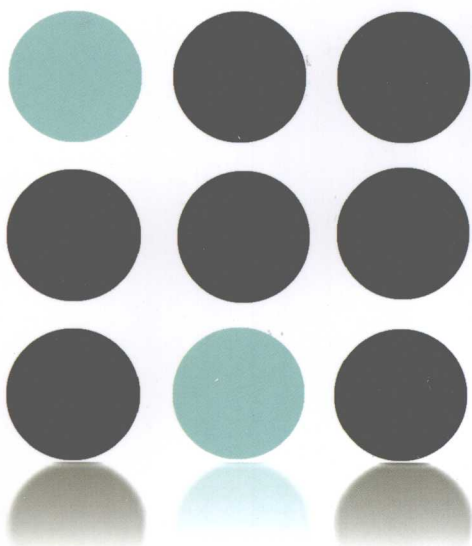


 高等院校计算机专业课程综合实验系列规划教材



丛书主编 何钦铭 陈根才

张 引 许端清 肖少拥

何钦铭 吕红兵 吴春明

李 彤 主审

著

C程序设计基础 课程设计



ZHEJIANG UNIVERSITY PRESS
浙江大学出版社

浙江大学魏绍相计算机教材建设基金资助
高等院校计算机专业课程综合实验系列规划教材

C 程序设计基础课程设计

张 引 等著

李 彤 主审

浙江大學出版社

内 容 提 要

本书在分析C语言高级编程技术的基础上,通过对“通讯录”、“计算器”、“俄罗斯方块”3个完整案例的解析,指导读者按照软件工程的生命周期开发过程,从问题分析、设计、编码到测试,循序渐进地完成大型程序的开发。书中提供了9个大型程序训练的题目及简要分析,供读者练习;为适合教学需要,还提供了可供参考的实施过程说明、开发文档模版、考核方法和评价标准,特别强调学生团队合作精神培养。本书还附光盘。光盘内容为:课件、书中案例的源程序、汉字字库、Turbo C 2.0的库函数使用手册、C语言上机操作指导。

图书在版编目(CIP)数据

C程序设计基础课程设计 / 张引等著. —杭州:浙江大学出版社, 2007.9

(高等院校计算机专业课程综合实验系列规划教材)

ISBN 978-7-308-05497-3

I. C… II. 张… 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134501号

C程序设计基础课程设计

张 引 许端清 肖少拥 何钦铭 吕红兵 吴春明 著
李 彤 主审

丛书主编 何钦铭 陈根才

策 划 黄娟琴 希 言

责任编辑 邹小宁 吴昌雷

封面设计 氧化光阴

出版发行 浙江大学出版社

(杭州天目山路148号 邮政编码310028)

(网址: <http://www.zjupress.com>

<http://www.press.zju.edu.cn>

<http://www.zjuxsj.com>)

电话:0571—88925592, 88273066(传真)

排 版 浙江大学出版社电脑排版中心

印 刷 临安市曙光印务有限公司

开 本 787mm×1092mm 1/16

印 张 16.25

字 数 416千

版 次 2007年9月第1版 2008年3月第2次印刷

印 数 3001—6000

书 号 ISBN 978-7-308-05497-3

定 价 26.00元(附光盘)

版权所有 翻印必究 印装差错 负责调换

高等院校计算机专业课程

综合实验系列规划教材编委会

主 任

齐治昌 国防科技大学教授,教育部软件工程专业教学指导分委员会副主任

副主任

陈道蓄 南京大学计算机系教授,教育部计算机科学与技术专业教学指导分委员会副主任

蒋宗礼 北京工业大学计算机学院副院长,教授,教育部计算机科学与技术专业教学指导分委员会秘书长

委 员(按姓氏笔画排列)

