程设计案例精编

(第2版)

杨 恒 主 编



清华大学出版社
北 京

内 容 简 介

本书详细介绍了 C#语言编程的相关知识。

在本书中,我们精选了两个游戏案例、五个信息系统案例和三个系统应用案例,按照软件开发和游戏开发的步骤,详细地阐述系统的开发过程。

本书适合作为大中专院校计算机专业课程的教材,也可供自学者参考使用。

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#课程设计案例精编/杨恒主编. —2 版. —北京:清华大学出版社, 2016

(全国高等院校应用型创新规划教材·计算机系列)

ISBN 978-7-302-43588-4

I. ①C… II. ①杨… III. ①C 语言—程序设计—高等学校—教学参考资料 IV. ①TP312

中国版本图书馆 CIP 数据核字(2016)第 082141 号

责任编辑:孟 攀

装帧设计:杨玉兰

责任校对:李玉萍

责任印制:杨 艳

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>, 010-62791865

印 装 者:北京鑫海金澳胶印有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:23.75 字 数:579 千字

版 次:2008 年 6 月第 1 版 2016 年 5 月第 2 版 印 次:2016 年 5 月第 1 次印刷

印 数:1~2500

定 价:48.00 元

产品编号:066504-01

再 版 前 言

C#编程语言是由微软公司专门为.NET 平台设计的语言,它可以使用程序员把工作迁移到.NET 上。这种迁移对于广大的程序员来说是比较容易的,因为 C#从 C、C++和 Java 发展而来,采用了这三种语言最优秀的特点,并加入了它自己的特性。

C#是事件驱动的、完全面向对象的可视化编程语言,我们可以使用集成开发环境来编写 C#程序,程序员可以方便快速地建立、运行、测试和调试 C#程序。所以它一经推出,马上就受到广大程序员的青睐。

《C#课程设计案例精编》是清华大学出版社高等院校课程设计案例精编系列教材之一,第 1 版出版至今,已受到读者广泛好评。第 2 版在保持了前一版风格的基础上,根据读者的反馈,对部分内容进行了更新和修订,以达到与时俱进、满足读者需求的目的。

本书内容丰富,案例经典,从实际应用角度出发,涵盖了游戏开发、文件操作、网络编程等范畴,共 12 章,涉及 10 个经典案例,由浅入深、从简单到复杂,详细介绍了使用 C#开发应用程序的方法。本书是一本实践性和应用性很强的 C#语言实用教材。

在基础知识中,我们使用两章的篇幅重点介绍了.NET 框架和 C#语言的基本语法结构,对于初学者来说非常实用,同时也涉及到本书其他章节中所用到的知识点,比如数据库的连接等。

在游戏开发中,介绍了俄罗斯方块、贪吃蛇游戏的设计和实现,告诉读者如何使用 C#语言进行绘图、如何响应键盘按键操作和播放音乐文件;在文件操作中,介绍了目前应用十分广泛的员工管理信息系统、房屋出租管理系统、仓库管理信息系统、研究生管理信息系统、图书馆管理信息系统、影院语音播报系统的设计与实现,在这些案例中包含了数据库的操作、Excel 文件的操作和 Windows 语音功能的调用;在网络编程中,结合物联网的相关内容,介绍了网站监控系统、PM2.5 模拟采集系统的设计与实现,这部分内容涵盖了网络协议、多线程和地图 API 等相关技术。

本书从实践性和应用性出发,所有案例程序的开发都遵循软件工程的方法,即采取分析→设计→编码→运行调试的路线,内容组织合理、分析详细、通俗易懂。

本书中,所有案例均需在 VS 开发环境中进行开发和调试,其中前 7 个案例均在 Visual Studio 2003 中调试通过,后 3 个案例均在 Visual Studio 2013 + Windows 7 中调试通过。此外,文件操作和网络编程部分案例需要安装微软 Office 办公软件中的 Access 数据库和 Excel 电子表格应用。

