



“十二五”普通高等教育项目化规划教材
高等教育课程改革项目研究成果

C 语言程序设计项目教程

——项目驱动下的工资管理系统开发及应用

C YUYAN CHENGXU SHEJI XIANGMU JIAOCHENG
XIANGMU QUDONGXIA DE GONGZI GUANLI
XITONG KAIFA JI YINGYONG



主编 李震平 李晓霞
主审 李仁发



北京邮电大学出版社
www.buptpress.com



“十二五”普通高等教育项目化规划教材
高等教育课程改革项目研究成果

C 语言程序设计项目教程

——项目驱动下的工资管理系统开发及应用

主 编	李震平	李晓霞	
副主编	杨 丽	程慧芳	韩晓鸿
编 委	魏红君	李 燕	刘 群
	马丽艳	刘传志	常志葵
	李 鹰		
主 审	李仁发		

北京邮电大学出版社
• 北京 •

内 容 简 介

本书以 CDIO 为理念,以项目驱动为主线,通过一个典型项目——工资管理系统带动 C 语言所有知识点,采用了真正的项目驱动方式,展现了全新的教与学的方法。

本书共十章,前九章包含了软件开发基础、C 语言程序概述、结构化程序的三大结构、数组、结构体数据类型、函数、指针、文件操作和编译预处理,第十章对 C 语言所有知识点进行了分类汇总,以体现知识点之间的联系,利于整体学习和把握 C 语言。本书除了通过职工工资管理系统带动 C 语言知识点外,还提供了内容丰富的、趣味性较强的实用案例,同时配套出版的《C 语言程序设计实验实训教程》也保持了同样的风格。

本书既可以作为面向工程类非计算机专业本科学生的计算机语言教材,也可以作为高等学校计算机专业本科和专科学生的基础教材,以及自学和教师参考资料。

图书在版编目(CIP)数据

C 语言程序设计项目教程:项目驱动下的工资管理系统开发及应用/李震平,李晓霞主编.

-- 北京:北京邮电大学出版社,2015.1(2016.12 重印)

ISBN 978-7-5635-4275-8

I. ①C… II. ①李… ②李… III. ①C 语言—程序设计—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2014)第 304421 号

书 名 C 语言程序设计项目教程——项目驱动下的工资管理系统开发及应用
主 编 李震平 李晓霞
责任编辑 向 蕾
出版发行 北京邮电大学出版社
社 址 北京市海淀区西土城路 10 号(100876)
电话传真 010-82333010 62282185(发行部) 010-82333009 62283578(传真)
网 址 www3.buptpress.com
电子信箱 ctrd@buptpress.com
经 销 各地新华书店
印 刷 北京泽宇印刷有限公司
开 本 787 mm×1 092 mm 1/16
印 张 21.5
字 数 537 千字
版 次 2015 年 1 月第 1 版 2016 年 12 月第 4 次印刷

ISBN 978-7-5635-4275-8

定价: 39.00 元

如有质量问题请与发行部联系

版权所有 侵权必究

自 20 世纪 80 年代中期 C 语言引入我国以来,已被许多高等学校列为程序设计课程的首选语言。尤其是随着我国经济的快速发展和科学技术的不断进步,工业化大力发展,自动化程度越来越高,社会对科技人员的计算机技术技能要求也越来越高,C 语言既是重要的计算机技术基础语言,也是工业自动化控制的重要开发语言,因此,学习和掌握 C 语言的重要性更加突出。

近年来,“亚太大学生机器人大赛”、“飞思卡尔杯全国大学生智能车竞赛”等面向在校大学生的各类电子对抗竞赛,从程序控制角度看,对 C 语言的驾驭能力是决定能否取胜的关键。

C 语言程序设计是一门实践性很强的课程,学习者必须在实践中才能逐步理解和掌握程序设计的思想和方法,而在传统的 C 语言程序设计教材中,往往采用“概念—解释—举例”的三部曲,这种方式既不利于培养学生的程序设计能力和语言应用能力,又不利于学生将来适应社会的需要。

本教材依据 CDIO 工程教育模式的理念,结合作者多年的 C 语言教学经验和软件开发实践,对 C 语言知识点的编排进行了精心的策划和组织,直接从应用入手,通过虚拟项目工资管理系统的实现,带动了绝大部分的 C 语言知识点:C 语言程序结构、数据类型、变量作用域、结构化程序三大结构、结构体、数组、指针、函数、文件和编译预处理等。因此,本书有鲜明的项目特色,更有利于学生在实践中领会所学知识,解决实际问题。

1. 项目驱动。本教材真正实现了项目驱动知识点的组织和编排。通过对工资管理系统的学习与开发,将分散的知识点进行有机联系,真正达到学以致用目的,读者可以通过学习、临摹、借鉴,最终将所学知识用于开发其他实用的小软件。

2. 实用性强。附录一给出了开发一个软件项目的整体参考流程:文档资料、结构框图和流程图,以及采用结构体数据类型组织工资记录的工资管理系统源程序,方便学习者学习和借鉴。附录二给出了采用双向链表组织工资记录的工资管理系统的源程序。附录三给出了 ASCII 字符编码。附录四给出了 C 语言常用标准库函数,方便查阅。更加重要的是第十章给出了 C 语言全面的知识点汇总,将分散却相关的知识点,有机地汇总于此章,更加适合初学者和有一定基础的读者。

3. 趣味性强。本教材的所有案例都是经过精心选择与设计的,除一些经典算法外,在趣味性上也做了足够的工作,彻底改变以往学习计算机语言的生硬与冷漠,极大提高了读者的学习兴趣和读者的好奇心。

4. 符合认识规律。根据知识点的特点,或采用“实践—理论—实践”,或采用“理论—实践—理论—实践”的编排模式,以利于读者学习和掌握。

5. 强调“阅读、编写、调试”三大能力的培养。这是确定能否顺利掌握 C 语言的根基,本书在许多案例的分析过程中,均渗透了这三种能力的培养。

本书所有案例的调试环境均为 Microsoft Visual C++ 6.0,以适应现实的需要和全国计算机等级考试二级(C 语言)考试环境,并且所有案例均调试通过。

本书由李震平、李晓霞担任主编,由李震平、李晓霞、杨丽负责统稿,由湖南大学计算机与通信学院院长李仁发担任主审。其中第一、第二章由杨丽编写,第三章由李晓霞编写,第四章由魏红君编写,第五章由刘群编写,第六章由韩晓鸿编写,第七章由程慧芳编写,第八、第十章及附录四由李震平编写,第九章由马丽艳编写,附录一由李燕编写,附录二由常志英编写,附录三由刘传志编写。

在本书的撰写过程中,得到了河北工程大学张永强教授和贺洪江教授及天津大学李鹰的大力支持和指导,在此表示衷心感谢。同时还要感谢张艳丽、赵辉、刘志敏、杨丽(大)、薛红梅、宁振刚、申艳光、王瑞林、范永健、王彬丽、杜巍等,他们为本书的撰写亦提供了有益的建议、帮助和支持。

由于作者水平有限,书中错误和不当之处在所难免,恳请读者批评指正。

李震平

2014 年 9 月



CONTENTS

目 录

第一章 软件开发基础——工资管理系统之项目管理	1
任务一 学习目标	1
任务二 了解软件工程	1
任务三 了解数据结构	5
任务四 了解算法设计	7
任务五 本章要点汇总	11
任务六 实力检验	12
第二章 C 语言程序设计概述	13
任务一 了解程序与程序设计	13
任务二 了解 C 语言	18
任务三 了解 Visual C++ 6.0 集成开发环境	21
任务四 创建、编辑源程序	22
任务五 编译、连接、运行和调试	24
任务六 程序的保存、打开、关闭与退出	30
任务七 本章要点汇总	31
任务八 实力检验	32
第三章 结构化程序的三大结构——工资管理系统之主菜单设计	34
任务一 学习目标	34
任务二 掌握 C 语言的词法	35
任务三 理解 C 语言的数据类型	36
任务四 理解变量与常量	38
任务五 掌握运算符、表达式的使用	42
任务六 掌握输入/输出数据的方法	52
任务七 顺序结构程序设计	65
任务八 选择结构程序设计	66
任务九 循环结构程序设计	77
任务十 知识扩展	88



任务十一 精选案例	91
任务十二 应用实例——工资管理系统之主菜单设计	99
任务十三 本章要点汇总	100
任务十四 实力检验	101
第四章 数组——工资管理系统之数据管理	106
任务一 学习目标	106
任务二 数组基础	107
任务三 数组应用	112
任务四 知识扩展——二维数组与多维数组	130
任务五 精选案例	137
任务六 本章要点汇总	141
任务七 实力检验	142
第五章 结构体数据类型——工资管理系统之数据结构设计	143
任务一 学习目标	143
任务二 声明结构体类型	143
任务三 职工工资数据结构设计与结构体变量定义	145
任务四 输入职工工资	149
任务五 计算职工工资	154
任务六 输出职工工资	155
任务七 知识扩展	155
任务八 精选案例	159
任务九 本章要点汇总	162
任务十 实力检验	163
第六章 函数——工资管理系统之模块功能的实现	165
任务一 学习目标	165
任务二 了解模块化程序设计	165
任务三 了解函数的作用	166
任务四 函数的分类	167
任务五 函数的定义	169
任务六 函数的类型和返回值	172
任务七 函数调用及参数传递	175
任务八 函数的嵌套调用、递归调用	180
任务九 知识扩展 1——函数的作用域	184
任务十 知识扩展 2——变量的作用域与存储方式	185
任务十一 精选案例	191
任务十二 本章要点汇总	203
任务十三 实力检验	204

第七章 指针——工资管理系统之双向链表结构设计及应用	206
任务一 学习目标	206
任务二 了解指针的概念	206
任务三 指针变量的定义与初始化	208
任务四 掌握指针的常规应用	211
任务五 了解指针与数组	213
任务六 了解指向结构体变量的指针	224
任务七 了解指针与函数	226
任务八 知识扩展	229
任务九 精选案例	242
任务十 本章要点汇总	247
任务十一 实力检验	248
第八章 文件操作——工资管理系统之数据永久记录	251
任务一 学习目标	251
任务二 了解文件	251
任务三 了解文件的存储形式	252
任务四 了解文件操作	252
任务五 打开文件——fopen()	252
任务六 将职工工资数据写入文件——fwrite()函数的应用	255
任务七 从文件中读取职工工资数据——fread()函数的应用	256
任务八 关闭文件——fclose()函数的应用	257
任务九 更多的读写文件函数	257
任务十 了解文件数据的定位	262
任务十一 知识扩展	265
任务十二 精选案例	265
任务十三 本章要点汇总	273
任务十四 实力检验	274
第九章 编译预处理——工资管理系统之预处理应用	278
任务一 学习目标	278
任务二 文件包含	278
任务三 宏定义	279
任务四 条件编译	285
任务五 知识扩展	288
任务六 精选案例	288
任务七 本章要点汇总	290
任务八 实力检验	291

第十章 C 语言程序设计之技术与知识点分类汇总	294
一、C 语言程序设计之结构	294
二、C 语言程序设计之语句	295
三、C 语言程序设计之数据类型	295
四、C 语言程序设计之运算符的优先级与结合性	296
五、C 语言程序设计之函数	297
六、C 语言程序设计之指针	297
七、C 语言程序设计之文件操作	298
八、C 语言程序设计之编译与预处理	298
附录一 基于软件工程的工资管理系统软件开发(附源程序)	300
附录二 工资管理系统链表结构程序源代码	319
附录三 ASCII 字符编码一览表	329
附录四 Turbo C 部分常用库函数	330
参考文献	336

第一章

软件开发基础——工资管理系统之项目管理

任务一 学习目标

C 语言(Combined Language)是最通用和流行的计算机程序设计语言之一。Dennis M. Ritchie 于 1972 年在 BCPL(Basic Combined Programming Language)的基础上设计出 C 语言,并用 C 语言重写了第一个在 DEC PDP-11 计算机上实现的 UNIX 操作系统。从诞生至今,C 语言历经了 40 余年,仍然以其紧凑的代码、高效的运行效率、强大的功能,以及可以直接操作计算机硬件等特点,深受广大编程爱好者和软件开发人员的欢迎。

C 语言的应用领域十分广泛,如人工智能、单片机控制、计算机系统软件等的底层开发、工业控制、智能仪表、嵌入式系统及硬件驱动程序开发等。近年来,在“飞思卡尔杯全国大学生智能汽车竞赛”、“亚太大学生机器人大赛”及某些面向在校大学生的各种电子竞赛中,许多参赛院校都是使用 C 语言编制其控制程序的。有人风趣地形容:“学好 C 语言,走遍天下都不怕”。因为,目前有较多高级语言都是在 C 语言的基础上设计出来的,如 C++、JAVA 等。

在学习 C 语言之前,需要先了解一些程序设计与开发的相关知识,这些是程序设计的基础。本章的后续 3 项任务中分别介绍了软件工程、数据结构和算法设计的一些基本概念和思想。如果读者需要深入理解和掌握这些理论知识,可以查阅相关书籍。

任务二 了解软件工程

1. 软件的概念及特点

软件是程序、数据及其相关文档的完整集合。也可以说,软件由两部分组成:一是程序和数据;二是文档。文档是与软件的程序设计、开发、维护和使用有关的图文资料。

软件具有以下几个特点。

①软件是一种逻辑产品,它与物质产品有很大的区别,以程序和文档的形式出现,保存在计算机的存储器中(磁盘或光盘),通过计算机的运行才能体现它的功能和作用。

②软件产品的生产主要是研制。其成本主要体现在软件的开发和研制上,软件开发研制完成后,通过复制可以生产大量相同的软件产品。



③软件产品不会用坏,不存在磨损、消耗的问题,但是软件在使用过程中,往往需要根据用户需求变化或软硬件环境的变化对软件进行修改,这种修改被称为软件维护。

④软件产品的生产主要是脑力劳动。

⑤软件的费用是不断增加的,软件成本相当昂贵。

2. 软件工程相关知识

1) 软件工程

20 世纪 60 年代中后期,软件数量越来越多,规模越来越大,而软件的生产基本上是各自为战,缺乏科学规范的系统规划与测试、评估标准,致使大批耗费巨资研制出来的软件系统因有错误而无法使用,造成了巨大损失。由于问题软件不断出现,给人的感觉是软件越来越不可靠。这一切极大地震撼了计算机界,历史上称为“软件危机”。一些软件开发人员认识到:大型程序的编制不同于编写小程序,它是一项新的技术,应该像管理工程一样管理软件研制的全过程。于是,1968 年北大西洋公约组织在联邦德国的一次学术会议上,首次提出了“软件工程”的概念,从而形成一门新兴的学科——软件工程。

软件工程是一门用工程化方法,指导计算机软件开发和维护的学科,它采用工程的概念、原理、技术和方法来开发、维护以及管理软件。软件工程是目前计算机学科中一个年轻并且充满活力的研究领域。

2) 软件工程的 3 要素

软件工程包括 3 个重要因素,即方法、工具和过程。

①软件工程的方法,是完成软件工程项目的手段。它包括许多方面,例如,项目计划与估算、软件需求分析、系统总体结构设计、数据结构设计、算法设计、编写代码、软件测试及维护等。

②软件工程的工具,是人类在开发软件的活动中智力和体力的扩展与延伸,它提供了自动或半自动的软件支撑环境。

③软件工程的过程,贯穿于软件开发的各个环节,支持软件开发各个环节的控制和管理。

3) 软件生命周期

任何有生命的动物、植物和人,都有一个生存周期,没有生命的事物或实体。例如,PC 机(个人计算机)、家具、房子、汽车等,也有一个生存周期,这个生存周期就是使用寿命。软件和其他事物相似,有它的产生、发展和消亡的过程。

软件生命周期是指从开发软件概念的提出起,直到该软件的使用、失去使用价值而被废弃的整个过程,也可以称作软件生存周期。

在软件生命周期中,每个阶段都有确定的任务,并产生一定规格的文档传递给下一阶段,下一阶段在前一阶段所提供文档的基础上继续开展工作。软件生命周期包括以下六个阶段。

(1) 可行性研究与计划制订

在此阶段,软件开发方与需求方需要共同讨论以确定软件的开发目标,开发方还要进行软件项目的可行性研究。可行性研究的目的是研究这个软件项目是否值得去开发,以及其中的问题能否解决。可行性研究的内容包括四个方面。

①经济可行性:进行成本/效益分析,从经济角度判断系统开发是否“合算”。

②技术可行性:进行技术风险评估,从开发者的技术实力、工作经验、问题的复杂程度等方面出发,判断系统开发在时间、费用等限制条件下软件项目能否实现。

③社会可行性:研究在系统开发过程中可能涉及人力资源、各种合同、知识产权纠纷、责任以及各种与法律相抵触的问题。

④方案的选择:提出并评价系统的各种开发方案,从中选出一种最适宜的方案。

(2)需求分析

软件需求是指用户对目标系统在功能、行为、性能等方面的期望。需求分析是发现、求精、建模和产生规格说明的过程。软件开发人员需要在对应用问题及环境的理解和分析基础上,为问题涉及的信息、功能及行为建立模型。需求分析阶段研究的对象是用户对软件项目的要求。在此阶段,一方面,必须全面理解用户的各项要求,但并不一定全盘接受所有的要求;另一方面,要准确地表达出所接受的用户要求,只有经过确切描述的软件需求才能成为软件设计的基础。

需求分析阶段的工作可概括为四个方面:需求获取、需求分析、编写需求规格说明书和需求评审。

(3)软件设计

软件设计是将需求分析阶段得出的逻辑模型设计成具体的软件方案。软件设计主要包括设计软件的总体结构、数据结构和软件具体模块的算法,解决“系统怎么做”的问题。软件设计阶段结束之前,也要进行有关评审,评审通过后才能进入编码阶段。软件设计阶段也可以根据具体软件的规模、类型等决定是否细分成概要设计和详细设计两个阶段。

本书中的软件项目实例“河北工程大学职工工资管理系统”,实现对学校职工的有关个人信息和工资信息进行存储与管理。

①软件的总体结构设计:工资管理的软件结构如图 1-1 所示。

②数据结构设计:采用链表处理工资信息。工资信息包括职工工号、职工姓名、岗位工资、薪级工资、职务津贴、绩效工资、应发工资、个人所得税和实发工资。

③具体模块的实现算法设计:查询职工工资记录、修改职工工资记录、添加职工工资记录、删除职工工资记录。

每个模块的算法的设计为下一个“软件实现”阶段做准备。比如,查询工资记录模块,该模块的算法设计思路是以“工号”为查询对象,在全体职工范围内进行查找,通过比较“工号”找到相应的工资记录并输出,如果找不到,则显示“职工工号有误或不存在”的提示信息,并返回到主菜单。具体算法可以采用流程图或自然语言进行描述。

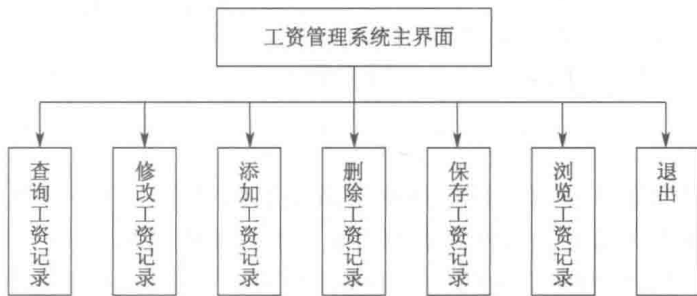


图 1-1 工资管理系统功能模块图

(4)软件实现

软件实现即编写代码。此阶段程序员按照软件设计说明书的要求,为每个模块编写程序



代码。编写的程序应该结构合理、清晰易读,并且与设计说明书要求一致。

(5)软件测试

软件测试是保证软件质量的重要手段,目的是尽可能多地发现程序中的错误。它分为模块测试、组装测试和确认测试。在软件测试阶段要做好软件测试记录,写出软件测试与评价报告。

(6)运行与维护

运行与维护是对软件运行提供支持,软件系统经过测试后仍可能存在潜在的错误或缺陷,当软件投入运行后,这些潜在的问题可能会显现出来。用户的需求和系统的运行环境也可能发生改变,需要对软件进行修改,成功维护后的软件成为新的软件版本,进行软件维护也要详细做好维护修改记录。

4)软件开发模型

软件开发模型是软件开发全部过程、活动和任务的结构框架。它能直观表达软件开发全过程,明确规定要完成的主要活动、任务和开发策略。常用的软件开发模型有:瀑布模型、快速原型模型、螺旋模型、基于构件的开发模型以及基于知识的模型。

①瀑布模型严格按照生存周期各个阶段的目标、任务、文档和要求来进行软件开发。强调了每一阶段的严格性,尤其是开发前期的良好需求说明。开发人员试图在每一阶段活动结束后,通过严格的阶段性复审与确认,得到该阶段的一致、完整、准确和无二义性的文档,冻结这些文档作为该阶段的结束标志,并为下一阶段的活动奠定基础,从而形成一个理想的线性开发序列,以每一步的正确性和完整性来保证最终系统的质量。瀑布模型如图 1-2 所示。

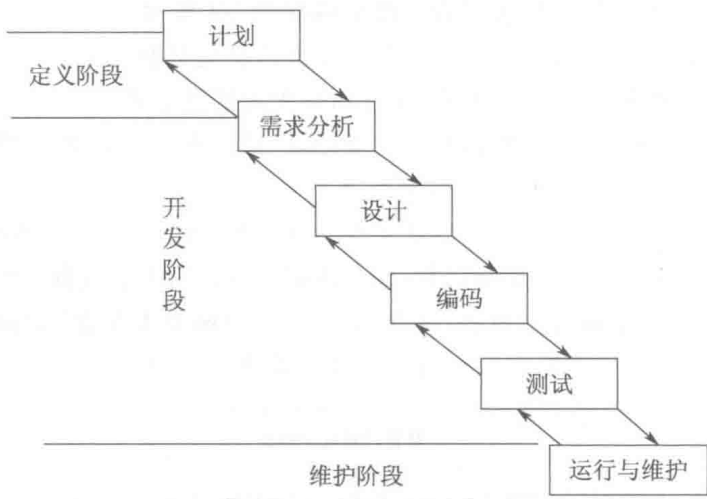


图 1-2 软件生存周期的瀑布模型

瀑布模型在大量的软件开发实践中逐渐暴露出它的缺点。它是一种理想的线性开发模式,缺乏灵活性,特别是无法解决软件需求不明确或不准确的问题。这些缺点给软件开发带来了严重的影响,最终可能导致开发出的软件不是用户真正需要的,并且这一点在开发过程完成后才能发现。

②原型模型即快速模型。其第一步是快速建立一个能反映用户主要需求的原型系统,让用户在计算机上使用它,通过实践来了解目标系统的概貌。通常,用户试用原型系统之后会提

出许多修改意见,开发人员按照用户的意见快速地修改原型系统,然后再次请用户试用。一旦用户认为这个原型系统确实能满足他们的需求,开发人员便可根据这份规格说明文档设计软件,此软件可以满足用户的真实需求。

其他模型的开发过程可查阅软件工程方面的书籍。

3. 软件工程目标

软件工程是一门工程性学科,目的是成功地建造一个大型软件系统,同时达到以下几个目标:较低的成本、较好的性能、完善的功能、较好的移植性、较低的维护费用、及时的交付以及较高的可靠性等。

任务三 了解数据结构

1. 数据结构的概念

数据结构是指相互之间存在一种或多种特定关系的数据元素的集合,是计算机存储、组织数据的方式。数据结构主要研究和讨论数据的逻辑结构、存储结构和数据的运算。

2. 数据结构的分类

数据结构分为逻辑结构和存储结构两大类,具体分类如图 1-3 所示。

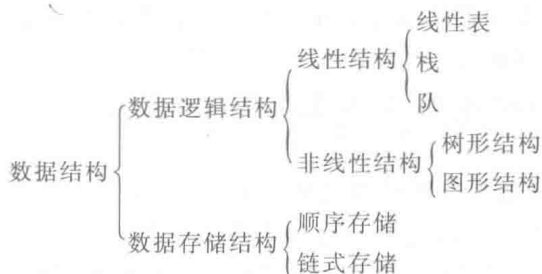


图 1-3 数据结构分类

(1) 数据的逻辑结构

数据的逻辑结构是对数据元素之间的逻辑关系的描述,是数据的组织形式。数据元素之间有四种基本逻辑结构:集合、线性结构、树形结构和图状结构。

①集合:其中任意两个数据元素间没有逻辑关系,组织形式松散,如图 1-4 所示。

②线性结构:数据元素间构成一种顺序的线性关系,如图 1-5 所示。

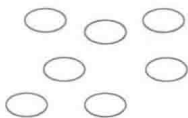


图 1-4 集合



图 1-5 线性结构

③树形结构:具有分支、层次特性,数据元素间形成一种一对多的关系,如图 1-6 所示。

④图状结构:也叫网状结构,是最复杂的一种,结构中数据元素之间存在多对多的关系,如图 1-7 所示。

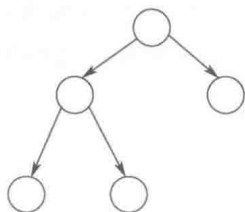


图 1-6 树形结构

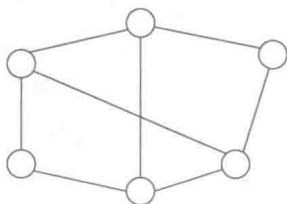


图 1-7 图状结构

(2) 数据的存储结构

数据的逻辑结构是独立于计算机的、与数据在计算机中的存储无关,但是,要使用计算机对数据进行处理,就必须将数据存储于计算机中,数据结构在计算机中的表示称为数据的存储结构,也称为物理结构。那么,数据在存储器中是如何存放的呢?试想,如果英汉词典中的单词没有任何规律地随意罗列,那么,用这样的词典查词将犹如大海捞针。同样,如果数据在计算机中无规律地存储,则处理时也会非常困难。

数据的存储结构包括顺序存储结构、链式存储结构、索引存储结构和散列存储结构四种。顺序存储结构通常用于存储线性结构的数据,逻辑上相邻的节点在存储器中也是相邻的。在链式存储结构中,节点在存储器中随意存放,逻辑上相邻的节点在存储器中不一定是相邻的,也就是说节点之间的物理关系与逻辑关系无关。链式存储结构的特点是每个节点由数据域和指针域组成;插入、删除节点灵活,不必移动节点,只需要改变节点中的指针;查找节点时链式存储要比顺序存储慢。索引存储结构中每个节点仅含一个数据元素,所有节点连续存放,此外,增设一个索引表,索引表中的索引号指示各存储节点的存储位置,换句话说,根据节点的索引号可以确定该节点的存储地址。散列存储结构的思想是构造一个从集合 K 到存储区域 M 的函数 h ,该函数的定义域为 K ,值域为 M , K 中的每个节点 k_i 在计算机中的存储地址由 $h(k_i)$ 确定。

在程序设计中,选择合适的数据结构是非常重要的。许多大型系统的开发经验表明,系统实现的难易程度、系统的质量和执行效率在很大程度上取决于是否选择了最优的数据结构。选择合适的数据结构,可以使算法的设计更加容易。当然也可以根据特定算法来选择数据结构与之适应。但不论哪种情况,选择合适的数据结构都是非常重要的,它可以带来更高的运行效率和存储效率。

(3) 线性结构与非线性结构

根据数据结构中各元素之间的前驱后继关系不同,一般将数据结构分为线性结构和非线性结构两大类。

如果在一个数据结构中,数据元素之间为一对一的线性关系,第一个元素没有直接前驱,最后一个元素没有直接后继,其余元素都有且只有唯一的一个直接前驱和直接后继,则称这种数据结构为**线性结构**。如果一个数据结构不是线性结构,则称为**非线性结构**。

(4) 常用的几种数据结构

①**线性表**:其中的数据元素是一一对应的关系,即除了第一个和最后一个数据元素之外,其他数据元素都是首尾相接的。它是最基本、最简单的一种数据结构,便于实现和操作,因此,线性表在实际应用中被广泛采用。

线性表的存储结构有两种:顺序存储结构和链式存储结构。

如果线性表采用顺序存储的方式存储则称为顺序表。顺序表是将表中的节点依次存放在

计算机内存中一组地址连续的存储单元中。由于一维数组采用顺序存储结构,因此,可以用数组来实现线性表的顺序存储结构。线性链表是具有链式存储结构的线性表。

线性表可以进行插入、删除、排序、查找、计数等操作。

②栈:是只能在一端进行插入和删除操作的特殊线性表。它按照“后进先出”的原则存储数据,先进入的数据被压入栈底,最后进入的数据在栈顶,需要读数据的时候从栈顶开始弹出数据(最后进入的数据被第一个读出来)。

③队列:是只允许在表的前端(front)进行删除操作、在表的后端(rear)进行插入操作的线性表。进行插入操作的端称为队尾,进行删除操作的端称为队首。队列是“先进先出”的线性表。队列中没有元素时,称为空队列。

④链表:是一种物理存储单元上非连续、非顺序的存储结构,数据元素的逻辑顺序是通过链表中的指针链接次序实现的。链表由一系列节点(链表中每一个元素称为节点)组成,节点可以在运行时动态生成。每个节点包括两部分:一部分是存储数据元素的数据域,另一部分是存储下一个节点地址的指针域。

⑤树:是包含不少于一个节点的有穷集合。例如,有 $n(n>0)$ 个节点的集合 K ,且在 K 中定义了一个关系 N , N 满足以下条件:有且仅有一个节点 k_0 ,它对于关系 N 来说没有前驱,称 k_0 为树的根节点。除 k_0 外, K 中的每个节点对于关系 N 来说,有且仅有一个前驱,可以有 m 个后继($m \geq 0$)。树是一种简单的非线性结构,所有元素之间具有明显的层次特性。

⑥图:由节点的有穷集合 V 和边的集合 E 组成。其中,为了与树形结构加以区别,在图结构中常常将节点称为顶点。若有两个顶点 v_i, v_j 之间存在一条边,则用偶对 (v_i, v_j) 来表示无向边,用 $\langle v_i, v_j \rangle$ 来表示从 v_i 到 v_j 的有向边。

任务四 了解算法设计

1. 算法的概念

算法是解决问题所采取的步骤和方法。在人们的生产活动和日常生活中,算法也无处不在。例如,超市购物的算法:确定购买的商品清单;确定采购的超市;到达超市;购物商品;返回。还有经典的“五位哲学家就餐”的问题:五位哲学家围坐一圆桌,每人面前有一盘面条,每人左手侧和右手侧各有一叉子。吃面条时必须左手和右手各拿一叉子。如果五位哲学家同时拿起左手侧的叉子,则谁也吃不成面条。这里有一个并行算法问题。可以设计一程序,它能保证任何时候有两位哲学家在吃面条。现实中,很多资源分配和并行算法问题都与该问题类似。

计算机算法分为数值运算算法和非数值运算算法两类。数值运算算法的目的是求数值解,例如,函数的定积分、多项式与线性代数方程组,矩阵与非线性方程等。非数值运算算法的范围很广,最常见的是事务管理领域,例如,考勤管理、职工信息管理、超市商品管理等。目前,计算机在非数值运算方面的应用远远超过了数值运算方面。

下面通过一个简单的例子来说明如何设计一个算法。

【案例 1-1】 计算 $1+2+3+\cdots+100$ 。

【分析】

顺序算法:

第一步:计算 1 加 2 得 3;

第二步:将 3 与 3 相加得 6;

第三步:再将 6 与 4 相加得 10;

.....

最终算法描述中需要写 99 个步骤,显然此方法十分烦琐,不是最佳算法。

改进算法:

不妨设一个 s 存放第一个加数,一个 i 存放第二个加数,相加后将和重新放在 s 中作为新的第一个加数,修改第二个加数 i (i 值加 1),重复 $s+i \rightarrow s$ 及 $i+1 \rightarrow i$,直到加完。描述如下:

第一步:使 $s=0$;

第二步:使 $i=1$;

第三步: $s+i$ 的和仍放在 s 中,即 $s+i \rightarrow s$;

第四步:使 i 的值加 1,即 $i+1 \rightarrow i$;

第五步:如果 i 不大于 100,重复第三、四、五步;否则,输出 s 的值后结束算法。

【思考】 如何描述 $1+3+5+\cdots+99$ 的算法?

【方案】 只要将算法的第四步改为: $i+2 \rightarrow i$ 即可。

由此可见,改进的算法具有一定的灵活性和通用性。

【案例 1-2】 从键盘输入任意两个不等的整数,输出较大的数。

【分析】

第一步:从键盘输入两个不相等的整数 x 和 y ;

第二步:比较 x 和 y 的大小,如果 $x > y$,执行第三步,否则执行第四步;

第三步:输出 x ,算法结束;

第四步:输出 y ,算法结束。

【思考】 如何输出较小的数?

【方案】 将案例 1-2 的算法的第二步中的“ $x > y$ ”改为“ $x < y$ ”即可。

2. 算法的特性

一个算法应具有以下几个特性。

(1) 有穷性

一个算法应包含有限的操作步骤,在执行有穷步后能够结束。对于包含循环结构的算法,应避免出现死循环,否则就会无限制地执行下去。当然,“有穷性”应该是在合理的限度之内,如果一个算法执行 100 年才结束,这虽然也是有穷的,但明显超过了合理的限度,因此,也不能称为有效的算法。这个“合理限度”是多少其实并没有严格的标准,由人们的常识和需要而定。

(2) 确定性

算法中的每一个步骤必须有确切的含义,不会产生歧义,并且在任何条件下,算法只有唯一的一条执行路径,对于相同的输入只能得到相同的结果。

(3) 有效性

算法的每一个步骤都应当能有效地执行,并得到确定的结果。例如,若 $b=0$,那么 a/b 就不能有效执行。

(4) 有零个或多个输入

输入是指在执行算法时需要从外界取得必要的信息,例如,在执行案例 1-2 的算法时,需要输入两个不等的整数,然后才能比较大小并输出。如果一个算法在程序中已经给出原始数据,那么此算法可以没有输入,即在执行算法时不需要输入任何信息。