

■权威 ■实用 ■紧扣大纲 ■直击要点 ■链接真题 ■一点就通

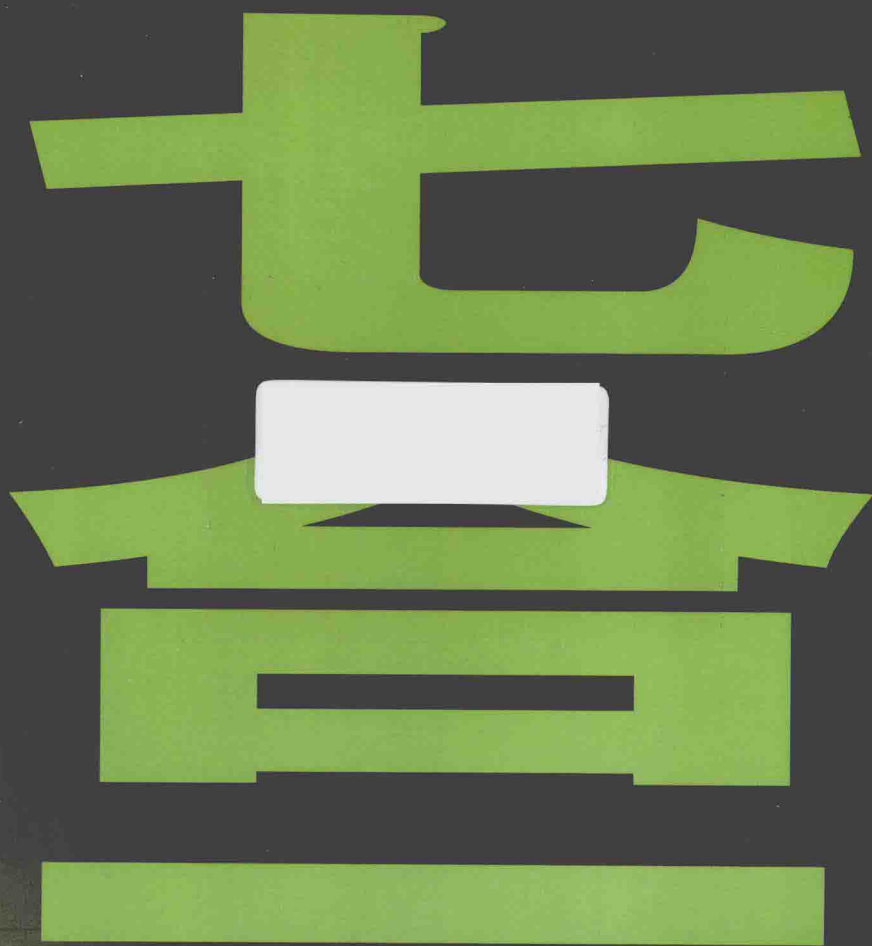
全国计算机等级考试 七合一应试通

——二级C语言

全国计算机等级考试命题研究组 编

(选择题考点精讲+选择题真题集+操作题考点精讲+操作题真题集+
选择题真题试卷+操作题真题试卷+智能模拟软件)

多角度细致点评 讲解精髓
题目精选 针对性强



北京邮电大学出版社
www.buptpress.com



(送考试光盘一张)

全国计算机等级考试七合一应试通

——二级 C 语言

全国计算机等级考试命题研究组 编

北京邮电大学出版社
· 北京 ·

内 容 简 介

本书结合最新版考试大纲,以历年真题(库)为基础,结合编者多年从事命题、阅卷及培训辅导的实际经验编写而成。本书章节安排与官方教程同步。分为“选择题考点精讲”、“选择题真题集”、“操作题考点精讲”、“操作题真题集”、“选择题真题试卷”、“操作题真题试卷”、“智能模拟软件”七个部分。“选择题考点精讲”中归纳出本节的核心知识点,对考点、重点、难点内容进行解释与剖析;“操作题考点精讲”精选出本节常在上机中考核的难点,揭示命题规律,把握考试范围;“选择题真题集”中精选出最近几次考试真题进行详细分析,增强学生解题能力;“操作题真题集”中挑选了历年常考真题,并对其进行细致深入的分析和解答,让考生透彻理解和掌握。提供多套选择题和操作题试卷,并配有详细答案解析,全面模拟真实考试,供考生实战演练。

本书配有智能模拟光盘。盘中提供数十套全真无纸化试题,其考试界面、考试过程、题型等与真实考场完全相同,便于考生实战演练,引领考生过关。

本书非常适合有关考生使用,作为各类全国计算机等级考试培训班的教材,以及大、中专院校师生的参考书。

图书在版编目(CIP)数据

全国计算机等级考试七合一应试通. 二级 C 语言/全国计算机等级考试命题研究组编. --北京:北京邮电大学出版社,2014.1
ISBN 978-7-5635-3296-4

I. ①全… II. ①全… III. ①电子计算机—水平考试—自学参考资料②C 语言—程序设计—水平考试—自学参考资料
IV. ①TP3

中国版本图书馆 CIP 数据核字(2012)第 269650 号

书 名: 全国计算机等级考试七合一应试通——二级 C 语言

作 者: 全国计算机等级考试命题研究组

责任编辑: 满志文 刘 磊

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(邮编:100876)

发 行 部: 电话: 010-62282185 传真: 010-62283578

E-mail: publish@bupt.edu.cn

经 销: 各地新华书店

印 刷: 北京联兴华印刷厂

开 本: 889 mm×1 194 mm 1/16

印 张: 14.5

字 数: 708 千字

版 次: 2014 年 1 月第 1 版 2014 年 1 月第 1 次印刷

ISBN 978-7-5635-3296-4

定 价: 29.50 元

• 如有印装质量问题,请与北京邮电大学出版社发行部联系 •

前 言

全国计算机等级考试是经原国家教育委员会(现教育部)批准,由教育部考试中心主办,面向社会,用于考察应试人员计算机应用知识与技能的全国性计算机水平考试体系。

为了引导考生顺利通过全国计算机等级考试,我们根据新大纲的要求,结合常考真题,按教育部考试中心指定教材的篇章结构,由从事全国计算机等级考试试题研究人员及在等级考试第一线从事命题研究、教学、辅导和培训的老师精心编写了《全国计算机等级考试七合一——二级 C 语言》。

本套书定位是“合七为一”的高效实用考前辅导书,将“选择题考点精讲”、“选择题真题集”、“操作题考点精讲”、“操作题真题集”、“选择题真题试卷”、“操作题真题试卷”、“智能模拟软件”七大块结合在一本书中,具体特点如下:

(1) 策划思路新颖。前四个板块“选择题考点精讲”、“选择题真题集”、“操作题考点精讲”、“操作题真题集”的特色如下。

- 选择题考点精讲:归纳出核心知识点,对考点、重点、难点内容进行详解与剖析。
- 选择题真题集:精选出最近几次考试真题进行详细分析,增强学生解题能力。
- 操作题考点精讲:精选出常在上机操作题中考核的要点,揭示命题规律,把握考试范围。
- 操作题真题集:挑选了历年常考真题,对其进行细致深入的分析和解答,以让考生透彻地理解和掌握。

