



高等学校计算机科学与技术教材

# C语言实训教程

COMPUTER Science and Technology

□ 韩莹 丰继林 单维锋 编著

- 原理与技术的完美结合
- 教学与科研的最新成果
- 语言精练，实例丰富
- 可操作性强，实用性突出



清华大学出版社

● 北京交通大学出版社

高等学校计算机科学与技术教材

# C 语言实训教程

韩莹 丰继林 单维锋 编著

清华大学出版社  
北京交通大学出版社

· 北京 ·

## 内 容 简 介

C 语言程序设计项目实习是 C 语言学习过程中的一个非常重要的环节,项目实习不仅帮助学生掌握 C 语言程序设计的基本知识,更重要的是培养学生掌握程序设计开发的基本素质、思维方法和技能,锻炼学生的程序设计综合能力,为今后较快地参与实际软件项目的开发奠定坚实的基础。书中精选了学生感兴趣的、难度适中的、很多高校实习用过的典型案例,教师可以根据学生的具体情况组织学生通过独立或分组讨论的方式进行实习。

本书共有 27 章,包含了 25 个实习项目,书中章节安排特点如下:第 1 章介绍了实训的要求及说明;第 2 章介绍了 VC 绘图 EasyX 库的使用方法;从第 3 章开始到第 27 章,每个项目作为一个章节,项目难度按章节递进。根据项目要求,数据多是基于文件操作的,在数据存储方式上既有选择数组的、也有选择链表的;游戏项目中的绘图操作,使用了由 EasyX 工作小组开发的,利用 VC 开发平台和 TC 绘图功能相结合而成的 EasyX 库,通过调用库中简单的函数实现绘图操作;所有项目的程序代码都是在 VC 环境下上机调试通过;各个项目主要包括了设计题目、设计目的、功能设计要求及项目详细设计,在详细设计小节中又主要包括项目功能模块图、项目流程图、项目数据结构、项目文件及项目主要函数的设计与实现。

这 25 个项目分别是:ATM 终端机模拟系统、学生宿舍管理系统、电子通讯录管理系统、万年历系统、学生选课系统、食堂饭卡管理系统、学生成绩管理系统、员工工资管理系统、图书馆借阅管理系统、飞机订票系统、实验室设备管理系统、小型的物业管理系统、小型超市管理系统、简单文本编辑器、简易四则运算计算器、俄罗斯方块、双人对战五子棋单机游戏、扫雷游戏、连连看游戏、打字母小游戏、贪吃蛇游戏、推箱子游戏、手动搜索迷宫游戏、基于词典的中文切词词典算法实现、一种基于二进制串的 Trie 索引树分词词典机制。

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 13501256678 13801310933

### 图书在版编目(CIP)数据

C 语言实训教程/韩莹,丰继林,单维锋编著. —北京:清华大学出版社;北京交通大学出版社,2013.1

(高等学校计算机科学与技术教材)

ISBN 978-7-5121-1351-0

I. ① C… II. ① 韩… III. ① C 语言-程序设计-高等学校-教材 IV. ① TP312

中国版本图书馆 CIP 数据核字(2013)第 011500 号

责任编辑:杨正泽

出版发行:清华大学出版社 邮编:100084 电话:010-62776969 <http://www.tup.com.cn>  
北京交通大学出版社 邮编:100044 电话:010-51686414 <http://press.bjtu.edu.cn>

印刷者:北京时代华都印刷有限公司

经 销:全国新华书店

开 本:185×260 印张:21 字数:598 千字

版 次:2013 年 1 月第 1 版 2013 年 1 月第 1 次印刷

书 号:ISBN 978-7-5121-1351-0/TP·722

印 数:1~3 000 册 定价:32.00 元

本书如有质量问题,请向北京交通大学出版社质监局反映。对您的意见和批评,我们表示欢迎和感谢。

投诉电话:010-51686043, 51686008; 传真:010-62225406; E-mail: [press@bjtu.edu.cn](mailto:press@bjtu.edu.cn)。