本书适合作为本科、高职高专院校计算机、机械、电子、自动化等专业的学生进行课程设计的参考教材,可供计算机专业编程人员参考使用,同时,也可以作为 C#语言开发人员和爱好者的参考读物。

第 1 版前言

C#是一种先进的、面向对象的语言，使用 C# 语言可以让开发人员快速地建立大范围的基于 MS 网络平台的应用，并且提供大量的开发工具和服务，帮助开发人员开发基于计算和通信的各种应用。由于 C#是一种面向对象的开发语言，所以 C#可以大范围地适用于高层商业应用和底层系统的开发。即使是通过简单的 C# 构造，也可以让各种组件方便地转变为基于 Web 的应用，并且能够通过 Internet 被各种系统或是其他开发语言所开发的应用调用。

本书精选了八个信息系统案例和两个游戏案例，按照开发信息系统和游戏的步骤详细阐述了系统的开发过程。这十个案例分别是员工管理信息系统、房屋出租管理系统、仓库管理信息系统、研究生管理信息系统、图书馆管理信息系统、宿舍管理信息系统、理财管理信息系统、IT 设备资产管理系统、俄罗斯方块游戏的编制和贪吃蛇游戏的编制。其中，房屋出租管理系统后台数据库采用 Microsoft SQL Server，其他系统后台数据库采用 Microsoft Access。Access 是 Office 系列软件中用来专门管理数据库的应用软件，是一种功能强大并且使用方便的关系型数据库管理系统，一般也称为关系型数据库管理软件。它可运行于各种 Microsoft Windows 系统环境中，由于它继承了 Windows 的特性，不仅易于使用，而且界面友好，如今在世界各地广泛流行。它不需要数据库管理者具有专业的程序设计水平，任何非专业的用户都可以用它来创建功能强大的数据库管理系统。

本书适合作为高等院校计算机、自动化、机械、电子等专业学生课程设计的指导书，也适合作为开发人员的参考书。

本书由段德亮、余健、张仁才等编著。参与编写的人员还包括张伟、陈喾、蔚辉、张坤、陈运来、田野、仇亚飞、刘广兴、王翠翠、代小华、王莹莹、韩忠明、张辰威。由于编者水平有限，加上时间仓促，书中难免有一些不足之处，欢迎同行和读者批评指正。

编 者

第 1 章 C#基础知识1	
1.1 Visual Studio .NET2	
1.1.1 什么是.NET2	
1.1.2 .NET 结构.....2	
1.2 初识 C#2	
1.2.1 什么是 C#2	
1.2.2 C#代码的结构.....3	
1.2.3 C#注释4	
1.2.4 标识符与关键字.....5	
1.3 C#基本类型6	
1.3.1 值类型.....7	
1.3.2 引用类型.....10	
1.3.3 类型转换.....14	
1.4 变量和常量.....17	
1.4.1 变量的定义.....17	
1.4.2 变量的命名.....17	
1.4.3 变量的类型.....18	
1.4.4 常量.....18	
1.5 运算符与表达式.....19	
1.5.1 运算符分类.....19	
1.5.2 算术运算符.....19	
1.5.3 关系运算符.....20	
1.5.4 赋值运算符.....22	
1.5.5 逻辑运算符.....23	
1.5.6 位运算符.....24	
1.5.7 其他运算符.....25	
1.5.8 运算符的优先级和结合性.....27	
第 2 章 C#程序设计29	
2.1 C#控制台应用程序30	
2.1.1 创建工程.....30	
2.1.2 修改代码.....31	
2.1.3 运行程序.....32	
2.2 C# Windows 应用程序.....33	
2.2.1 新建项目33	
2.2.2 添加新的窗口35	
2.2.3 添加菜单36	
2.3 SQL 入门.....37	
2.3.1 SQL 简介37	
2.3.2 SQL 的优点37	
2.3.3 从服务器资源管理器连接 数据库38	
2.4 连接数据库40	
2.4.1 .NET 中的连接对象40	
2.4.2 C#连接 Access40	
2.4.3 C#连接 SQL Server41	
2.4.4 C#连接 Oracle.....42	
2.4.5 C#连接 MySQL43	
第 3 章 俄罗斯方块游戏的编制45	
3.1 程序概述46	
3.1.1 游戏的功能46	
3.1.2 游戏的预览46	
3.2 游戏的概要设计48	
3.2.1 游戏实现方案48	
3.2.2 游戏逻辑设计48	
3.3 游戏的详细设计及编码48	
3.3.1 主界面设计编码48	
3.3.2 游戏控制设置设计编码56	
3.3.3 游戏方块设计编码58	
3.3.4 游戏声音设计编码73	
本章小结74	
第 4 章 贪吃蛇游戏的编制75	
4.1 程序概述76	
4.1.1 游戏的功能76	
4.1.2 游戏的预览76	
4.2 游戏的概要设计77	
4.2.1 游戏实现方案77	