王志英 国防科技大学计算机学院副院长,教授,教育部高等学校计算机科学与技术专业教学指导分委员会副主任

左保河 华南理工大学软件学院副教授,教育部高等学校软件工程专业教学指导分委员会委员

刘 强 清华大学副教授,教育部高等学校软件工程专业教学指导分委员会秘书长

孙吉贵 吉林大学计算机学院副院长,教授,教育部高等学校计算机科学与技术专业教学指导分委员会委员

庄越挺 浙江大学计算机学院副院长,教授,教育部计算机科学与技术专业教学指导分委员会委员

吴 跃 电子科技大学计算机学院教授,教育部计算机科学与技术专业教学指导分委员会委员

李 彤 云南大学软件学院副院长,教授,教育部计算机科学与技术专业教学指导分委员会委员

邹逢兴 国防科学技术大学教授,国家级教学名师

陈志刚 中南大学信息学院副院长,教授,教育部高等学校计算机科学与技术专业教学指导分委员会委员

岳丽华 中国科学技术大学教授,教育部高等学校计算机科学与技术专业教学指导分委员会委员

徐宝文 东南大学教授,教育部高等学校软件工程专业教学指导分委员会委员

廖明宏 哈尔滨工业大学计算机学院副院长,教授,教育部高等学校计算机科学与技术专业教学指导分委员会委员

管会生 兰州大学信息科学与工程学院副院长,教授,教育部高等学校理工类计算机基础课程教学指导分委员会秘书长

前言

结构化程序设计和面向对象程序设计是程序设计的两种典型的思想和方法。目前高校开设的程序设计课程也基本据此划分为两大系列。虽然面向对象程序设计语言已成为现阶段广泛使用的一类语言,甚至 C++ 从语法扩展的角度被称作“更好的 C”,但是 C 语言仍以其支持结构化的设计和固有的语言特色——兼具高级语言和低级语言的功能、丰富灵活的控制和数据结构、简洁而高效的语句表达、清晰的程序结构和良好的可移植性,保持着旺盛的生命力,广泛应用于系统软件和应用软件的开发中。

同其他计算机专业课程一样,C 语言程序设计是一门实践性很强的课程。学习者要达到灵活运用,真正掌握应用程序设计语言解决实际问题的能力,必须经历初、中、高三个实践阶段。初级阶段:初学者通过大量小型程序(几行至几十行代码)的编程训练,掌握基本的语法和良好的书写风格,培养程序设计和编码的基本能力,初步理解结构化程序设计的思想和方法。中级阶段:学习者通过大型程序(几百行至千行以上代码)的编程训练,进一步培养程序设计能力和进行初步的团队合作的训练,并在软件工程生命周期开发方法的指导下,深入理解和真正掌握结构化程序设计的思想和方法,以及数据结构和算法的基本设计能力。高级阶段:用 C 开发实际的软件项目,在实践中不断学习、理解、掌握和应用软件工程的开发方法和技术。

目前,在国内程序设计入门课程的教学实践中,大都以初级阶段的训练为主,缺乏综合性的大型程序设计训练,相关教材奇缺。从而,许多学生缺少大型程序分析、设计、测试和团队合作等方面的实际锻炼,影响了后继课程的深入学习,甚至到毕业时都缺乏良好的程序设计基本能力。浙江大学计算机专业自 1997 年起就在 C 程序设计课程中实施大型编程实践,经过 10 年的课程建设,已经形成了一套完整的课程设计实践体系。该课程还被评为 2004 年国家精品课程。这本书正是我们多年教学实践经验的总结。

本教材构思新颖,实例丰富。通过若干完整案例的解析,指导读者按照软件工程的生命周期开发过程,从问题分析、设计、编码到测试,循序渐进地完成大型程序的开发。本教材针对教学需要,提供了可参考的实施过程说明、开发文档模版、考核方法和评价标准,在实施过程中特别强调学生团队合作精神的培养。教材还提供了一系列大型程序训练的题目,供读者练习。

全书共分 5 章。第 1 章通过一个简单的“扫雷”游戏软件的开发,详细讲解基于软件工程的大型程序开发方法;第 2 章介绍编写大型程序可能会涉及的硬件基础知识;第 3 章重点介绍利用 Turbo C 2.0 实现的一些高级编程技术,包括文本和图形的显示、键盘和鼠标的操作控制、图形绘制、动画生成、乐曲演奏、汉字显示、图像显现和精确的时间控制等;第 4 章通过 3 个大型程序范例的解析,引导读者回顾和领略大型程序的设计思想和开发方法,并在程