(2) 全面真实模拟。从 2013 年 9 月真题库中精选了多套选择题和操作题试卷,并配有详细解答,供考生考前实战演练。

(3) 智能模拟,书盘结合。本书为考生提供智能模拟光盘。盘中提供十余套全真无纸化试题,其考试界面、考试过程、题型等与真实考场完全相同,便于考生实战演练,引领考生过关。

本书非常适合参加全国计算机等级考试二级 C 考试的考生考前复习使用,也可以作为相关等级考试培训班的辅导教材。

本书由陈帅,史国川主编,参与本书编写、校对的还有秦海泉、陈忠贤、樊圣兰、俞翠兰、陈海霞、袁鸿鹏、陈斌、蔡季平、陈娟娟、刘卉、徐杨阳、白晖、李海磊、顾兴建、王连涛、陈海燕、赵海峰、李晓飞、吴松松、何光明等。

由于作者水平有限,书中难免存在错误和不妥之处,恳请广大读者批评指正。联系邮箱:bjbaba@263.net。(也请参与我们的微博活动吧!活动如下:①关注@北邮等考,成为北邮等考的粉丝。②转发此微博:“北邮出版的等考图书刚买到,相信能成功。全国计算机等级考试复习资料首选北邮出版的。”并说出你购买图书、参加考试的心情和故事,也可以是生活中的乐趣。我们将对优秀粉丝进行送礼,一直有效。)

全国计算机等级考试命题研究组

目 录

第一部分 选择题考点精讲

第 1 章 公共基础知识	2	8.4 函数的说明	55
1.1 数据结构与算法	2	8.5 调用函数和被调用函数之间的数据传递	55
1.2 程序设计基础	7	第 9 章 指针	56
1.3 软件工程基础	8	9.1 变量的地址和指针	56
1.4 数据库设计基础	14	9.2 指针变量的定义和赋值	56
第 2 章 程序设计基本概念	21	9.3 对指针变量的操作	57
2.1 程序和程序设计	21	9.4 函数之间地址值的传递	58
2.2 算法	22	第 10 章 数组	60
2.3 结构化程序设计和模块化结构	23	10.1 一维数组的定义和元素的引用	60
第 3 章 C 程序设计的初步知识	25	10.2 一维数组和指针	61
3.1 C 语言的构成和格式	25	10.3 函数之间对一维数组和数组元素的引用	62
3.2 常量、变量和标识符	26	10.4 二维数组的定义和元素的引用	63
3.3 整型数据	27	10.5 二维数组和指针	65
3.4 实型数据	28	10.6 二维数组名和指针数组作为实参	65
3.5 算术表达式	29	第 11 章 字符串	67
3.6 赋值表达式	30	11.1 利用一维字符数组存放字符串	67
3.7 自增、自减和逗号运算符	31	11.2 指针指向一个字符串	68
第 4 章 顺序结构	32	11.3 字符串的输入和输出	68
4.1 赋值语句	32	11.4 字符串数组	70
4.2 数据输出	32	11.5 用于字符串处理的函数	70
4.3 数据输入	33	第 12 章 对函数进一步讨论	73
4.4 复合语句和空语句	35	12.1 传给 main 函数的参数	73
第 5 章 选择结构	36	12.2 通过实参向函数传递函数名或指向函数的指针变量	74
5.1 关系运算和逻辑运算	36	12.3 函数的递归调用	74
5.2 if 语句构成的选择结构	37	第 13 章 用户标识符的作用域和存储类	76
5.3 条件表达式构成的选择结构	39	13.1 局部变量、全局变量和存储分类	76
5.4 switch 语句和 break 语句构成的选择结构	40	13.2 局部变量的作用域和生存期	77
5.5 语句标号和 goto 语句	42	13.3 全局变量的作用域和生存期	78
第 6 章 循环结构	43	13.4 函数的存储分类	79
6.1 while 语句构成的循环结构	43	第 14 章 编译预处理和动态存储分配	80
6.2 do-while 语句构成的循环结构	43	14.1 编译预处理	80
6.3 for 语句构成的循环结构	44	14.2 动态存储分配	81
6.4 循环结构的嵌套	46	第 15 章 结构体、共用体和自定义类型	83
6.5 break 和 continue 语句的作用	47	15.1 用 typedef 说明一种新类型名	83
第 7 章 字符型数据	48	15.2 结构体类型	83
7.1 字符型常量和字符型变量	48	15.3 共用体类型	89
7.2 字符的输入和输出	50	第 16 章 位运算	91
第 8 章 函数	52	16.1 位运算符	91
8.1 库函数	52	16.2 位运算符的运算功能	91
8.2 函数的定义和返回值	52	第 17 章 文件	93
8.3 函数的调用	53	17.1 C 语言文件的概念	93
		17.2 文件的操作	93

第二部分 选择题真题集

第 1 章真题集	96
第 2 章真题集	100
第 3 章真题集	100
第 4 章真题集	102
第 5 章真题集	103
第 6 章真题集	105
第 7 章真题集	107
第 8 章真题集	108
第 9 章真题集	110
第 10 章真题集	110
第 11 章真题集	114
第 12 章真题集	117
第 13 章真题集	117
第 14 章真题集	119
第 15 章真题集	120
第 16 章真题集	121
第 17 章真题集	122
选择题真题集答案与解析	123

第三部分 操作题考点精讲

无纸化系统使用说明	136
考点 1 用 if 循环解决排序问题	138
考点 2 switch 与循环结合解决元素的遍历统计	140
考点 3 while 循环	141
考点 4 do-while 循环	143
考点 5 for 语句	144
考点 6 循环嵌套	145
考点 7 函数的调用	146
考点 8 参数传递	148
考点 9 指针变量	149
考点 10 函数中的指针传递	150
考点 11 指针的运算	151
考点 12 二维数组	153
考点 13 字符串处理函数	155
考点 14 递归	156
考点 15 宏定义	157

考点 16 结构体变量	158
考点 17 文件操作	160

第四部分 操作题真题集

操作题考点 1 真题集	166
操作题考点 2 真题集	167
操作题考点 3 真题集	167
操作题考点 4 真题集	168
操作题考点 5 真题集	168
操作题考点 6 真题集	170
操作题考点 7 真题集	171
操作题考点 8 真题集	172
操作题考点 9 真题集	172
操作题考点 10 真题集	173
操作题考点 11 真题集	173
操作题考点 12 真题集	174
操作题考点 13 真题集	175
操作题考点 14 真题集	176
操作题考点 15 真题集	176
操作题考点 16 真题集	177
操作题考点 17 真题集	178
操作题真题集答案与解析	179

第五部分 选择题真题试卷

2013 年 9 月选择题真题试卷 1	186
2013 年 9 月选择题真题试卷 2	192
2013 年 9 月选择题真题试卷 3	198
选择题真题试卷答案与解析	204

第六部分 操作题真题试卷

2013 年 9 月操作题真题试卷 1	214
2013 年 9 月操作题真题试卷 2	216
2013 年 9 月操作题真题试卷 3	218
操作题真题试卷答案与解析	221