4.2.2 游戏逻辑设计.....	77	5.7.2 浏览员工月收入信息.....	117
4.3 游戏的详细设计及编码.....	77	5.7.3 修改员工月收入信息.....	119
4.3.1 主界面设计编码.....	77	5.7.4 删除员工月收入信息.....	120
4.3.2 游戏颜色设置设计编码.....	82	本章小结.....	121
4.3.3 游戏蛇设计编码.....	85		
本章小结.....	89	第6章 房屋出租管理系统	123
第5章 员工管理信息系统	91	6.1 系统概述.....	124
5.1 系统概述.....	92	6.1.1 系统的应用背景.....	124
5.1.1 系统功能与应用背景.....	92	6.1.2 系统的功能.....	124
5.1.2 系统预览.....	92	6.1.3 系统的预览.....	124
5.2 系统设计.....	94	6.2 系统概要设计.....	128
5.2.1 系统设计思想.....	94	6.2.1 系统实现方案和系统模块划分.....	128
5.2.2 系统结构设计.....	95	6.2.2 数据库逻辑设计.....	130
5.3 数据库设计.....	95	6.3 系统详细设计.....	134
5.3.1 数据库需求分析.....	95	6.3.1 数据库连接.....	134
5.3.2 数据库概念结构设计.....	95	6.3.2 出租人信息管理.....	135
5.3.3 数据库逻辑结构设计.....	97	6.3.3 房屋信息管理.....	135
5.3.4 设置表与表之间的关系.....	98	6.3.4 房屋查询.....	136
5.4 工种种类设置.....	99	6.3.5 承租者入住管理.....	137
5.4.1 添加工种种类.....	99	6.3.6 承租者查询.....	138
5.4.2 浏览工种种类.....	101	6.3.7 利润信息.....	138
5.4.3 修改工种种类.....	102	6.4 系统编制.....	139
5.4.4 删除工种种类.....	103	6.4.1 主界面的编码.....	139
5.5 员工个人信息管理.....	104	6.4.2 出租人信息管理部分的编码.....	148
5.5.1 添加员工信息.....	104	6.4.3 房屋信息管理部分的编码.....	153
5.5.2 浏览员工信息.....	106	6.4.4 房屋查询部分的编码.....	155
5.5.3 修改员工信息.....	108	6.4.5 承租者入住部分的编码.....	159
5.5.4 删除员工信息.....	110	6.4.6 承租者查询部分的编码.....	160
5.6 员工所属部门信息管理.....	111	6.4.7 利润信息部分的编码.....	161
5.6.1 添加部门信息.....	111	本章小结.....	161
5.6.2 浏览部门信息.....	112		
5.6.3 修改部门信息.....	113	第7章 仓库管理信息系统	163
5.6.4 删除部门信息.....	115	7.1 系统概述.....	164
5.7 员工月收入信息管理.....	116	7.1.1 系统功能与应用背景.....	164
5.7.1 添加员工月收入信息.....	116		