序的实现过程中体验高级编程技术的应用;第 5 章为读者提供了一系列大型程序训练的题目,每道题不仅给出了问题的描述、难点分析、主要数据结构及主要算法提示,而且给出了基本要求和高级要求,读者可以根据实际情况进行不同层次要求的训练。本书还附光盘,光盘内容为:课件、书中案例的源程序、汉字字库、Turbo C 2.0 的库函数使用手册、C 语言上机操作指导。

本教材第 1,2,3 章由张引编写,第 4 章由何钦铭、肖少拥、吕红兵编写,第 5 章由许端清、吴春明编写。何钦铭交叉评审了第 2,4,5 章,许端清交叉评审了第 3 章,肖少拥交叉评审了第 1 章。全书由张引统稿。

本教材适用于对 C 程序设计语言及其编程有初步基础的读者学习大型程序的开发方法。

由于作者水平所限,对书中存在的谬误之处,敬请读者指正。

作 者

2007 年 6 月

序 言

近 10 多年来,以计算机和通信技术为代表的信息技术迅猛发展,并已深入渗透到国民经济与社会发展的各个领域。信息技术成为国家产业结构调整 and 推动国民经济与社会快速发展的最重要的支撑技术。与此同时,深入掌握计算机专业知识、具有良好系统设计与分析能力的计算机高级专业人才在社会上深受欢迎。

计算机科学与技术是一门实践性很强的学科。良好的系统设计和分析能力的培养需要通过长期、系统的训练(包括理论和实践两方面)才能获得。高等学校的实践教学一般包括课程实验、综合性设计(课程设计)、课外科技活动、社会实践、毕业设计等,基本上可以分为三个层次:第一,是紧扣课堂教学内容,以掌握和巩固课程教学内容为主的课程实验和综合性设计;第二,是以社会体验和科学研究体验为主的社会实践和课外科技活动;第三,是以综合应用专业知识和全面检验专业知识应用能力的毕业设计。课程实践(含课程实验和课程设计)是大学教育中最重要也最基础的实践环节,直接影响后继课程的学习以及后继实践的质量。由于课程设计是以培养学生的系统设计与分析能力为目标,通过团队式合作、研究式分析、工程化设计完成较大型系统或软件的设计题目的,因此课程设计不仅有利于学生巩固、提高和融合所学的专业课程知识,更重要的是能够培养学生多方面的能力,如综合设计能力、动手能力、文献检索能力、团队合作能力、工程化能力、研究性学习能力、创新能力等。

浙江大学计算机学院在专业课程中实施课程设计(project)已有 10 多年的历史,积累了丰富的经验和资料。为全面总结专业课程设计建设的经验,推广建设成果,我们特别组织相关课程的骨干任课教师编写了这套综合实验系列教材。本系列教材的作者们不仅具有丰富的教学和科研经验,而且是浙江大学计算机学院和软件学院的教学核心力量。这支队伍目前已经获得了两门国家精品课程以及四门省部级精品课程,出版了几十部教材。

本套教材由《C 程序设计基础课程设计》、《软件工程课程设计》、《数据结构课程设计》、《数值分析课程设计》、《编译原理课程设计》、《逻辑与计算机设计基础实验与课程设计》、《操作系统课程设计》、《数据库课程设计》、《Java 程序设计课程设计》、《面向对象程序设计课程设计》、《计算机组成课程设计》、《计算机体系结构课程设计》和《计算机图形学课程设计》等十三门课程的综合实验教材所组成。该系列教材构思新颖、案例丰富,许多案例直接取材于作者多年教学、科研以及企业工程经验的积累,适用于作为计算机以及相关专业课程设计的实验教材;也适用于对计算机有浓厚兴趣的专业人士进一步提升计算机的系统设计与分析