# 前 言

C语言是在国内外广泛使用的一种计算机语言。C语言功能丰富、表达能力强、使用灵活方便、应用面广、目标程序效率高、可移植性好，既具有高级语言的优点，又具有低级语言的许多特点。但是C语言牵涉的概念比较复杂，规则繁多，使用灵活，容易出错，不少初学者感到困难，要学好C程序设计不仅要掌握好理论知识，课程实训更不容忽视。通过上机实验，不仅可帮助学生消化和巩固课堂所学知识，也可激发和提高学生学习C程序设计的兴趣，更重要的是可培养学生的自学能力，锻炼学生的逻辑思维能力。C语言项目课程实训的主要目的是加强学生的编程能力，将理论知识与动手实践结合起来，锻炼自身的分析解决实际问题的能力，为后续各门计算机课程及其他语言的学习打下坚实的基础。

本书供刚学完C语言的大一学生课程实训使用。通过本次课程实训，对C语言程序设计进行一次全面的综合训练，加强学生的动手能力。最终目标是通过这种形式，帮助学生系统地掌握该门课程的主要内容，更好地完成教学任务，为学习后续课程和进行毕业设计做好准备。

本书有以下老师的参与帮助完成，王茂发、陈福明、李攀、杨秋格、潘志安等，谢谢他们的帮助和支持。

本书肯定有很多不足之处，希望得到广大读者的批评指正。

编 者

于2013年1月

# 目 录

<b>第 1 章 项目课程实训要求及说明</b> .....	1
1.1 项目教学目标 .....	1
1.2 项目课程实训总体要求 .....	2
1.3 项目课程实训过程及教学方式 .....	2
1.4 项目软件开发过程 .....	3
1.5 代码书写要求及编码原则 .....	3
1.6 项目课程实训设计报告内容要求及规范格式 .....	7
1.7 项目课程实训考核方式 .....	8
<b>第 2 章 VC6.0 简单绘图说明</b> .....	10
2.1 几个基本概念.....	11
2.2 函数说明.....	12
2.3 简单绘图.....	13
2.4 常用的绘图语句.....	13
2.5 结合流程控制语句来绘图.....	14
2.6 数学知识在绘图中的运用.....	16
2.7 实现简单动画.....	17
2.8 捕获按键,实现动画的简单控制.....	18
2.9 用函数简化相同图案的制作.....	20
2.10 绘图中的位运算 .....	22
2.11 用鼠标控制绘图 .....	25
2.12 随机函数 .....	27
2.13 数组 .....	29
2.14 加载图片 .....	32
2.15 通过位运算实现颜色的分离与处理 .....	34
2.16 窗体句柄 .....	35
2.17 设备上下文句柄 .....	36
<b>第 3 章 ATM 终端机模拟系统</b> .....	38
3.1 设计题目.....	38
3.2 功能设计要求.....	38
3.3 详细设计.....	40
3.3.1 数据结构.....	40
3.3.2 数据文件.....	40
3.3.3 功能模块图 .....	40

3.3.4	系统流程图	41
3.3.5	主要函数及算法的设计与现实	42
<b>第4章</b>	<b>学生宿舍管理系统</b>	<b>50</b>
4.1	设计题目	50
4.2	功能设计要求	50
4.3	详细设计	51
4.3.1	功能模块图	51
4.3.2	数据结构	51
4.3.3	数据文件	51
4.3.4	主要函数及算法的设计与现实	52
<b>第5章</b>	<b>电子通讯录管理系统</b>	<b>60</b>
5.1	设计题目	60
5.2	功能设计要求	60
5.3	详细设计	61
5.3.1	功能模块图	61
5.3.2	数据结构	61
5.3.3	数据文件	62
5.3.4	主要函数及算法的设计与现实	62
<b>第6章</b>	<b>万年历系统</b>	<b>68</b>
6.1	设计题目	68
6.2	功能设计要求	68
6.3	详细设计	68
6.3.1	功能模块图	68
6.3.2	主要函数及算法的设计与现实	69
<b>第7章</b>	<b>学生选课系统</b>	<b>74</b>
7.1	设计题目	74
7.2	功能设计要求	74
7.3	详细设计	75
7.3.1	功能模块图	75
7.3.2	数据结构	75
7.3.3	数据文件	76
7.3.4	主要函数及算法设计与实现	76
<b>第8章</b>	<b>食堂饭卡管理系统</b>	<b>87</b>
8.1	设计题目	87
8.2	功能设计要求	87
8.3	详细设计	88
8.3.1	功能模块图	88
8.3.2	数据结构	89
8.3.3	数据文件	89