7.1.2 系统预览.....	164	8.2.3 数据库设计.....	201
7.2 系统设计.....	165	8.3 系统详细设计.....	204
7.2.1 系统设计思想.....	165	8.3.1 数据库连接.....	204
7.2.2 系统功能模块设计.....	165	8.3.2 主界面.....	204
7.2.3 数据库设计.....	167	8.3.3 系统管理.....	205
7.3 登录界面与用户模块设计.....	169	8.3.4 专业管理.....	206
7.3.1 登录界面设计.....	169	8.3.5 课程管理.....	207
7.3.2 用户模块设计.....	170	8.3.6 研究生管理.....	209
7.3.3 系统模块设计.....	171	8.3.7 成绩管理.....	210
7.4 物资信息管理.....	174	8.3.8 用户管理.....	212
7.4.1 添加物资信息.....	174	8.4 系统程序设计.....	212
7.4.2 浏览物资信息.....	175	8.4.1 登录界面编码.....	212
7.4.3 修改物资信息.....	177	8.4.2 主界面编码.....	214
7.4.4 查询物资信息.....	178	8.4.3 系统管理编码.....	217
7.5 入库信息管理.....	180	8.4.4 专业管理编码.....	219
7.5.1 添加入库信息.....	180	8.4.5 课程管理编码.....	222
7.5.2 浏览入库信息.....	182	8.4.6 研究生管理编码.....	226
7.5.3 修改入库信息.....	183	8.4.7 成绩管理编码.....	227
7.5.4 查询入库信息.....	185	8.4.8 用户管理编码.....	231
7.6 出库信息管理.....	187	本章小结.....	232
7.6.1 添加出库信息.....	187	第9章 图书馆管理信息系统.....	233
7.6.2 浏览出库信息.....	189	9.1 系统概述.....	234
7.6.3 修改出库信息.....	190	9.1.1 系统功能.....	234
7.6.4 查询出库信息.....	192	9.1.2 系统预览.....	235
7.7 库存信息管理.....	194	9.2 系统概要设计.....	236
7.7.1 浏览库存信息.....	194	9.2.1 系统设计思想.....	236
7.7.2 查询库存信息.....	195	9.2.2 功能模块设计.....	237
本章小结.....	196	9.3 数据库设计.....	238
第8章 研究生管理信息系统.....	197	9.3.1 数据库概念设计.....	238
8.1 系统概述.....	198	9.3.2 数据库逻辑设计.....	239
8.1.1 系统功能.....	198	9.3.3 数据库表之间的关系.....	241
8.1.2 系统预览.....	198	9.4 系统详细设计.....	241
8.2 系统概要设计.....	199	9.4.1 数据库连接.....	241
8.2.1 功能模块设计.....	199	9.4.2 系统管理设计.....	242
8.2.2 文件架构设计.....	200	9.4.3 图书管理设计.....	243

9.4.4	读者管理设计	246
9.4.5	借还管理设计	248
9.4.6	查询管理设计	251
9.4.7	用户管理设计	252
9.5	系统程序设计	253
9.5.1	登录界面的编码	253
9.5.2	主界面编码	255
9.5.3	系统管理编码	259
9.5.4	图书管理编码	261
9.5.5	读者管理信息	263
9.5.6	借还管理信息	264
9.5.7	查询管理信息	269
9.5.8	用户管理信息	271
	本章小结	272
第 10 章 影院语音播报系统		
10.1	系统概述	274
10.1.1	系统功能	274
10.1.2	系统预览	274
10.2	系统概要设计	275
10.2.1	系统设计思想	275
10.2.2	功能模块设计	276
10.2.3	数据库设计	276
10.3	系统详细设计	277
10.3.1	数据库连接	277
10.3.2	系统登录设计	278
10.3.3	系统主界面设计	278
10.3.4	系统管理设计	279
10.3.5	场次管理设计	281
10.3.6	紧急播报设计	283
10.4	系统程序设计	284
10.4.1	登录界面编码	284
10.4.2	主界面编码	286
10.4.3	用户管理编码	295
10.4.4	修改密码编码	297
10.4.5	场次管理编码	298