能力。从实践的角度出发,大部分教材配备了随书光盘,以方便读者练习。

可以说,本套教材涵盖了计算机专业绝大部分必修课程和部分选修课程,是一套比较完整的专业课程设计系列教材,也是国内第一套由研究型大学计算机学院独立组织编写的专业课程设计系列教材。鉴于书中难免存在的谬误之处,敬请读者指正,以便不断完善。

主编 何钦铭、陈根才

2007年6月于求是园

目 录

第 1 章 大型程序的开发方法	1
1.1 软件工程的生命周期方法学	1
1.2 大型程序的开发流程	2
1.2.1 问题定义	3
1.2.2 程序分析	3
1.2.3 程序设计	4
1.2.4 编 码	17
1.2.5 测试和调试	30
1.2.6 运行与维护	35
1.2.7 分工协作	35
1.3 大型程序设计综合实验的实施	37
1.3.1 过程及要求	37
1.3.2 评价标准	39
第 2 章 高级编程的硬件基础	41
2.1 计算机硬件组成	41
2.2 80X86 微机系统	42
2.2.1 主存储器结构	43
2.2.2 寄存器与伪变量	44
2.2.3 实地址寻址方式	46
2.2.4 I/O 接口的寻址方式	47
2.3 Turbo C 编译方式简介	48
2.3.1 Turbo C 的内存模式	48
2.3.2 指针的分类	49
2.4 BIOS 与 DOS 调用	51
2.4.1 中断	51
2.4.2 BIOS 调用	52
2.4.3 DOS 调用	54
2.5 多文件的程序	57
2.6 Turbo C 2.0 库函数介绍	58

第 3 章 高级编程技术的实现	60
3.1 操作手段	60
3.1.1 键 盘	60
3.1.2 鼠 标	64
3.2 显示方式	67
3.2.1 显示器和显卡	67
3.2.2 文本显示方式	69
3.2.3 图形显示方式	80
3.3 图形绘制	88
3.3.1 画笔定位函数	88
3.3.2 点和直线的画法	89
3.3.3 矩形、圆和弧线的画法	93
3.3.4 封闭图形的填色函数及有关画图函数	98
3.4 图像技术	103
3.4.1 屏幕图像	103
3.4.2 图像文件	108
3.5 动画技术	115
3.5.1 简单的实现方法	115
3.5.2 利用动态开辟图视口的方法	118
3.5.3 利用屏幕图像存储再放的方法	119
3.5.4 利用页交替的方法	120
3.6 发声技术	122
3.6.1 计算机发声原理	122
3.6.2 声音函数	123
3.6.3 计算机乐谱	124
3.7 汉字显示技术	126
3.7.1 汉字编码	127
3.7.2 汉字显示	128
3.8 精确的时间控制技术	134
3.8.1 编写自己的中断程序	134
3.8.2 精确计时实现方案	136
第 4 章 大型综合程序范例解析	140
4.1 通讯录	140
4.1.1 问题描述	140
4.1.2 功能分析	140
4.1.3 设 计	141
4.1.4 源程序	144

4.1.5 测 试	168
4.2 简单计算器	169
4.2.1 问题描述	169
4.2.2 功能分析	169
4.2.3 设 计	170
4.2.4 源程序	173
4.2.5 测 试	193
4.3 俄罗斯方块	194
4.3.1 问题描述	194
4.3.2 功能分析	194
4.3.3 设 计	195
4.3.4 源程序	203
4.3.5 测 试	219
第5章 综合程序练习题	221
5.1 打字练习	221
5.1.1 问题描述	221
5.1.2 设计及实现要点	222
5.2 电子琴	223
5.2.1 问题描述	223
5.2.2 设计及实现要点	224
5.3 简易画图板	227
5.3.1 问题描述	227
5.3.2 设计及实现要点	228
5.4 贪吃蛇	229
5.4.1 问题描述	229
5.4.2 设计及实现要点	230
5.5 五子棋	232
5.5.1 问题描述	232
5.5.2 设计及实现要点	232
5.6 搬运工	235
5.6.1 问题描述	235
5.6.2 设计及实现要点	235
5.7 大富翁	238
5.7.1 问题描述	238
5.7.2 设计及实现要点	238
5.8 赛 车	240
5.8.1 问题描述	240
5.8.2 设计及实现要点	241