第七部分 智能模拟软件

智能模拟软件功能展示	224
------------------	-----

第一部分

选择题考点精讲

本部分结合最新版考试大纲,章节安排与官方教程结构保持同步,归纳出每章节的核心知识点,对考点、重点、难点内容进行解释与剖析。讲解知识点的同时在正文穿插最新考试真题,帮助考生深入理解知识点,把握考试重点。

- 第1章 公共基础知识
- 第2章 程序设计基本概念
- 第3章 C 程序设计的初步知识
- 第4章 顺序结构
- 第5章 选择结构
- 第6章 循环结构
- 第7章 字符型数据
- 第8章 函数
- 第9章 指针
- 第10章 数组
- 第11章 字符串
- 第12章 对函数进一步讨论
- 第13章 用户标识符的作用域和存储类
- 第14章 编译预处理和动态存储分配
- 第15章 结构体、共用体和自定义类型
- 第16章 位运算
- 第17章 文件

1.1 数据结构与算法

一、算法

算法(Algorithm)是指为解决某个特定问题而采取的确定且有限的步骤的一种描述,它是指令的有限序列,使得给定类型的问题通过有限的指令序列、在有限的时间内被求解。其中每一条指令表示一个或多个操作。

1. 算法的基本特性

- (1) 可行性:是指算法中指定的操作都可以通过基本运算执行有限的次数后实现。
- (2) 确定性:是指算法中每一步骤都必须是有明确定义的,不允许有多义性。
- (3) 有穷性:是指算法必须能在有限时间内做完,即算法必须能在执行有限的步骤之后终止。
- (4) 拥有足够的情报:一个算法的执行结果总是与输入的初始数据有关,它可以有多个输入,也可以不要输入,但必须有一个或多个输出,不同的输入将会有不同的输出结果。

2. 算法复杂度

算法的复杂度主要包括时间复杂度和空间复杂度。

(1) 时间复杂度:是指执行算法所需要的计算工作量。算法的工作量用算法所执行的基本运算次数来度量,而算法所执行的基本运算次数是问题规模的函数,即:算法的工作量 $=f(n)$,通常记作:

$$T(n)=O(f(n))$$

常见的时间复杂度有:

$$O(1)<O(\log_2 n)<O(n)<O(n\log_2 n)<O(n^2)<O(n^3)<O(2^n)$$

(2) 空间复杂度:执行这个算法所需要的内存空间。

二、数据结构

1. 数据结构的定义

数据结构(Data Structure)是指反映数据元素之间关系的数据元素集合的表示。数据结构作为计算机的一门学科,主要研究和讨论以下 3 个方面的问题:

- (1) 数据集中各数据元素之间所固有的逻辑关系,即数据的逻辑结构。
- (2) 各数据元素在计算机中的存储关系,即数据的存储结构。
- (3) 对各种数据结构进行的运算。

2. 数据的逻辑结构

数据元素之间的前后件关系是指它们之间的逻辑关系。所谓数据的逻辑结构,是指反映数据元素之间逻辑关系的数据结构。根据数据元素之间关系的不同特性,通常有 4 类基本的逻辑结构,如图 1-1 所示。

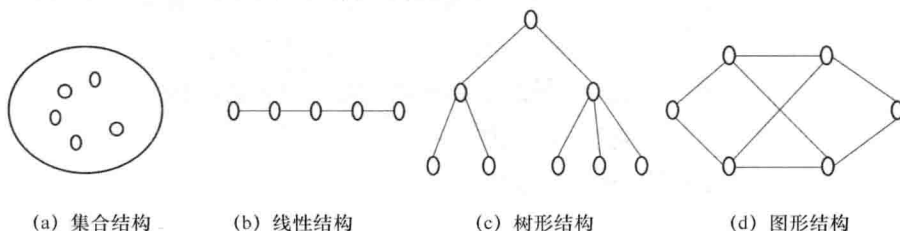


图 1-1 数据的逻辑结构

3. 数据的存储结构

数据的逻辑结构在计算机存储空间中的存放形式称为数据的物理结构,或称数据的存储结构。在数据的存储结构中,不仅要存放各数据元素的信息,还需要存放各数据元素之间的前后件关系的信息。这里要理解顺序存储结构和链式存储结构的区别。

(1) 顺序存储方法:是把逻辑上相邻的元素存储在物理位置相邻的存储单元中。

(2) 链式存储方法:对逻辑上相邻的元素不要求其物理位置相邻,元素间的逻辑关系通过附设的指针字段来表示,由此得到的存储表示称为链式存储结构,链式存储结构通常借助于程序设计语言中的指针类型来实现。

4. 线性结构与非线性结构

根据数据结构中各数据元素之间前后件关系的复杂程度,又可以将数据结构分为两大类:线性结构和非线性结构。

如果一个非空的数据结构满足下列两个条件:

(1) 有且只有一个根结点。

(2) 每个结点最多有一个直接前件,也最多有一个直接后件。

则称该数据结构为线性结构,又称线性表。栈和队列都是线性结构。

如果一个数据结构不是线性结构,则称之为非线性结构。在非线性结构中,一个结点可以有多个直接后件,或者有多个直接前件;或者既有多个直接后件又有多个直接前件。树形结构和图形结构都是非线性结构。

5. 线性表的存储结构

(1) 顺序存储结构。特点如下:

- 线性表中所有元素所占的存储空间是连续的。
- 线性表中各元素在存储空间中是按逻辑顺序依次存放的。

在程序设计语言中,一维数组在内存中占用的存储空间就是一组连续的存储区域,因此,用一维数组来表示顺序表的数据存储区域是再合适不过的。

顺序表的基本运算有:插入运算、删除运算等。

(2) 链式存储结构:将存储空间中的每一个存储结点分为两部分:一部分用于存储数据元素的值,称为数据域;另一部分用于存放下一个数据元素的存储序号(存储结点的地址),即指向后件结点,称为指针域。

为了存储线性表中的每一个元素,一方面要存储数据元素的值,另一方面要存储各数据元素之间的前后件关系。因此,链式存储结构所需要的存储空间比顺序结构要多。

在链式存储结构中,存储数据结构的存储空间可以不连续,各数据结点的存储顺序与数据元素之间的逻辑关系可以不一致,数据元素之间的逻辑关系由指针域来确定。

线性链表的基本运算包括:查找指定元素、插入运算、删除运算等。

循环链表是对线性链表的扩展,增加了一个表头结点,其数据域为任意或者根据需要来设置,指针域指向线性表的第一个元素的结点。循环链表的头指针指向表头结点;循环链表中最后一个结点的指针域不是空,而是指向表头结点。即在循环链表中,所有结点的指针构成了一个环状链。带头结点的单循环链表如图 1-2 所示。

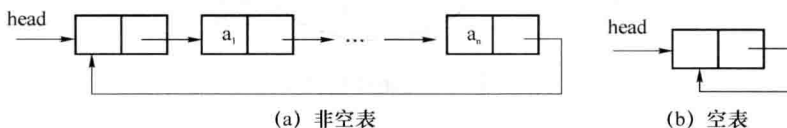


图 1-2 带头结点的单循环链表

在单链表中若需找出某个结点的前件结点,则此时需从表头出发查找。双向链表可克服单链表的这种缺点。在双向链表中,每一个结点除了数据域外,还包含两个指针域,一个指针(next)指向该结点的后件结点,另一个指针(prior)指向它的前件结点。双向循环链表如图 1-3 所示。