10.4.6	播报设置编码	300
10.4.7	紧急播报编码	301
10.4.8	编码补充说明	301
	本章小结	304

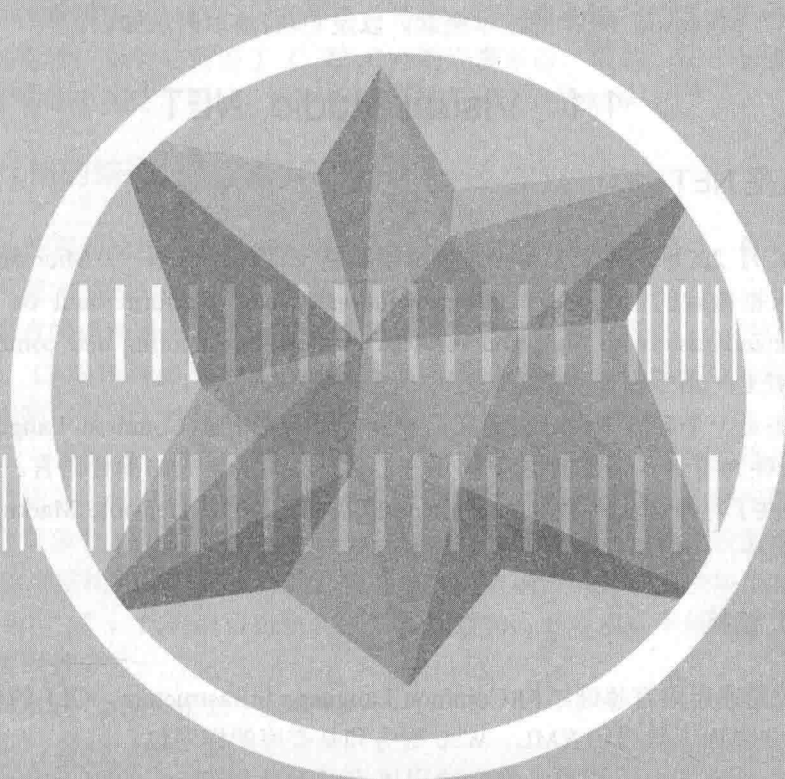
第 11 章 网站监控系统

11.1	系统概述	306
11.1.1	系统功能	306
11.1.2	系统预览	306
11.2	系统概要设计	307
11.2.1	系统设计思想	307
11.2.2	功能模块设计	308
11.2.3	数据库设计	308
11.3	系统详细设计	310
11.3.1	数据库连接	310
11.3.2	系统登录设计	311
11.3.3	系统主界面设计	312
11.3.4	系统管理设计	313
11.3.5	监控项目设计	314
11.3.6	报警配置设计	315
11.4	系统程序设计	316
11.4.1	登录界面编码	316
11.4.2	主界面编码	318
11.4.3	用户管理编码	328
11.4.4	修改密码编码	331
11.4.5	监控项目编码	332
11.4.6	报警配置编码	333
11.4.7	编码补充说明	335
	本章小结	335

第 12 章 PM2.5 模拟采集系统

12.1	系统概述	338
12.1.1	系统功能	338
12.1.2	系统预览	338
12.2	系统概要设计	340
12.2.1	系统设计思想	340

12.2.2	系统功能模块设计.....	340	12.4.1	Socket 调用编码	348
12.2.3	数据库设计	340	12.4.2	主服务端编码	350
12.3	系统详细设计	341	12.4.3	采集端口编码	356
12.3.1	数据库连接	341	12.4.4	设备管理编码	356
12.3.2	服务端界面设计	343	12.4.5	虚拟终端编码	360
12.3.3	系统管理设计	344	12.4.6	编码补充说明	364
12.3.4	设备管理设计	345	本章小结		364
12.3.5	虚拟终端设计	347			
12.4	系统程序设计	348			