5.9 坦克大战	243
5.9.1 问题描述	243
5.9.2 设计及实现要点	243
参考文献	246

本章从软件工程的角度系统地讲述了大型程序的开发应遵循的开发流程及开发方法。通过本章的学习,读者应认识到编码只是软件(程序)生命周期的一个阶段,前期的分析和设计对解决问题而言是非常重要的,后期的测试和调试则进一步保证了软件(程序)的质量与可靠性。本章重点讲述的自顶向下逐步求精的结构化程序设计方法和良好的程序设计编码风格,必须通过长期的实践和经验积累才能真正掌握,它们也是一个优秀的软件开发人员应该具备的能力。这样,无论是采用何种程序设计语言,都能够写出结构清晰、易读易懂的好程序。因此,本章的学习将为今后的开发奠定良好的方法论基础。

1.1 软件工程的生命周期方法学

软件开发历史上的诸多惨痛教训使人们逐渐认识到,软件不等于源代码,它有自己的“生命周期”,大型软件系统的开发与其他工程项目如建造桥梁、制造飞机、轮船等的开发是同理的,必须有计划、分层次进行。软件工程正是随着软件的发展而诞生的一门学科,以提高质量、降低成本为目的。它视软件的开发为一项工程,借鉴传统工程的原则和方法,将正确的管理方法和当前能够得到的最好的开发技术结合起来。

一个软件从定义、开发、使用和维护,直到最终被废弃,要经历一个漫长的时期,这就如同一个人要经历胎儿、儿童、青年、中年、老年,直到最终死亡的漫长时期一样。通常把软件经历的这个漫长的时期称为生命周期。软件工程的传统解决途径强调使用生命周期方法学和各种结构分析及结构设计技术。

人类在解决复杂问题时普遍采用的一个策略就是“各个击破”,也就是对问题进行分解然后再分别解决各个子问题。软件工程采用的生命周期方法学就是从时间角度对软件开发和维护的复杂问题进行分解,把软件生命的漫长周期依次划分为若干个阶段,每个阶段有相对独立的任务,然后逐步完成每个阶段的任务。前一个阶段任务的完成是进行后一个阶段工作的前提和基础,而后一阶段任务的完成通常是使前一阶段提出的解法更进一步具体化,加进了更多的实现细节。在每一个阶段结束之前都必须进行正式严格的技术审查和管理复审,若检查通不过,则必须进行必要的返工,返工后还要再经过审查。审查的一个主要标准就是每个阶段都应该提交与所开发的软件完全一致的高质量文档资料,从而保证在软件开发工程结束时有一个完整准确的软件配置交付使用。文档不仅是前后阶段的通信工具,而且是软件交付使用后进行维护的依据。所以,采用生命周期方法学或称传统的软件工程方法,就使软件开发工程的全过程以一种有条不紊的方式进行,保证了软件的质量,特别是