8.3.4	主要函数及算法的设计与实现 .....	89
<b>第9章</b>	<b>学生成绩管理系统</b> .....	102
9.1	设计题目 .....	102
9.2	功能设计要求 .....	102
9.3	详细设计 .....	103
9.3.1	功能模块图 .....	103
9.3.2	数据结构 .....	103
9.3.3	数据文件 .....	104
9.3.4	系统流程图 .....	104
9.3.5	主要函数及算法的设计与实现 .....	104
<b>第10章</b>	<b>员工工资管理系统</b> .....	114
10.1	设计题目 .....	114
10.2	功能设计要求 .....	114
10.3	详细设计 .....	115
10.3.1	功能模块图 .....	115
10.3.2	数据结构 .....	115
10.3.3	数据文件 .....	116
10.3.4	主要函数及算法的设计与实现 .....	116
<b>第11章</b>	<b>图书馆借阅管理系统</b> .....	123
11.1	设计题目 .....	123
11.2	功能设计要求 .....	123
11.3	详细设计 .....	124
11.3.1	功能模块图 .....	124
11.3.2	数据结构 .....	125
11.3.3	数据文件 .....	126
11.3.4	系统流程图 .....	126
11.3.5	主要函数及算法的设计与实现 .....	127
<b>第12章</b>	<b>飞机订票系统</b> .....	139
12.1	设计题目 .....	139
12.2	功能设计要求 .....	139
12.3	详细设计 .....	140
12.3.1	功能模块图 .....	140
12.3.2	数据结构 .....	141
12.3.3	数据文件 .....	141
12.3.4	系统流程图 .....	142
12.3.5	主要函数及算法的设计与实现 .....	142
<b>第13章</b>	<b>实验室设备管理系统</b> .....	150
13.1	设计题目 .....	150
13.2	功能设计要求 .....	150

13.3	详细设计	151
13.3.1	功能模块图	151
13.3.2	数据结构	151
13.3.3	数据文件	152
13.3.4	主要函数及算法的设计与实现	152
<b>第14章</b>	<b>小型的物业管理系统</b>	<b>171</b>
14.1	设计题目	171
14.2	功能设计要求	171
14.3	详细设计	172
14.3.1	功能模块图	172
14.3.2	数据结构	172
14.3.3	数据文件	172
14.3.4	主要函数及算法的设计与实现	172
<b>第15章</b>	<b>小型超市管理系统</b>	<b>185</b>
15.1	设计题目	185
15.2	功能设计要求	185
15.3	详细设计	186
15.3.1	功能模块图	186
15.3.2	数据结构	186
15.3.3	主要文件	187
15.3.4	系统流程图	187
15.3.5	主要函数及算法的设计与实现	188
<b>第16章</b>	<b>简单文本编辑器</b>	<b>201</b>
16.1	设计题目	201
16.2	功能设计要求	201
16.3	详细设计	202
16.3.1	功能模块图	202
16.3.2	数据结构	202
16.3.3	主要函数及算法的设计与实现	203
<b>第17章</b>	<b>简易四则运算计算器</b>	<b>214</b>
17.1	设计题目	214
17.2	功能设计要求	214
17.3	详细设计	215
17.3.1	功能模块图	215
17.3.2	数据结构	215
17.3.3	系统流程图	216
17.3.4	主要函数及算法的设计与实现	216
<b>第18章</b>	<b>俄罗斯方块</b>	<b>227</b>
18.1	设计题目	227