第 1 章

C#基础知识



本章将学习 Microsoft .NET 的基本概念, 以及 C#的基本语法知识。

1.1 Visual Studio .NET

1.1.1 什么是.NET

2006 年 6 月 22 日, 微软公司正式推出了其新一代计算计划——Microsoft .NET(以下简称.NET)。根据微软的定义(.NET is a revolutionary new platform built on open Internet protocols and standards, with tools and services that meld computing and communications in new ways), .NET = 新平台 + 标准协议 + 统一开发工具。

.NET 首先是一个开放平台, 它定义了一种公用语言子集(Common Language Subset, CLS), 这是一种为符合其规范的语言和类库之间提供无缝集成的混合语言。.NET 统一的编程类库, 提供了对新一代网络通信标准、可扩展标记语言(Extensible Markup Language, XML)的完全支持, 使应用程序的开发变得更容易、更简单。

1.1.2 .NET 结构

.NET 结构是通用语言基础结构(Common Language Infrastructure, CLI)的微软实现, 加上几个支持用户界面、数据和 XML、Web 服务和基类库的程序包。

.NET 结构分为三个主要的子集: 通用语言运行环境(Common Language Runtime, CLR)、库和语言。图 1-1 给出了.NET 结构。



图 1-1 .NET 结构

1.2 初识 C#

1.2.1 什么是 C#

C#语言是一门简单、现代、优雅、面向对象、类型安全、平台独立的新型组件编程语言, 是微软公司为了能够利用.NET 平台优势而开发的一种新型编程语言。

C#语法风格源自 C/C++家族, 融合了 Visual Basic 的高效和 C/C++的强大, 是微软为奠定其互联网霸主地位而打造的 Microsoft .NET 平台的主流语言。一经推出, 便以其强大

的操作能力、优雅的语法风格、创新的语言特性、面向组件编程的支持等特色而深受世界各地程序员的喜爱。尽管它借鉴了 C 和 C++ 的许多东西，但是，在一些诸如名称空间、类、方法和异常处理等特定领域，它们之间还存在着巨大的差异。

1.2.2 C#代码的结构

C#代码的外观和操作方式与 C++ 和 Java 非常类似，在 C#编程中，使用的样式是比较清晰的，不用花太多的力气，就可以编写出可读性很强的代码。

与其他语言的编译器不同，无论代码中是否有空格、回车符或 Tab 字符(这些字符统称为空白字符)，C#编译器都不考虑这些字符。这样，在格式化代码时就有很大的自由度，但遵循某些规则将有助于使代码易于阅读。

C#代码由一系列语句组成，每个语句都用一个分号来结束。因为空格被忽略，所以一行可以有多个语句，但从可读性的角度来看，通常在分号的后面加上回车符，这样，就不能在一行上放置多个语句了。但一句代码放在多个行上是可以的(也比较常见)。

C#是一种块结构的语言，所有的语句都是代码块的一部分。这些块用花括号来界定，即使用“{”和“}”，代码块可以包含任意多行语句，或者根本不包含语句。注意，花括号字符不需要附带分号。

所以，简单的 C#代码块如下所示：

```
{  
    <code line 1, statement 1>;  
    <code line 2, statement 2>;  
    <code line 3, statement 2>;  
}
```

其中<code line x, statement y>部分作为 C#语句的占位符。

另外，在代码块中，我们还使用了缩进格式，使 C#代码的可读性更高，这是一个标准规则，实际上，在默认情况下，VS 会自动缩进代码。一般情况下，每个代码块都有自己的缩进级别，即它向右缩进了多少。

代码块可以互相嵌套(即块中可以包含其他块)，而被嵌套的块要缩进得多一些，如下面的示例：

```
{  
    <code line 1>;  
    {  
        <code line 2>;  
        <code line 3>;  
    }  
    <code line 4>;  
}
```

其中，第 3~6 行的代码就是被嵌套的块。

1.2.3 C#注释

在 C#代码中, 另一个常见的语句是注释。注释并不是严格意义上的 C#代码, 但代码最好有注释。注释就是解释, 即给代码添加描述性文本, 编译器会忽略这些内容。在处理比较长的代码段时, 注释可用于给正在进行的工作添加提示, 例如“这行代码要求用户输入一个数字”, 或“这段代码由 Bob 编写”。另外, 注释增加了代码的可读性。

C#添加注释的方式有两种——单行的和多行的。

1. 单行注释

单行注释一次只在一行注释。它们用双正斜线标记(//)开始。单行注释可以从给定行的任何位置开始, 并通过回车结束。例如, 下面示例中的第 1、4 和 7 行就是单行注释:

```
1. //主窗体被关闭的时候, 断开与数据库的连接
2. private void MainForm_Closed(object sender, System.EventArgs e)
3. {
4.     //判断与数据库的连接是否断开
5.     if(oleConnection1 != null)
6.     {
7.         //断开数据库的连接
8.         oleConnection1.Close();
9.     }
10. }
```

在 C#中, 还有第三类注释, 严格地说, 这是//语法的扩展。它们都是单行注释, 用三个“/”符号来开头, 而不是两个。例如:

```
/// A special comment
```

在正常情况下, 编译器会忽略它们, 就像其他注释一样; 但可以配置 VS, 在编译项目时, 提取这些注释后面的文本, 创建一个特殊格式的文本文件, 该文件可用于创建文档说明书。

2. 多行注释

多行注释在一组注释定界符内有一行或多行叙述。注释定界符由开始注释标记(/*)和结束注释标记(*/)组成。在这两个标记符中的所有内容都被看作是注释, 在编译器读取源代码时被忽略掉。下面的实例就是一个多行注释:

```
1. /*
2.  * 作者: 王霞
3.  * 开始时间: 6月2号
4.  * 功能: 建立数据库连接
5.  * */
```

但是, 需要注意的一点是, 在 C#中, 不允许出现多行注释的嵌套。例如, 下面的注释嵌套就是错误的:


```

1.  /*
2.   * 作者：王霞
3.   * 开始时间：6月2号
4.   * 功能：建立数据库连接
5.   *          /*第一次修改时间：6月3号
6.   *          第二次修改时间：6月6号
7.   *          第二次修改人：何亮
8.   *          */
9.   * */

```

这里，第1行的开始注释标记了多行注释，第5行的第二个开始注释标记作为注释中的一个字符而被忽略。在第8行的结束注释标记与第1行的开始注释标记相匹配，最后第9行的结束注释标记则会被编译器报告为语法错误，因为没有与它匹配的开始注释标记。

1.2.4 标识符与关键字

标识符是唯一的标识代码中的各种程序元素的名称，例如变量或字段等元素。它们由程序员指定，并且名称应该能够呈现标识的用途。

关键字是C#语言中的保留字。由于它们是保留的，所以不能把它们用作标识符。

1. 标识符

标识符是用于识别代码元素的名称。可以在程序中用作变量、对象、类、结构、枚举、类型、函数等的名字。

C#中，标识符的定义需要遵守以下几条规则。

- (1) 只能由大写字母、小写字母、下划线、数字(0~9)组成。
- (2) 必须以大写字母、小写字母或下划线开头。
- (3) 标识符大小写敏感，比如变量名 `f1` 和 `F1` 代表不同的变量。尽管如此，我们仍不建议仅利用大小写的不同来代表不同的标识符。大多数情况下，标识符应该望名知意。

标识符规则完全符合Unicode标准推荐的规则，但以下情况除外：

- 允许将下划线用作首字符。
- 允许在标识符中使用Unicode转义序列。
- 允许@字符作为前缀，以使关键字能够用作标识符。

下面是一些合法的C#标识符的例子：

```

oleConnection1
_command
@string

```

下面是一些无效的标识符的例子：

```

1tree
namespace

```

其中，`1tree` 无效是因为它的第一个字符是数字，标识符不允许以数字开始。标识符的第一个字符必须是字母或下划线。`namespace` 无效是因为它是保留字，不能用作标识符。

2. 关键字

关键字是对编译器具有特殊意义的预定义保留标识符。它们不能在程序中用作标识符,除非它们有一个@前缀。例如,@if 是一个合法的标识符,而 if 不是合法的标识符,因为它是关键字。C#的关键字清单如表 1-1 所示。

表 1-1 C#关键字

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

1.3 C#基本类型

应用程序总是要处理数据的,而现实世界中的数据类型多种多样,我们必须让计算机了解需要处理什么样的数据,以及采用哪种方式处理,按什么样的格式保存数据等。

任何一个完整的应用程序都可以看作是数据和作用于这些数据上的操作的说明。每一种高级语言都为开发人员提供了一组数据类型,不同的语言提供的数据类型不尽相同。

变量关系到数据的存储。实际上,可以把计算机内存中的变量视为架子上的盒子。在盒子中,可以放入一些东西,再把它们取出来,或者只是看看盒子里是否有东西。变量也是这样,数据可以放在变量中,可以从变量中取出数据或查看它们。

尽管计算机中的所有数据都是相同的东西(一组 0 和 1),但变量有不同的内涵,称为类型。下面仍使用盒子来类比,盒子有不同的形状和尺寸,某些东西只能放在特定的盒子中。建立这种类型系统的原因是,不同类型的数据需要用不同的方法来处理。变量限定为不同的类型,可以避免混淆它们。例如,组成数字图片的 0 和 1 序列与组成声音文件的 0 和 1 序列,其处理方式是不同的。

实际上,可以使用的变量类型是无限多的。其原因是可以自己定义类型,存储各种复杂的数据。

尽管如此,C#中有两种基本数据类型:值类型和引用类型。值类型直接存储它的数据内容,而引用类型存储的是对象的引用。这两种类型对变量的赋值有着不同的含义。

1.3.1 值类型

基于值类型的变量直接包含值。将一个值类型变量赋给另一个值类型变量时,将复制包含的值。这与引用类型变量的赋值不同,引用类型变量的赋值只复制对对象的引用,而不复制对象本身。

C#的值类型可以分为以下几种:

- 简单类型(Simple Types)。
- 结构类型(Struct Types)。
- 枚举类型(Enumeration Types)。

简单类型就是组成应用程序中基本组成部件的类型,例如数值类型和布尔类型。数值类型又进一步分为整数类型、实数类型和字符类型。

1. 布尔类型

布尔类型是用来表示“真”和“假”这两个概念的。布尔类型表示的逻辑变量只有两种取值: true 或 false。在 C 和 C++中,用 0 来表示“假”,其他任何非 0 的值表示“真”,但这种表示方法在 C#中是错误的。在 C#中, true 值不能被任何非 0 的值代替,在其他整数类型和布尔类型之间不存在任何转换。例如:

```
Bool x = true; //正确
Bool y = 1;  //错误
```

2. 整数类型

整数类型变量的值为整数。数学上的整数可以从负无穷大到正无穷大,但由于计算机的存储单元是有限的,所以计算机语言提供的整数类型有一个取值范围。根据该类型的变量在内存中占的位数,可以将 C#中的整数类型分为 8 种: 短字节型(sbyte)、字节型(byte)、短整型(short)、无符号短整型(ushort)、整型(int)、无符号整型(uint)、长整型(long)、无符号长整型(ulong)。一些变量名称前面的 u 是 unsigned 的缩写,表示不能在这