提高了软件的可维护性。如果软件的开发按生命周期方法学,各阶段严格按时间顺序排列,其过程可以用如图 1.1 所示的瀑布模型来模拟。

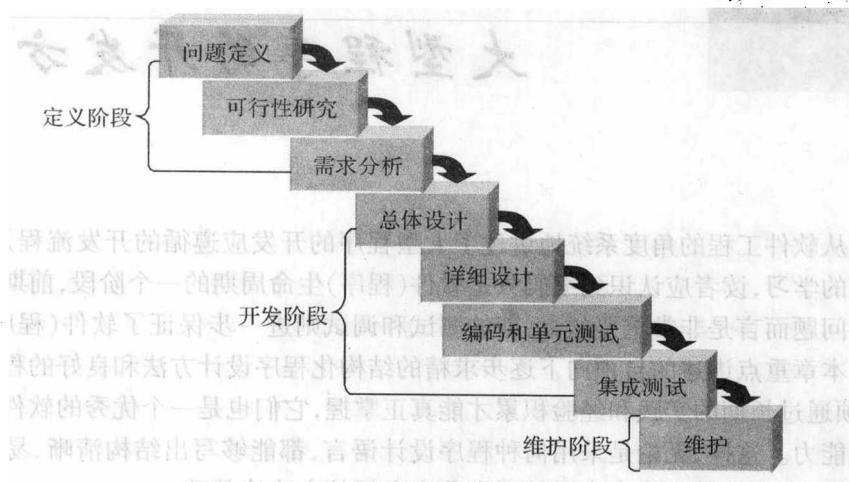


图 1.1 瀑布模型

一般来说,软件生命周期由软件定义、软件开发和软件维护三个时期组成,每个时期又可进一步划分成若干个阶段。

软件定义时期要回答“做什么”的问题,任务是确定软件开发工程必须完成的总目标、确定工程的可行性、导出实现工程目标应该采用的策略及系统必须完成的功能、估计完成工程需要的资源和成本,并且制定工程进度表。这个时期的工作通常又称为系统分析。通常进一步划分成三个阶段,即问题定义、可行性研究和需求分析。

软件开发时期要回答“怎么做”的问题,任务是具体设计和实现在前一个时期定义的软件,它通常由下述四个阶段组成:总体设计、详细设计、编码和单元测试、集成测试。其中前两个阶段又称为系统设计,后两个阶段又称为系统实现。

软件维护时期的主要任务是使软件持久地满足用户的需要。具体地说,当软件在使用过程中发现错误时应该加以改正;当环境改变时应该修改软件以适应新的环境;当用户有新要求时应该及时改进软件以满足用户的新需要。每一次维护活动本质上都是一次压缩和简化了的定义和开发过程。

在下一节,针对用 C 语言开发一个大型综合程序这一主题,我们将在软件工程瀑布模型的基础上详细介绍开发大型程序的方法和流程。

1.2 大型程序的开发流程

根据传统的生命周期方法学,我们知道编码只是软件开发的一个很小的阶段,而且是处在实现阶段。对于 C 语言的初学者,由于没有正式接受系统化开发方法的指导,往往会形成一个错误的认识:程序的开发就是编码。也就是说,拿到问题后,马上就开始写程序。这种做法的不良后果初学者无法体会到,因为他们所面临的需要解决的问题,无论从规模,还

是从难易程度而言,实在是太小了。所以在直接编写程序的过程中,大脑已经让初学者无意识地完成了问题的定义和设计过程。但是,这种侥幸的“个体化”做法对于复杂的现实问题的解决,即软件项目的开发是绝对行不通的。相对初学阶段,我们实际上已经可以解决较为复杂的问题了,也就是说已进入大型程序的开发阶段。因此,必须从现在开始,树立正确的开发观,为今后专业化开发打好基础。

遵循瀑布模型的开发流程,大型综合程序的开发要经历问题定义、分析、设计、编码、测试和维护几个阶段。

1.2.1 问题定义

问题定义阶段是在整个过程中占用时间最少的阶段,在这个步骤中我们的任务是明确要解决的问题是什么。如果不知道问题是什么就试图解决这个问题,显然是盲目的,只会浪费时间和精力,结果是毫无意义的。

在大型综合程序训练时期,欲解决的问题可由教师提供,或者由学生自行选题。假若是后者,那么学生必须动动脑筋,寻找身边有哪些事情可用计算机解决,然后确定一个可行的题目。例如,某位读者非常熟悉 Windows 系统中自带的扫雷游戏,于是他以此为题,自己设计实现一个类似功能的扫雷游戏。

1.2.2 程序分析

这个阶段的任务仍然不是具体地解决问题,而是理解问题和分析问题,确定“为了解决这个问题,目标系统必须做什么”,主要是确定目标系统必须具备哪些功能。除此之外,还要确定可能的输入或输出数据是什么。

我们在问题定义步骤中得到的问题,有时仅仅是一个抽象的题目,有时除题目外还附一段简要说明。无论问题以何种形式出现,都需要做进一步的分析,以获得系统必须实现哪些功能。下面我们以扫雷游戏为例,看看如何分析问题。这可通过两步来完成:首先必须对 Windows 系统中的扫雷游戏进行了解,然后确定我们将实现的扫雷系统要做什么。