18.2	功能设计要求	228
18.3	详细设计	228
18.3.1	功能模块图	228
18.3.2	数据结构	228
18.3.3	系统流程图	230
18.3.4	主要函数及算法的设计与实现	230
<b>第 19 章</b>	<b>双人对战五子棋单机游戏</b>	<b>240</b>
19.1	设计题目	240
19.2	功能设计要求	240
19.3	详细设计	241
19.3.1	功能模块图	241
19.3.2	数据结构	241
19.3.3	系统流程图	242
19.3.4	主要函数及算法的设计与实现	242
<b>第 20 章</b>	<b>扫雷游戏</b>	<b>248</b>
20.1	设计题目	248
20.2	功能设计要求	248
20.3	详细设计	249
20.3.1	功能模块图	249
20.3.2	数据结构	250
20.3.3	位图工作准备	251
20.3.4	系统流程图	251
20.3.5	主要函数及算法的设计与实现	252
<b>第 21 章</b>	<b>连连看游戏</b>	<b>261</b>
21.1	设计题目	261
21.2	功能设计要求	261
21.3	详细设计	262
21.3.1	功能模块图	262
21.3.2	数据结构	262
21.3.3	数据文件	263
21.3.4	位图工作准备	263
21.3.5	系统流程图	263
21.3.6	主要函数及算法设计	264
<b>第 22 章</b>	<b>打字母小游戏</b>	<b>272</b>
22.1	设计题目	272
22.2	功能设计要求	272
22.3	详细设计	273
22.3.1	功能模块图	273
22.3.2	数据文件	273

22.3.3	系统流程图	273
22.3.4	主要函数及算法的设计与实现	273
<b>第 23 章</b>	<b>贪吃蛇游戏</b>	<b>279</b>
23.1	设计题目	279
23.2	功能设计要求	279
23.3	详细设计	279
23.3.1	功能模块图	279
23.3.2	数据结构	280
23.3.3	系统流程图	281
23.3.4	主要函数及算法设计与实现	282
<b>第 24 章</b>	<b>推箱子游戏</b>	<b>286</b>
24.1	设计题目	286
24.2	功能设计要求	286
24.3	详细设计	287
24.3.1	功能模块图	287
24.3.2	数据结构	287
24.3.3	图片文件	287
24.3.4	系统任务流程图	288
24.3.5	主要函数及算法设计与实现	289
<b>第 25 章</b>	<b>手动搜索迷宫游戏</b>	<b>294</b>
25.1	设计题目	294
25.2	功能设计要求	294
25.3	详细设计	294
25.3.1	功能模块图	294
25.3.2	数据结构	295
25.3.3	系统流程图	296
25.3.4	主要函数及算法设计与实现	296
<b>第 26 章</b>	<b>基于词典的中文切词词典算法实现</b>	<b>302</b>
26.1	设计题目	302
26.2	设计目的	302
26.3	功能设计要求	303
26.4	详细设计	303
26.4.1	数据结构	303
26.4.2	数据文件	306
26.4.3	系统流程图	306
26.4.4	主要函数及算法的设计与实现	306
<b>第 27 章</b>	<b>一种基于二进制串的 Trie 索引树分词词典机制</b>	<b>314</b>
27.1	设计题目	314
27.2	设计目的	314

27.3 功能设计要求.....	315
27.4 详细设计.....	315
27.4.1 Trie 索引树词典机制建立.....	315
27.4.2 数据结构.....	316
27.4.3 数据文件.....	318
27.4.4 主要函数及算法的设计与实现.....	318
<b>参考文献</b> .....	<b>321</b>

# 第 1 章

## 项目课程实训要求及说明

C 语言程序设计是一门实践操作性很强的课程，要学好 C 程序设计不仅要掌握好理论知识，实验课程更不容忽视。通过上机实验，不仅可帮助学生消化和巩固课堂所学知识，也可激发和提高学生学习 C 程序设计的兴趣，更重要的是可培养学生的自学能力，锻炼学生的逻辑思维能力。C 语言项目课程实训的主要目的是加强自身的编程能力，将理论知识与动手实践结合起来，锻炼自身的分析解决实际问题的能力，为后续各门计算机课程及其他语言打下坚实的基础。

### 1.1 项目教学目标

- 进一步巩固和复习 C 语言程序设计的基础知识，加深对基础内容的理解和把握；
- 使学生更加系统地理解和掌握 C 语言的基本概念、语言特点和编程技巧；
- 通过对该系统的设计与实现使掌握结构体的定义、结构体数组的基本操作、函数的定义、数组作为函数参数的传递及文件的读写操作；
- 掌握链表的操作；
- 培养学生结构化程序、模块化程序设计的方法和能力；
- 学习和掌握 C 语言中的图形处理函数及其应用；
- 能正确分析现实生活中的问题，并抽象成数学模型，进行模块分析和编程；
- 掌握基本的分析问题和利用计算机求解问题的能力，具备初步的高级语言程序设计的能力；
- 在程序调试过程中，能根据运行环境给出的错误提示，正确解决程序中的语法错误；
- 在程序调试过程中，能根据运行结果，运用相应的手段，正确地找出并解决程序中的逻辑错误；
- 在实训过程中，适当进行小组分工，培养学生软件开发的能力及团队协调和团队合作的能力；
- 提交课程实训任务书，培养专业文档书写的能力；
- 为后续各门计算机课程的学习和毕业设计打下坚实的基础。

## 1.2 项目课程实训总体要求

学习《C 语言程序设计》课程必须重视实践环节，为了达到独立编辑、调试、运行程序的目的，在学完该课程后为学生安排 1~3 周的项目课程实训。项目课程实训过程中，以小组形式进行，一般为 1~5 人一组。根据所选项目首先进行算法分析，根据项目实现的功能进行数据分析，设计相应的数据结构，同时为贯彻“程序设计=算法+数据结构”的思想，要求学生完成流程图的设计，然后根据流程图写出课程实训代码，按时上机调试，由设计小组共同完成，设计的功能应相对完善，小组各成员全程参与程序构思、基本结构设计、变量设计、函数设计、文件操作等，并完成自己的设计任务。在设计中要综合运用所学内容，顺利调试通过并运行所编制的程序，最后完成课程实训报告中，报告书中包含设计的算法及部分程序代码。

## 1.3 项目课程实训过程及教学方式

在教学过程中，可以采取一看（先让学生看懂读懂范例程序），二模仿（模仿编写例题程序），三独立编程的循序渐进的教学方法。课程实训作为一个紧凑的教学阶段也可以使用基于行动导向教学范式中的协作-反思教学法来进行教学活动。在该教学法的教学过程中，学习者通过参加协商、决策、行动和检查的整个工作过程，提高解决问题的能力 and 交流协作的能力。项目综合课程实训时间安排提前两周布置课程实训任务，在课程实训的整个过程中，把学生分成若干个工作小组，培养学生的交流协作能力。教师不仅负责学习进度的监控及提供咨询，必要时还要提供学生工作小组学习所需要的信息，帮助小组成员自我约束并对信息进行深入分析，以促进小组成员工作能力的形成。项目课程实训任务简单说明如表 1-1 所示。

表 1-1 项目课程实训任务简单说明

序号	课程实训项目任务	主要内容	学时
1	需求分析	根据项目需求，描述模型	根据项目大小来定
2	系统分析	分析系统，建立模型	
3	系统设计	系统架构	
		结构设计	
		功能设计	
4	系统实现	硬件环境实现	
		软件环境实现	
		功能实现	

续表

序号	课程实训项目任务	主要内容		学时
5	系统测试	单元测试	设计测试用例, 对单个类及类方法进行测试	根据项目大小来定
		功能测试	设计测试用例, 对系统关键功能进行测试	
		性能测试	使用测试工具, 对系统进行性能测试	
6	系统部署	描述具体部署过程		
7	整理项目文档	根据自己完成的情况, 整理项目文档		

## 1.4 项目软件开发过程

编程是为了解决问题。有了良好的问题解决方案, 才有机会编出高质量的程序。掌握一定的软件开发方法非常重要。

① 明确问题的需求: 明确问题的需求必须将问题陈述清楚, 并明白解决问题需要什么。这一步的目标就是清除不重要的方面, 将注意力集中在根本的问题上。

② 分析问题: 分析问题涉及需要明确的问题为输入(要处理的数据)、输出(希望的结果)和解决方案的任何附加需求或约束条件。

③ 设计解决问题的算法: 设计解决问题的算法(处理流程)。

④ 实现算法: 实现算法, 将算法写成程序, 将算法的每一步转化为编程语言的一条或者多条语句。

⑤ 测试并验证整个程序: 测试整个程序来验证它能否按照预期工作。不能仅仅只依赖一次测试的结果, 而要利用不同的测试数据运行程序若干次来确定程序在算法中的每一种情况下均能正确工作。

⑥ 维护并更新程序: 通过修改程序来去除以前未检测到的错误。

## 1.5 代码书写要求及编码原则

在软件业高速发展、团队开发成为基本开发模式的今天, 几乎每个软件企业或软件开发团队都会制定各种各样的规范, 应用于软件开发过程的各个环节, 编码规范是最基础的规范之一。不同公司, 对同一种开发语言, 一般都采用类似的规范, 也就是说编码规范具有一些通用性的标准。

为保证开发团队的协作和后期修改能有效进行, 使学生遵照统一的排版风格、注释标准、命名规则及编码诸原则进行 C 语言编程, 特提出以下规范要求。

### 1. 缩进

程序块要采用缩进风格编写, 缩进以 4 个空格 (Space) 键为单位, 不使用“TAB 键”;

函数体、结构体、循环体以及分支结构中的语句行都须采用缩进风格；所有的 if、while、for、do 结构中的语句即使只有一行也须用括号括起来；if、while、for、do 语句单独占一行，左、右花括号也各占一行且不缩进。例如：

```
if (temp==1)
{
    printf("OK!");
}
```

## 2. 间隔

所有的标识符都必须被空白字符包围。

例如：

```
Int the_size=5;
if (the_size==5)
```

## 3. 空行

代码段之间须以一个空行进行间隔。

## 4. 空格位置

在一个关键字和左括号“(”之间。注意：不要在函数名和左括号之间加空格；在参数列表的每个逗号“,”之后；

二元操作符前后。注意：一元操作符前后都不加空格。例如：int a=10; a=a+1; a++; for 语句的每个表达式之间。例如：for (int i=0; i < 20; i++)

## 5. 代码行

不允许把多个短语句写在一行中，即一行只写一条语句；

长语句、参数、表达式（超过 80 个 ASCII 字符）应分行书写，操作符放在新行之首，划分出的新行要进行适当的缩进，使排版整齐，语句可读。

例如：

```
#001 perm_count_msg.head.len= NO7_TO_STAT_PERM_COUNT_LEN
#002 +STAT_SIZE_PER_FRAM ;
#003 act_task_table[STAT_TASK_CHECK_NUMBER+index].occupied
# 004 =stat_poi[index].occupied;
```

## 6. 命名规则

所有的标识符只能用字母（A~Z 或 a~z）和数字（0~9）。不得包含货币符号或其他非 ASCII 字符；

变量的命名应尽可能采用有意义的名词，力求可顾名思义；

可以采用一个单词或多个单词的缩写作为名字，缩写单词的每个字母都要大写；

对于难以使用英文的情况，可以参考相关行业标准，比如使用国标；

采用约定俗成的习惯用法。

常见的习惯用法：

循环变量: i、j、k、m、n

长度: length

数量: count

位置: pos 或 position

下标或索引: i 或 index

设置/获取: set/get

大小: size

## 7. 注释

为确保程序易读性, 注释语句不得小于全部代码行的 30%;

说明性文件(如头文件 .h 文件、.inc 文件、.def 文件、编译说明文件 .cfg 等)头部应进行注释, 注释必须列出: 版权说明、版本号、生成日期、作者、内容、功能、与其他文件的关系、修改日志等, 头文件的注释中还应包含函数功能简要说明;

源文件头部应进行注释, 列出: 版权说明、版本号、生成日期、作者、模块目的/功能、主要函数及其功能、修改日志等;

函数头部应进行注释, 列出: 函数的目的/功能、输入参数、输出参数、返回值、调用关系(函数、表)等;

边写代码边注释, 修改代码同时修改相应的注释, 以保证注释与代码的一致性, 不再有用的注释要删除;

注释的内容要清楚、明了, 含义准确, 防止注释二义性;

避免在注释中使用缩写, 特别是非常用缩写;

注释应与其描述的代码相近, 对代码的注释应放在其上方或右方(对单条语句的注释)相邻位置, 不可放在下面, 如放于上方则需与其上面的代码用空行隔开;

对于所有有物理含义的变量、常量, 如果其命名不是充分自注释的, 在声明时都必须加以注释, 说明其物理含义。变量、常量、宏的注释应放在其上方相邻位置或右方;

数据结构声明(包括数组、结构、类、枚举等), 如果其命名不是充分自注释的, 必须加以注释。对结构中的每个域的注释放在此域的右方;

全局变量要有较详细的注释, 包括对其功能、取值范围、哪些函数或过程存取它以及存取时注意事项等的说明;

注释与所描述内容进行同样的缩排;

将注释与其上面的代码用空行隔开;

对变量的定义和分支语句(条件分支、循环语句等)必须编写注释;

对于 switch 语句下的 case 语句, 如果因为特殊情况需要处理完一个 case 后进入下一个 case 处理, 必须在该 case 语句处理完、下一个 case 语句前加上明确的注释。

## 8. 编码原则

注意运算符的优先级, 并用括号明确表达式的操作顺序, 避免使用默认优先级;

避免使用不易理解的数字, 用有意义的标识来替代; 涉及物理状态或者含有物理意义的常量, 不应直接使用数字, 必须用有意义的枚举或宏来代替;

去掉没必要的公共变量;

仔细定义并明确公共变量的含义、作用、取值范围及公共变量间的关系;



明确公共变量与操作此公共变量的函数或过程的关系，如访问、修改及创建等；  
 当向公共变量传递数据时，要十分小心，防止赋予不合理的值或越界等现象发生；  
 防止局部变量与公共变量同名；

严禁使用未经初始化的变量；

对所调用函数的错误返回码要仔细、全面地处理；

明确函数功能，精确（而不是近似）地实现函数设计。

**【例 1-1】** 格式规范让代码更清晰。

```
void main( )
{
    int i,s=0,t=0;
    for(i=1;i<=100;i++)
    {
        if(i%2==0)
            s+=i;
        else t+=i;
    }
    printf("s=%d,t=%d",s,t);
}
```

**【例 1-2】** 命名，容易区别，注释容易维护。

```
#define PI 3.14159 /*宏名和结构类型大写*/
typedef struct{int age,score;}STU;
int get_score(int score[N]) /*变量和函数名第一个字母小写,其后每个单词第一个字母大写*/
{
    ...
    char passscore;
    int*p_name;
    long sum=0L; /*避免使用 0 和 1。*/
    ...
}
```

**【例 1-3】** 空行让程序更容易读。

```
#include <stdio.h>
# define M 100
/*.....
void fun1( )
{
    ...
}
/*.....*/
void main( )
```