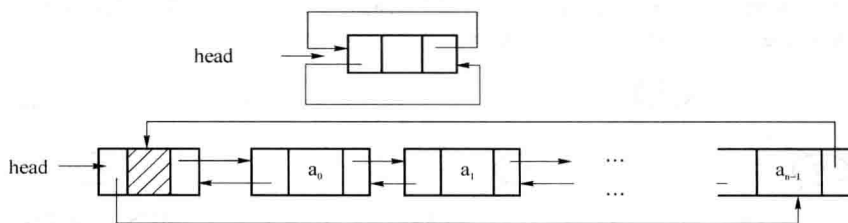


图 1-3 双向循环链表

6. 栈

栈是限定在一端进行插入与删除操作的特殊线性表。即栈的一端是封闭的,不允许进行插入与删除元素;另一端是开口的,允许插入与删除元素。

栈顶元素总是最后入栈的,因而是最先出栈;栈底元素总是最先入栈的,因而也是最后出栈。也就是说,栈是按照后进先出(Last

In First Out, LIFO) 的原则组织数据的。由此可以看出, 栈具有记忆作用。

通常用指针 top 指示栈顶的位置, 用指针 $bottom$ 指向栈底。栈顶指针 top 动态反映栈的当前位置。图 1-4 是一个栈的示意图。通常以 $top=0$ 时空栈, 在元素进栈时指针 top 不断地加 1, 当 top 等于数组的最大下标值时则栈满。

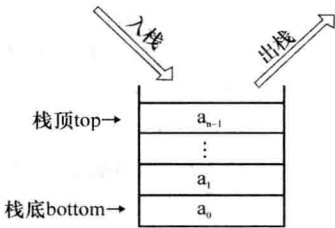


图 1-4 栈的示意图

7. 队列

队列 (Queue) 是一种只允许在一端进行插入, 而在另一端进行删除的特殊线性表。在表中只允许进行插入的一端称为队尾 (Rear), 只允许进行删除的一端称为队头 (Front)。根据队列的定义可知, 队头元素总是最先进队列的, 也总是最先出队列; 队尾元素总是最后进队列, 因而也是最后出队列。可见, 队列是按照先进先出 (First In First Out, FIFO) 的原则组织数据的。

图 1-5 是一个队列的示意图, 通常用指针 $front$ 指示队头元素的前一位置, 用指针 $rear$ 指向队尾元素。

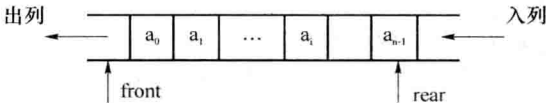


图 1-5 队列示意图

在实际应用中, 队列的顺序存储结构一般采用循环队列的形式, 如图 1-6 所示。入队时, 队尾指针进一, 即 $rear=rear+1$; 出队时, 将队头指针进一, 即 $front=front+1$ 。在队满情况下有: $front=rear$ 。在队空情况下也有: $front=rear$ 。就是说“队满”和“队空”的条件是相同的了。这显然是必须要解决的一个问题。

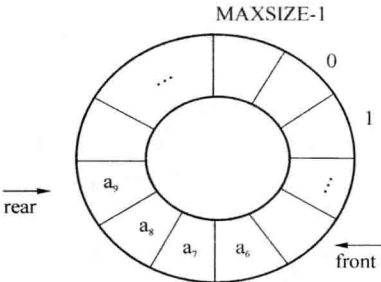


图 1-6 循环队列

8. 二叉树

二叉树 (Binary Tree) 是一种很有用的非线性结构。在二叉树中, 每一个结点的度最大为 2, 即所有子树 (左子树或右子树) 也均为二叉树, 而树结构中的每一个结点的度可以是任意的。另外, 二叉树中的每一个结点的子树被明显地分为左子树与右子树。关于二叉树, 要重点掌握满二叉树、完全二叉树、二叉树的性质、二叉树的链式存储结构和二叉树的遍历方法。

- (1) 满二叉树: 所有分支结点都存在左子树和右子树, 并且所有叶子结点都在同一层上, 如图 1-7(a) 所示。
- (2) 完全二叉树: 叶子结点只能出现在最下层和次下层, 且最下层的叶子结点集中在树的左部, 如图 1-7(b) 所示。显然, 一棵满二叉树必定是一棵完全二叉树, 而完全二叉树未必是满二叉树。

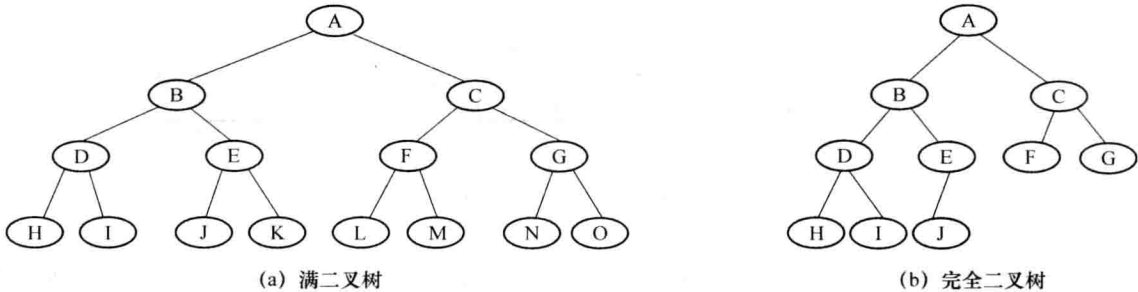


图 1-7 满二叉树和完全二叉树示意图

(3) 对于二叉树的性质,要重点掌握如下4点:

- 一棵非空二叉树的第 i 层上最多有 2^{i-1} 个结点 ($i \geq 1$)。
- 一棵深度为 k 的二叉树中,最多具有 $2^k - 1$ 个结点。
- 对于一棵非空的二叉树,如果叶子结点数为 n_0 ,度数为 2 的结点数为 n_2 ,则有

$$n_0 = n_2 + 1$$

- 具有 n 个结点的完全二叉树的深度 k 为 $\lceil \log_2 n \rceil + 1$ 。

(4) 二叉树通常采用链式存储结构,即用链来指示着元素的逻辑关系。链表中每个结点由 3 个域组成,除了数据域外,还有两个指针域,分别用来给出该结点左孩子和右孩子所在的链结点的存储地址。结点的存储的结构为:

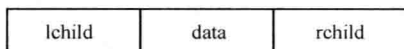


图 1-8 所示为二叉树的二叉链表表示。

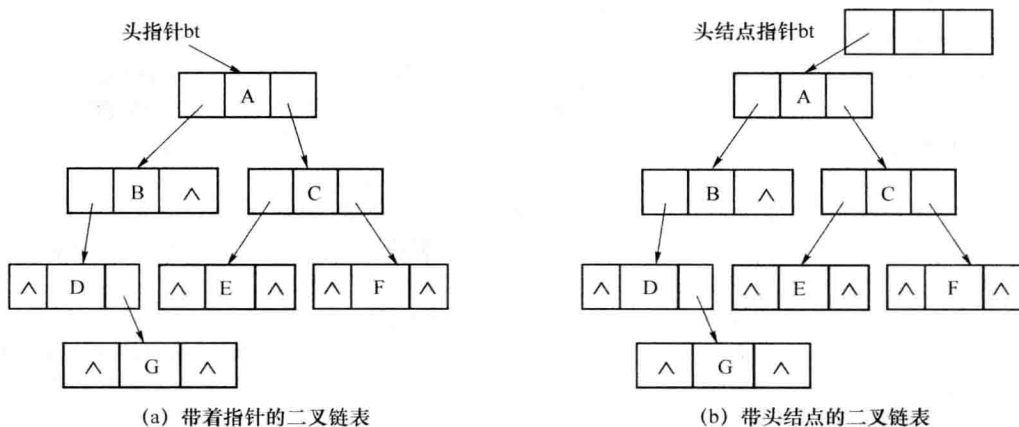


图 1-8 二叉树的二叉链表表示

(5) 二叉树的遍历是指不重复地访问二叉树中的所有结点。在遍历二叉树的过程中,一般先遍历左子树,然后再遍历右子树。在先左后右的原则下,根据访问根结点的次序,二叉树的遍历可以分为 3 种:前序遍历、中序遍历、后续遍历。

3 种遍历的遍历序列如图 1-9 所示。

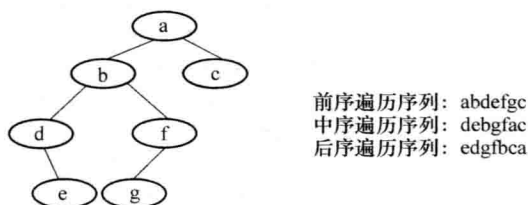


图 1-9 二叉树的遍历序列

9. 查找

查找是指在一个给定的数据结构中查找某个指定的元素。在最近几次考试中很少会考查到查找算法,了解顺序查找和二分查找的主要思路即可。

顺序查找的思路是:从表中的第 1 个元素开始,将给定的值与表中逐个元素的关键字进行比较,直到两者相符,查到所要找的元素为止。否则就是表中没有要找的元素,查找不成功。

二分法查找又称折半查找,其基本思想是:首先选取表中间位置的记录,将其关键字与给定关键字 k 进行比较,若相等,则查找成功;否则,若 k 值比该关键字值大,则要找的元素一定在表的后半部分(或称右子表),则继续对右子表进行折半查找;若 k 值比该关键字值小,则要找的元素一定在表的前半部分(左子表),同样应继续对左子表进行折半查找。每进行一次比较,要么找到要查找的元素,要么将查找的范围缩小一半。如此递推,直到查找成功或把要查找的范围缩小为空(查找失败)。注意,二分查找只适用于顺序存储的有序表。

10. 排序

排序是指将一个无序序列整理成按值非递减顺序排列的有序序列。在最近几次考试中没有考查过排序算法,了解以下排序算法的主要思路即可。

(1) 简单插入排序的思路是:初始可认为文件中的第 1 个记录已经排好序,然后将第 2 个到第 n 个记录依次插入已排序的记录组成的文件中。在对第 i 个记录 R_i 进行插入时, $R_1, R_2, \dots, R_{i-1}$ 已排序,将记录 R_i 的排序码与已经排好序的排序码从右向左依次比较,找到 R_i 应插入的位置,将该位置以后直到 R_{i-1} 各记录顺序后移,空出该位置让 R_i 插入。

(2) 希尔排序的方法如下:

① 选择一个增量序列 $t_1, t_2, \dots, t_k$, 其中 $t_i > t_j, t_k = 1$ 。

② 按增量序列个数 k , 对序列进行 k 趟排序。

③ 每趟排序, 根据对应的增量 t_i , 将待排序列分割成 t_i 个子序列, 将所有距离为 t_i 倍数的记录放在同一个子序列中, 然后分别对各子序列进行直接插入排序。当增量序列为 1 时, 整个序列作为一个表来处理, 表长度即为整个序列的长度。

希尔排序方法是一个不稳定的排序方法, 其时效分析很难, 关键码的比较次数与记录移动次数依赖于增量序列的选取, 特定情况下可以准确估算出关键码的比较次数和记录的移动次数。

(3) 冒泡排序是通过相邻数据元素的交换逐步将线性表变成有序。其基本过程如下:

首先, 从表头(假设线性表的长度为 n)开始往后扫描线性表, 在扫描过程中逐次比较相邻两个元素的大小。若相邻两个元素中, 前面的元素大于后面的元素, 则将它们互换, 称之为消去了一个逆序。显然, 在扫描过程中, 不断地将两相邻元素中的大者往最后移动, 最后就将线性表中的最大者换到了表的最后, 这也是线性表中最大元素应有的位置。

然后对剩下的 $n-1$ 个待排序记录重复上述过程, 又将一个排序码放于最终位置, 反复进行 $n-1$ 次, 可将 $n-1$ 个排序码对应的记录放至最终位置, 剩下的即为排序码最小的记录, 它在第 1 的位置处。

冒泡排序是稳定的排序, 其时间复杂度为 $O(n^2)$, 空间复杂度是 $O(1)$ 。

(4) 快速排序比冒泡排序法的速度快, 基本思路是: 从 n 个待排序的记录中任取一个记录 K (不妨取第 1 个记录), 将 K 后小于 K 的记录移到 K 前, 而 K 前大于 K 的记录放置于 K 后, 使它前面的记录排序码都不大于它的排序码, 而后面的记录排序码都大于它的排序码, 这样就以 K 为分界线, 把记录分成了两个子表。

然后对前、后两部分待排序记录重复上述过程, 直到所有子表为空, 排序即告完成。

(5) 直接选择排序的基本思想是: 首先从所有 n 个待排序记录中选择排序码最小的记录, 将该记录与第 1 个记录交换, 再从剩下的 $n-1$ 个记录中选出排序码最小的记录与第 2 个记录交换。重复这样的操作直到剩下 2 个记录时, 再从中选出排序码最小的记录和第 $n-1$ 个记录交换。剩下的那 1 个记录肯定是排序码最大的记录, 这样排序即告完成。

直接选择排序是不稳定的排序, 其时间复杂度是 $O(n^2)$, 空间复杂度是 $O(1)$ 。

(6) 堆排序的基本思想是: 设有 n 个元素, 将其按关键码排序, 首先将这 n 个元素按关键码建成堆, 将堆顶元素输出, 得到 n 个元素中关键码最小(或最大)的元素。然后, 再对剩下的 $n-1$ 个元素建成堆, 输出堆顶元素, 得到 n 个元素中关键码次小(或次大)的元素。如此反复, 便得到一个按关键码有序的序列。

真题链接

【试题 1-1】下列叙述中正确的是_____。(2010.3)

- A) 对长度为 n 的有序链表进行查找, 最坏情况下需要的比较次数为 n
- B) 对长度为 n 的有序链表进行对分查找, 最坏情况下需要的比较次数为 $(n/2)$
- C) 对长度为 n 的有序链表进行对分查找, 最坏情况下需要的比较次数为 $(\log_2 n)$
- D) 对长度为 n 的有序链表进行对分查找, 最坏情况下需要的比较次数为 $(n \log_2 n)$

解析: 对于顺序查找, 在最坏的情况下, 查找的是链表的最后一个元素, 或者查找的元素不在表中, 此时需要比较 n 次。对长度为 n 的有序链表进行对分查找, 最坏情况下需要的比较次数为 $\log_2 n$ 。

答案: A

【试题 1-2】下列数据结构中, 属于非线性结构的是_____。(2013.3)

- A) 双向链表
- B) 循环链表
- C) 二叉链表
- D) 循环队列

解析: 对于线性结构, 除了首结点和尾结点外, 每一个结点只有一个前驱结点和一个后继结点。线性表、栈、队列都是线性结构, 循环链表和双向链表是线性表的链式存储结构; 二叉链表是二叉树的存储结构, 而二叉树是非线性结构, 因为二叉树有些结点有两个后继结点, 不符合线性结构的定义。

答案: C

【试题 1-3】在下列链表中, 能够从任意一个结点出发直接访问到所有结点的是_____。(2013.3)

- A) 单链表
- B) 循环链表
- C) 双向链表
- D) 二叉链表

解析: 由于线性单链表的每个结点只有一个指针域, 由这个指针只能找到其后件结点, 但不能找到其前件结点。也就是说, 只能顺着之后向链尾方向进行扫描, 因此必须从头指针开始, 才能访问到所有的结点。循环链表的最后一个结点的指针域指向表头结点, 所有结点的指针构成了一个环状链, 只要指出表中任何一个结点的位置就可以从它出发访问到表中其他所有的结点。双向链表中的每个结点设置有两个指针, 一个指向其前件, 另一个指向其后件, 这样从任意一个结点开始, 既可以向前查找, 也可以向后查找, 在结点的访问过程中一般从当前结点向链尾方向扫描, 如果没有找到, 则从链尾向头结点方向扫描, 这样部分结点就要被遍历两次, 因此不符合题意。二叉链表是二叉树的一种链式存储结构, 每个结点有两个指针域, 分别指向左右子结点, 可见, 二叉链表只能由根结点向叶子结点的方向遍历。

答案: B

【试题 1-4】下列与栈结构有关联的是_____。

- A) 数组的定义域使用
- B) 操作系统的进程调度
- C) 函数的递归调用
- D) 选择结构的执行

解析: 递归调用就是在当前的函数中调用当前的函数并传给相应的参数, 这是一个动作, 这一动作是层层进行的, 直到满足一般情况的时候, 才停止递归调用, 开始从最后一个递归调用返回。函数的调用原则和数据结构栈的实现是相一致的。也说明函数调用是通过栈实现的。

答案: C

1.2 程序设计基础

一、程序设计方法和风格

程序设计方法是研究问题求解和如何进行系统构造的软件方法学。除了好的程序设计方法,程序设计风格同样非常重要。程序设计的风格总体而言应该强调简单和清晰,程序必须是可以理解的,著名的“清晰第一,效率第二”的观点已经成为当今主导的程序设计风格。

要形成良好的程序设计风格,主要应注重和考虑源程序文档化、数据说明、语句结构和输入输出等因素。

二、结构化程序设计

1. 结构化程序设计的原则

结构化程序设计的基本原则包括:自顶向下原则,逐步求精原则,模块化原则和限制使用 goto 语句。

2. 结构化程序的基本结构

程序设计语言的 3 种基本结构分别是:顺序结构、选择结构和循环结构,如图 1-10、图 1-11 和图 1-12 所示。

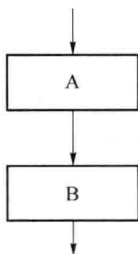


图 1-10 顺序结构

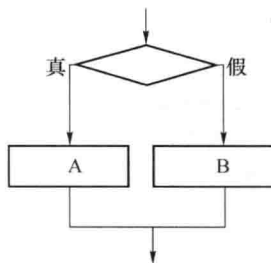
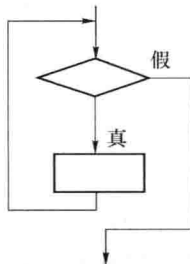
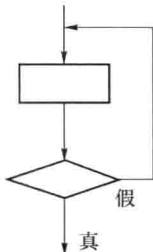


图 1-11 选择结构



(a) 当型循环结构



(b) 直到型循环结构

图 1-12 两种循环结构

三、面向对象的程序设计

面向对象方法的本质就是主张从客观世界固有的事物出发来构造系统,用人类在现实生活中常用的思维方法来认识、理解和描述客观事物。

关于面向对象程序设计需要掌握以下概念:

1. 对象

对象是包含客观事物特征的抽象实体,是属性和行为的封装体。对象包含的信息则称为属性,通常在设计阶段已经确定,通过定义好的对象的操作可以改变属性。

操作的对外过程是封闭的,这个操作是封装在对象内部的,用户看不到,只是通过该对象提供的接口调用,对象的这一特征,称为对象的封装性。

2. 类和实例

类(Class)是具有共同属性、共同方法的一组对象的集合,它为属于该类的全部对象提供了统一的抽象描述,而一个对象是其对应类的一个实例。可以说,类是对象集合的再抽象。

3. 消息

消息(Message)是描述事件发生的信息。消息是实例之间传递的信息,用于请求对象执行某一处理或回答某一要求,它统一了数

据流和控制流。

4. 抽象性

抽象(Abstract)就是忽略事物中与当前目标无关的非本质特征,更充分地注意与当前目标有关的本质特征。从而找出事物的共性,并把具有共性的事物划为一类,得到一个抽象的概念。

5. 继承性

继承性是指特殊类的对象拥有其一般类的属性和行为。继承意味着“自动地拥有”,即特殊类中不必重新定义已在一般类中定义过的属性和行为,而它却自动地、隐含地拥有其一般类的属性与行为。继承允许和鼓励类的重用,提供了一种明确表述共性的方法。一个特殊类既有自己新定义的属性和行为,又有继承下来的属性和行为。

在软件开发过程中,继承性实现了软件模块的可重用性、独立性,缩短了开发周期,提高了软件开发的效率,同时使软件易于维护和修改。

6. 多态性

多态性是指当对不同类的对象执行同样的方法时,系统能根据不同类的对象正确辨别调用各对象所属类的相应方法,从而产生不同的结果。

利用多态性,用户可以发送一般形式的消息,而将所有的实现细节都留给接收消息的对象。多态性不仅增加了面向对象技术的灵活性,进一步减少了信息冗余,而且大大提高了软件的可重用性和可扩展性。

真题链接

【试题 1-5】面向对象方法中,继承是指_____。(2010.9)

- A) 一组对象所具有的相似性质
- B) 一个对象具有另一个对象的性质
- C) 各对象之间的共同性质
- D) 类之间共享属性和操作的机制

解析:面向对象方法中,继承是指新类从已有类那里得到已有的特性。

答案:D

1.3 软件工程基础

一、软件工程基本概念

1. 软件的定义

软件(Software)是计算机系统中与硬件(Hardware)相互依存的另一部分,是包括程序(Program)、数据(Data)及其相关文档(Document)的完整集合。程序是软件开发人员按照用户的需求设计,用程序语言实现的,适合计算机执行的指令序列;数据是使程序能正常操纵信息的数据结构;文档是与程序开发维护和使用有关的各种图文资料。

软件按照功能可以分为应用软件、系统软件和支撑软件(又称为工具软件)。为解决各类实际问题而设计的程序系统称为应用软件,例如:文字处理、表格处理、电子演示、电子邮件收发、绘图软件(AutoCAD)、图像处理软件(Photoshop)等。系统软件是计算机用来管理、控制和维护计算机软、硬件资源,使其充分发挥作用,提高效率,并能使用户可以方便地使用计算机的程序集合,主要包括操作系统、数据库管理系统、网络通信管理程序和其他常用的服务程序等。支撑软件是介于上面两种软件之间,协助用户开发软件的工具性软件,例如需求分析工具软件、设计工具软件、编码工具软件、测试工具软件等。

2. 软件危机

软件危机包含两方面问题:其一,如何开发软件,以满足不断增长,日趋复杂的需求;其二,如何维护数量不断膨胀的软件产品。具体地说,软件危机主要有以下表现:

- 对软件开发成本和进度的估计常常不准确。开发成本超出预算,实际进度比预定计划一再拖延的现象并不罕见。
- 软件需求的增长难以得到满足。用户对系统不满意的现象经常发生。
- 软件产品的质量难以保障。
- 软件难以维护或者可维护程度非常之低。
- 软件的成本不断提高。
- 软件开发生产率的提高赶不上硬件的发展和人们需求的增长。

3. 软件工程

人们针对软件危机的表现和原因,经过不断的实践和总结,越来越认识到:按照工程化的原则和方法组织软件开发工作,是摆脱软件危机的一个主要出路。

1993年,IEEE给出了一个软件工程的定义:将系统化的、规范的、可度量的方法应用于软件的开发、运行和维护的过程,即将工程化应用于软件中。这些主要思想都是强调在软件开发过程中需要应用工程化原则。

软件工程包括3个要素:方法、工具和过程。方法是完成软件工程项目的手段;工具支持软件的开发、管理和文档生成;过程支持软件开发的各个环节的控制、管理。

4. 软件工程过程

软件工程通常包含4种基本活动。

- (1) P(Plan):软件规格说明。规定软件的功能及其运行时的限制。
- (2) D(Do):软件开发。产生满足规格说明的软件。
- (3) C(Check):软件确认。确认软件能够满足用户提出的要求。
- (4) A(Action):软件演进。为满足客户的变更要求,软件必须在使用的过程中演进。

5. 软件生命周期

总体上分析,软件生命周期包括可行性研究和需求分析、设计、实现、测试、交付使用和维护等几个阶段。这几个阶段又可以归纳为3个大的阶段,即定义阶段、开发阶段和维护阶段。如图1-13所示。这些活动可以有重复,执行时也可以有迭代。

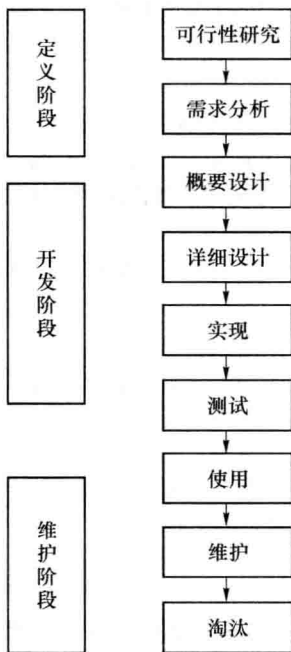


图 1-13 软件生命周期

6. 软件工程的原则

在软件开发过程中必须遵循软件工程的基本原则。这些原则适用于所有的软件项目,包括抽象、信息隐蔽、模块化、局部化、确定性、一致性、完备性和可验证性。

二、结构化分析方法

结构化方法包括已经形成了配套的结构化分析方法、结构化设计方法和结构化编程方法,其核心和基础是结构化程序设计理论。

1. 需求分析

软件需求分析是指用户对目标软件系统在功能、行为、性能、设计约束等方面的期望。需求分析的任务是发现需求、求精、建模和定义需求的过程。需求分析将创建所需的数据模型、功能模型和控制模型。

需求分析阶段的工作可以概括为4个方面:

- (1) 需求获取。需求获取的目的是确定对目标系统的各方面需求。涉及的主要任务包括建立获取用户需求的方法框架,并支持和监控需求获取的过程。
- (2) 需求分析。对获取的需求进行分析和综合,最终给出系统的解决方案和目标系统的逻辑模型。
- (3) 需求编写规格说明书。需求规格说明书作为需求分析的阶段成果,可以为用户、分析人员和设计人员之间的交流提供方便,可以直接支持目标系统的确认,又可以作为控制软件开发过程的依据。
- (4) 需求评审。在需求分析的最后阶段,对需求分析阶段的工作进行复审,验证需求文档的一致性、可行性、有效性。

2. 需求分析方法

常见的需求分析方法有两种。

(1) 结构化分析方法:常见的结构化分析方法包括面向数据流的结构化分析方法(SA-Structured analysis),面向数据结构的 Jackson 方法(Jackson system development method,JSD),面向数据结构的结构化数据系统开发方法(Data structured system development method,DSSD)。

(2) 面向对象的分析方法(Object-Oriented analysis method,OOA):从需求分析建立的模型的特性来分,需求分析又分为静态分析方法和动态分析方法。

3. 结构化分析方法

对于面向数据流的结构化分析方法,按照 DeMarco 的定义,“结构化分析就是使用数据流图(DFD)、数据字典(DD)、结构化英语、判定表和判定树等工具,来建立一种新的、称为规格化说明的目标文档”。

结构化分析方法的实质是着眼于数据流,自顶向下,逐层分解,建立系统的处理流程,以数据流图和数据字典为主要工具,建立系统的逻辑模型。

结构化分析的常用工具如下:

(1) 数据流图:从数据传递和加工的角度,来刻画数据流从输入到输出的移动变化过程。数据流图中的主要图形元素说明如下:

 加工(转化)。表示接收输入,经过变化,继而产生输出的处理过程。

 数据流。表示数据的流向和路径,通常在旁边标注数据流名称。

 存储文件。表示处理过程中存放各种数据的文件。

 表示外部实体,代表数据源和数据池。

(2) 数据字典:是用来定义数据流图中各个成分具体含义的,它以一种准确的、无二义性的说明方式为系统的分析、设计维护提供了有关元素的一致性定义和详细的描述。数据字典中有四种类型的条目:数据流、数据项、数据存储和加工。

(3) 判定树:先从问题定义的文字描述中分清哪些是判定的条件,哪些是判定的结论,根据描述材料中的连接词找出判定条件之间的从属关系、并列关系、选择关系,根据它们构造判定树。

(4) 判定表:与判定树类似,当数据流图中的加工要依赖多个逻辑条件的取值,即完成该加工的一组动作是由于某一组条件取值的组合而引发的,使用判定表描述比较合适。

4. 需求规格说明书

软件需求规格说明书是作为需求分析的一部分而制定的可交付文件,是需求分析阶段的最后结果。该说明书把在软件计划中所确定的软件范围加以展开,制定出完整的信息描述,详细的功能说明、恰当的检验标准以及其他与需求有关的数据。

软件需求规格说明书的作用是:

① 作为用户和开发人员之间共同的文件,便于用户和开发人员进行理解和交流。

② 反映出用户问题的结构,可以作为软件开发工作的基础和依据。

③ 作为确认测试和验收的依据。

作为设计的基础和验收的依据,软件需求规格说明书应该是精确而无二义性的,需求说明书越精确,则以后出现错误、混淆、反复的可能性越小。用户能看懂需求说明书,并且发现和指出其中的错误是保证软件系统质量的关键,因而需求说明书必须简明易懂,尽量少包含计算机的概念和术语,以便用户和软件人员双方都能接受它。

三、结构化设计方法

1. 软件设计基础

软件设计的基本目标是用比较抽象概括方式确定目标系统如何完成预定任务,即软件设计是确定系统的物理模型。

从技术观点来看,软件设计包括软件结构设计、数据设计、接口设计、过程设计。其中,结构设计是定义软件系统各主要部件之间的关系;软件设计是将分析时创建的模型转化为数据结构的定义;接口设计是描述软件内部、软件和协作系统之间以及软件与人之间如何通信;过程设计则是把系统结构部件转换成软件的过程性描述。

从工程管理角度来看,软件设计分两步完成:概要设计和详细设计。概要设计的基本目标就是回答“总体上说,系统应该如何实现”这个问题。概要设计的任务是设计软件的结构,也就是确定系统由哪些模块组成的,以及这些模块相互间的关系。详细设计的任务是设计每个模块实现的细节及对局部数据进行设计,而不是具体编写程序。详细设计的结果基本决定了最终程序的质量。

2. 软件设计的基本原理

(1) 模块化:指把一个待开发的软件分解成若干个小的简单的部分,软件结构的好坏完全由模块的属性体现出来。模块化的目的是为了降低软件复杂性,使软件设计、测试、调试、维护等工作变得简易。在考虑模块化时,应该避免模块划分过多或者过少。

(2) 抽象:把事物本质的共同特性提取出来而不考虑其他细节。

(3) 信息隐蔽:是指在设计和确定模块时,使得一个模块内包含的信息(过程或数据),对于不需要这些信息的其他模块来说,是不能访问的。

(4) 模块独立性:是通过内聚性和耦合性两个指标来衡量的。

内聚性是一个模块内部包含的信息(过程或数据),对于不需要这些信息的其他模块来说是不能访问的。内聚有如下种类,它们的内聚性由弱到强排列。

耦合性是模块间互相连接的紧密程度的度量。耦合性取决于各个模块之间接口的复杂度、调用方式以及哪些信息通过接口。

对模块进行设计时,应该尽量做到高内聚、低耦合,即减弱模块之间的耦合性和提高模块的内聚性,这样有利于提高模块的独立性。

3. 概要设计的任务

- (1) 设计软件系统结构。
- (2) 数据结构及数据库设计。
- (3) 编写概要设计文档,包括:概要设计说明书、数据库设计说明书、集成测试计划等。
- (4) 概要设计文档评审。

4. 软件结构工具

常用的软件结构工具是结构图,也称程序结构图。使用结构图描述软件系统的层次和分块结构关系,它反映了整个系统的功能实现以及模块与模块之间的联系与通信,是未来程序中的控制层次体系。结构图是用来描述软件结构的图形工具。

模块是用一个矩形表示,矩形内注明模块的功能和名字;箭头表示模块间的调用关系。在结构图中还可以用带注释的箭头表示模块调用过程只能够来回传递的信息。如果希望进一步表明传递的信息是数据还是控制信息,则可用带实心圆的箭头表示传递的是控制信息,用带空心圆的箭头表示传递的是数据。

经常使用的结构图有四种模块类型:传入模块、传出模块、变换模块和协调模块。

5. 面向数据流的设计方法

面向数据流的设计方法定义了一些不同的映射方法,利用这些映射方法可以把数据流图变换成结构图表示的软件结构。首先需要了解数据流图表示的数据处理的类型,然后针对不同类型分别进行分析处理。

结构化设计方法是一种面向数据流的设计方法,它可以与 SA 方法衔接。通常用数据流图(DFD)描述系统中加工和流动的情况。DFD 的数据流可以分为两种类型:变换流和事务流。

6. 软件结构设计的准则

- 改进软件结构,提高模块独立性。
- 模块的规模要适中。
- 模块的作用范围应该保持在模块的控制范围内。
- 软件结构的深度、宽度、扇入、扇出要适当。
- 降低模块接口的复杂度。
- 设计单入口单出口的模块。
- 模块的功能可预测。

7. 详细设计

详细设计的任务是为软件结构图中的每一个模块确定实现算法和局部数据结构,用某种选定的表达工具表示算法和数据结构的细节。

详细设计阶段常见的工具有:程序流程图,N-S 流程图和问题分析图。

(1) 程序流程图(PFD):用方框表示一个处理步骤,菱形代表一个逻辑条件,箭头表示控制流方向。图 1-14 给出了程序流程图的基本结构。

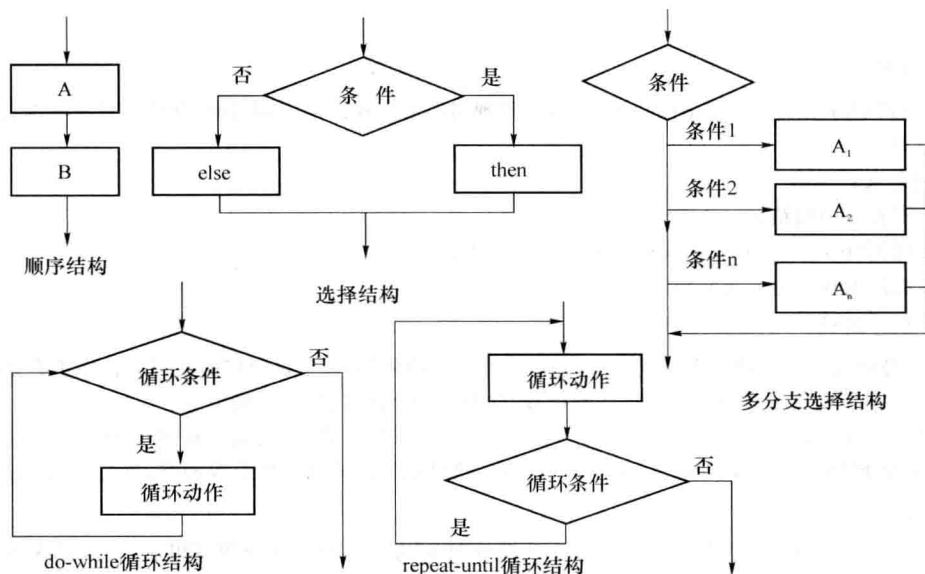


图 1-14 程序流程图的基本结构