通过亲身体验或其他交流手段,可获知这个游戏是在屏幕显示的一个雷区范围内(如图 1.2(a)所示),系统预先埋设了一定数目的地雷(图 1.2(a)中笑脸左侧小窗显示的数字),游戏者在游戏过程中通过判断,若能正确标记出雷区中的所有地雷,则游戏胜利(如图 1.2(b)所示);否则踩雷,游戏失败(如图 1.2(c)所示)。具体的游戏规则如下:

(1) 在“游戏”菜单上,单击“开局”,出现图 1.2(a)所示的游戏界面,其中包括地雷计数器窗口、计时器窗口和雷区。开局后,单击雷区中的任何一个方块,便启动计时器。每标记一个地雷,地雷计数器减 1。

(2) 单击某个方块,可挖开它。若所揭方块下有雷,则踩雷,此时所有含地雷的方块都标记 * (如图 1.2(c)所示),这局游戏失败;如果方块上出现数字,它代表在它周围的八个方块中共有多少颗地雷。

(3) 右击某个方块,则标记此方块下埋着地雷(实际上可能是误标),显示为■。每标记一个地雷,地雷计数器减 1。

(4) 右击某个方块两次,则在某块上面标一个问号(?),意味着没有把握判定它是否有雷。标记为“?”的块可在恰当的时候再右击两次或单击左键,将其标记为地雷或挖开。

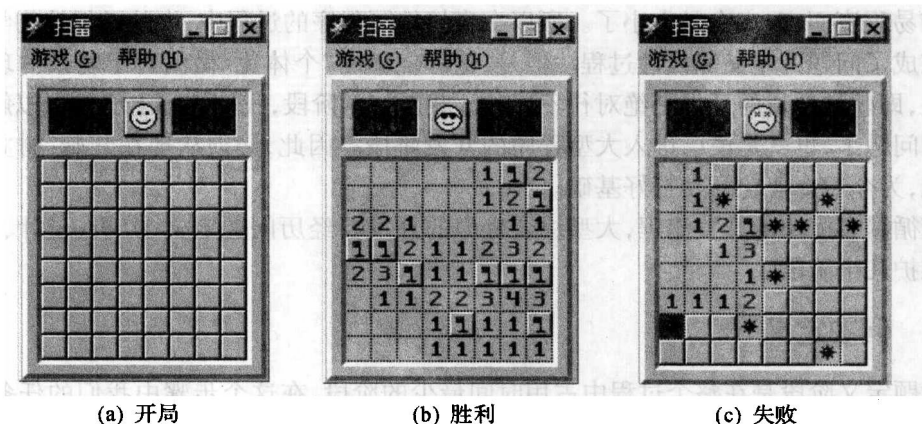


图 1.2 Windows 扫雷游戏

(5) 如果某个数字方块周围的地雷全都标记完,可以同时单击鼠标左右键,将其剩下的方块挖开。如果挨着这个方块的地雷没有全部标记完,则未挖开的方块将闪烁。

通过分析上述游戏规则,抛开 Windows 扫雷游戏中界面显示、鼠标操作等非本质的东西,我们从 Windows 扫雷游戏中提取的主要功能包括:

规则(1)隐含的初始化新游戏的开局功能。

规则(2)隐含的挖雷功能。

规则(3)对应的标记地雷功能。

规则(4)对应的标记疑问功能。

规则(5)的自动挖开功能。

另外,我们还可以看到游戏的输入是启动这些功能的指示信息,通过鼠标左右键的操作来表达;游戏的输出是游戏是否成功。

我们所要实现的扫雷游戏正是 Windows 扫雷游戏的模仿版,也就是说能够实现 Windows 扫雷游戏的主要功能,因此,我们的目标系统必须具备:开局功能、挖雷功能、标记地雷功能、标记疑问功能和自动挖开功能。到这里,程序分析的工作就结束了。接下来,将进入下一阶段——程序设计的工作。

1.2.3 程序设计

著名计算机科学家 Nikiklaus Wirth 提出了一个公式:程序 = 算法 + 数据结构,这一公式反映了程序的两个要素:算法和数据结构,实际上,程序还应当有另外两个要素:程序设计方法和语言工具。这四个方面是一个程序设计人员所应具备的知识。算法是灵魂,数据结构是加工对象,语言是工具,而程序设计需要好的设计方法。

经过程序分析阶段的工作,综合程序必须“做什么”已经清楚了,现在是决定“怎样做”的时候了,也就是到了设计算法和重要数据的数据结构的时候了。程序这个阶段的设计工作,应该对要解决的问题设计出具体的解决方案,得出对目标系统的精确描述,从而在编码阶段可以把这个描述直接翻译成用 C 语言书写的程序。

程序设计将采用结构化程序设计方法,自顶向下逐步求精地设计出综合程序的实现“蓝图”。为此,我们首先介绍结构化程序设计方法,然后列举描述算法的常用工具,接着仍以扫

雷游戏为例,详细说明设计阶段的工作和结构化程序设计方法的应用,最后谈谈团队合作情况下如何分工协作的问题。

1. 结构化程序设计方法

结构化程序设计的概念最早由 E. W. Dijkstra 提出。在软件发展的早期,即 20 世纪 60 年代末到 70 年代初,虽然科技在高速发展,程序规模越来越大,但是当时的编程技术却停留在手工作业的方式:设计各自为政、滥用 GOTO 语句造成的“意大利面条式”程序效率低下、可读性差、无章可循、错误百出、调试困难。在这种局面下,Dijkstra 提出结构化程序设计的理论。经过多年的实践,结构化程序设计的理论和实践日益完善,成为现代程序设计的主流方法之一。

那么,什么是结构化程序设计呢?结构化程序设计是一种设计程序的技术,采用自顶向下逐步求精的设计方法和单入口单出口的顺序、选择和循环三种基本控制结构。它提出的原则可归纳为 32 字:“自顶向下,逐步细化;清晰第一,效率第二;书写规范,缩进格式;基本结构,组合而成。”

自顶向下逐步求精的设计方法符合人类解决复杂问题的普遍规律,这种设计方法的过程是将问题求解由抽象逐步具体化的过程。例如,设计房屋就采用了这种方法,先进行整体规划,然后确定建筑物方案,再进行各部分的设计,最后进行门窗、楼道等的细节设计。在制定问题的解决方法时同样也能采用这种方法,例如,考虑如下的问题描述:

某学院为准备参加网络工程师认证的人员开办了相应的培训课程。当首批培训人员参加了认证考试后,该学院自然想知道这些学生考得如何,以便调整教学方案和收费标准。现在需要编写一个统计考试结果的程序,能够显示出通过和没有通过的人数,并且如果通过人数不低于总人数的 70%,显示提高学费的建议。已知参加考试的人员有 500 名,并且已经在通过者的名字旁边标注了 1,在未通过者的名字旁标注了 2。

仔细读完上述问题后,可分析出如下的结论:

(1) 程序必须要处理 500 个应试者的考试结果,所以要使用循环。

(2) 每个考试结果都是一个数,即 1 或 2。程序每次读取一个结果后,必须确定该数是 1 还是 2。

(3) 使用两个计数器。一个用于计算通过考试的人数,另一个用于计算没有通过的人数。

(4) 程序处理完所有结果后,必须判断是否有 70% 以上的人员通过了认证考试。

我们用自顶向下逐步求精的方法解决这个问题。问题的顶层表示为:

分析认证考试的结果并决定培训费是否提高

顶层是程序功能的完整的表述。第一次求精结果为:

初始化变量

输入 500 个考试结果,统计通过和未通过的人数

打印统计结果,并决定培训费是否提高

虽然上述表示了完整的程序,但是还需要进一步求精。对具体的变量来说,程序中需要用两个计数器 `passes` 和 `failures` 分别记录通过和没有通过认证考试的人数,用一个计数器 `counter` 控制循环过程,用一个变量 `result` 来存储当前输入的考试结果。这样,对“初始化变量”可做如下的求精: