



教育部大学计算机课程改革项目规划教材

C 语言程序设计

主 编 吴宏瑜
副主编 孙亚飞 陈杰华

高等教育出版社



教育部大学计算机课程改革项目规划教材

C 语言程序设计

C Yuyan Chengxu Sheji

主 编 吴宏瑜
副主编 孙亚飞 陈杰华

高等教育出版社·北京

内容提要

“C 语言程序设计”是高校非计算机专业本科生的一门重要的公共技术基础课,是多门后续专业课的基础。本书旨在培养学生熟练使用 C 语言编程分析和解决实际问题的能力,为进一步学习其他专业课程和今后从事软件开发工作打下坚实的基础。

本书共 9 章,主要包括算法与 C 语言概述,数据类型及表达式, C 语言的语句及程序流程控制,数组,函数,编译预处理,指针,结构体、共用体与枚举类型,文件。为便于教学,随书提供电子教案、程序调试过程微视频、习题参考答案、示例程序源代码等教学资源。本书可作为高校非计算机专业本科生的程序设计课程教材,也可供软件开发技术人员参考。

图书在版编目 (C I P) 数据

C 语言程序设计 / 吴宏瑜主编.--北京:高等教育出版社,2016.2

ISBN 978-7-04-044904-4

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

中国版本图书馆 CIP 数据核字 (2016) 第 024737 号

策划编辑 刘 茜 责任编辑 倪文慧 封面设计 张 志 版式设计 杜微言
插图绘制 杜晓丹 责任校对 刘娟娟 责任印制 朱学忠

出版发行	高等教育出版社	网 址	http://www.hep.edu.cn
社 址	北京市西城区德外大街 4 号		http://www.hep.com.cn
邮政编码	100120	网上订购	http://www.hepmall.com.cn
印 刷	北京鑫海金澳胶印有限公司		http://www.hepmall.com
开 本	850mm×1168mm 1/16		http://www.hepmall.cn
印 张	19.25	版 次	2016 年 2 月第 1 版
字 数	440 千字	印 次	2016 年 2 月第 1 次印刷
购书热线	010-58581118	定 价	32.00 元
咨询电话	400-810-0598		

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换

版权所有 侵权必究

物 料 号 44904-00

数字课程资源使用说明

与本书配套的数字课程资源发布在高等教育出版社易课程网站,请登录网站后开始课程学习。

一、网站登录

1. 访问 <http://abook.hep.com.cn/1852130>,单击“注册”按钮。在注册页面输入用户名、密码及常用的邮箱进行注册。已注册的用户直接输入用户名和密码登录即可进入“我的课程”界面。

2. 课程充值:登录后单击右上方“充值”图标,正确输入教材封底标签上的明码和密码,单击“确定”按钮完成课程充值。

3. 在“我的课程”列表中选择已充值的数字课程,单击“进入课程”即可开始课程学习。

账号自登录之日起一年内有效,过期作废。使用本账号如有任何问题,请发邮件至:
ecourse@pub.hep.cn。



易课程
course

C 语言程序设计

吴宏编

用户名

密码

验证码

进入课程

注册

重要通知

数字课程介绍

纸质教材

版权信息

联系方式

书名: 《C语言程序设计》

作者: 吴宏编

ISBN:

出版商: 高等教育出版社

出版日期: 2016年2月

定价: 元



“C语言程序设计”是高校非计算机专业本科生的一门重要的公共技术基础课,是多门后续专业课的基础。本书旨在培养学生熟练使用C语言编程分析和解决实际问题的能力,为进一步学习其他专业课

因系统升级,所有用户都需要先注册(不能用书后的明码密码直接登录)。注册后的用户登录后,需先点击页面右上方“充值”,正确输入教材封底标签上的明码和密码完成课程选择。

注册

登录

充值

在“我的课程”列表中选择已充值的数字课程,点击“进入课程”即可浏览或下载

二、资源使用

与本书配套的易课程数字课程资源按照章、节知识树的形式构成,包括电子教案、微视频、程序源代码、习题参考解答等内容的资源,以便读者学习使用。

1. 电子教案:与课程和教材紧密配套的教学 PPT,可供教师上课使用,也可供学生课前预习或课后复习使用。

2. 微视频:书中部分示例进行调试过程的录屏演示,这些微视频以二维码的形式标注在书的边栏,扫描后即可观看,供学生了解程序的运行环境与调试过程。

3. 程序源代码:书中所列出的部分示例程序源代码,可供学生下载使用,以便完成程序的调试。

4. 习题参考解答:书中各章作业的参考解答,可供学生随时检验自己对课堂知识的把握情况。

。前言

“C 语言程序设计”是面向高校非计算机专业本科生开设的一门重要的公共技术基础课,是多门后续专业课程的基础。本书旨在帮助学生掌握 C 语言的基本语法、语句、控制结构以及结构化程序设计的基本思想和方法,使学生认识到算法、良好的程序设计风格以及上机实践在学习中的重要性,培养学生熟练使用 C 语言编程分析和解决实际问题的能力,使之无论以后在学习、工作中使用什么语言编程都能灵活地应用这些思想和方法,为进一步学习其他专业课程和今后从事软件开发工作打下坚实的基础。

在教材编写中作者力求知识准确、文字亲切、示例有趣、内容实用。因为准确才能让学习者掌握,亲切才会被学习者喜欢,有趣才能给学习者留下深刻的印象,实用才会对学习者的真正起到帮助的作用。本书主要有以下特点:

(1) 采用启发式写作风格。讲解力求通俗易懂,算法循序渐进,难点化整为零,用类比和直观的图示讲解概念和难点,如 C 语言的数据类型、常量和变量、预编译机制、运算符和表达式、数据的输入和输出、选择语句结构、循环语句结构、选择的嵌套、循环的嵌套、选择和循环的中断、goto 语句等。

(2) 以经典算法为背景,兼顾趣味性和实用性。所有程序实例都有很强的应用背景,如体型判断、身高预测、猜数游戏等,激发学生的学习兴趣,使学生感受到“学在其中、乐在其中、用在其中”。

(3) 以实例引入,不单纯讲语法,更注重能力培养和思维训练。部分实例提供“一题多解”,用多种算法和多种编程带动学生去探索更多、更好的实现方法,提高学生分析和解决问题的能力。

(4) 采取循序渐进的程序编制方法。算法由浅入深,从简单且规模较小的程序开始编起,逐渐过渡到复杂且规模较大的程序,如函数的参数传递、数组、指针是整个 C 语言学习中的重点和难点。另外,结构体和数据链表也非常重要,尤其是结构体,在编程开发中更是重中之重。

(5) 有机地贯穿算法设计、数据结构、程序设计方法和软件工程思想等内容,将所有程序都按统一的编码规范进行书写,使学生自然而然地养成良好的程序设计风格。

(6) 提供配套的教学参考书《C 语言程序设计上机实践教程》(孙亚飞主编、高等教育出版社出版)及电子教案、程序调试微视频、程序源代码、习题参考解答等教学资源,供教师和学生使用。

本书由吴宏瑜担任主编,孙亚飞、陈杰华担任副主编,其中第 1 章和第 9 章由陈杰华编写,第 2 章和第 5 章由张宇编写,第 3 章和附录由吴宏瑜编写,第 4 章和第 6 章由赵伟

庆编写,第7章由孙亚飞编写,第8章由葛龙编写,夏欣、戴丽娟、郭新明、孟宏源、李霓、张珏、伍琳、张玲等参与本书部分程序的验证和素材制作等工作。全书由吴宏瑜负责统稿。

由于时间仓促,加之编者水平有限,书中难免有不足之处,恳请读者批评指正。作者的邮件地址为 wuhongyu@scu.edu.cn。

编者

2016年1月

目 录

第 1 章 算法与 C 语言概述

1.1 程序设计的基本概念	002	1.3.1 C 语言的发展及特点	016
1.1.1 程序设计的基本步骤与 程序设计语言	002	1.3.2 C 语言程序的构成及程序的 书写格式	017
1.1.2 程序设计方法	005	1.3.3 程序调试步骤	020
1.2 算法	009	1.3.4 用 VC 系统实现 C 程序的 操作过程	020
1.2.1 算法概述	009	本章小结	023
1.2.2 算法表示	010	习题 1	023
1.2.3 算法示例	014		
1.3 C 语言概述	016		

第 2 章 数据类型及表达式

2.1 标识符与关键字	026	2.4.2 赋值运算符与赋值表达式	034
2.2 数据类型	026	2.4.3 自增、自减运算符	035
2.3 常量与变量	027	2.4.4 逗号运算符与逗号表达式	036
2.3.1 常量	027	2.4.5 强制类型转换	037
2.3.2 变量	029	2.4.6 位运算	039
2.4 运算符与表达式	032	本章小结	041
2.4.1 算术运算符与算术表达式	032	习题 2	041

第 3 章 C 语言的语句及程序流程控制

3.1 C 语言语句概述	046	基本结构	050
3.1.1 表达式语句	046	3.2.2 基本的输入/输出函数	051
3.1.2 函数调用语句	048	3.2.3 顺序结构程序设计应用实例	059
3.1.3 控制语句	048	3.3 选择结构程序设计	061
3.1.4 复合语句	048	3.3.1 关系运算符与关系运算 表达式	061
3.1.5 空语句	049	3.3.2 逻辑运算符与逻辑运算 表达式	062
3.2 顺序结构程序设计	050		
3.2.1 顺序结构程序设计的			

3.3.3 if 语句	063	3.4.4 for 语句	080
3.3.4 条件运算符	068	3.4.5 循环的嵌套	085
3.3.5 switch 开关语句	069	3.4.6 几种循环语句的比较	090
3.3.6 程序举例	072	3.4.7 break 和 continue 语句	092
3.4 循环控制语句	074	3.5 程序举例	096
3.4.1 goto 语句	075	本章小结	103
3.4.2 while 语句	075	习题 3	104
3.4.3 do while 语句	078		

第 4 章 数 组

4.1 一维数组	118	4.2.4 二维数组应用实例	126
4.1.1 一维数组的定义	118	4.3 字符数组与字符串	128
4.1.2 一维数组的初始化	119	4.3.1 字符数组的定义	128
4.1.3 一维数组元素的引用与地址 引用	120	4.3.2 字符数组的初始化	128
4.1.4 一维数组应用实例	121	4.3.3 字符数组的引用	129
4.2 二维数组	123	4.3.4 字符串和字符串结束标志	129
4.2.1 二维数组的定义	123	4.3.5 字符串的输入/输出	130
4.2.2 二维数组的初始化	124	4.3.6 字符串常用函数	131
4.2.3 二维数组元素的引用与 地址引用	125	4.3.7 字符数组应用实例	134
		本章小结	136
		习题 4	137

第 5 章 函 数

5.1 函数的概述	144	5.5.3 函数的嵌套调用	156
5.2 函数的定义	144	5.5.4 函数的递归调用	158
5.2.1 无参数函数的定义	144	5.6 变量的作用域	161
5.2.2 有参数函数的定义	145	5.6.1 局部变量	162
5.3 函数间的数据传递	147	5.6.2 全局变量	163
5.3.1 实际参数和形式参数	147	5.7 变量的存储类别	164
5.3.2 数组作为参数	149	5.7.1 自动变量	164
5.4 函数的返回值	152	5.7.2 静态变量	165
5.4.1 无返回值的函数	152	5.7.3 寄存器变量	166
5.4.2 有返回值的函数	153	5.7.4 外部变量	166
5.5 函数的调用	153	5.7.5 内部函数和外部函数	168
5.5.1 函数的调用方式	154	本章小结	169
5.5.2 函数的原型声明	154	习题 5	170

第6章 编译预处理

6.1 宏定义	176	6.3 条件编译	180
6.1.1 无参数的宏定义	176	本章小结	182
6.1.2 带参数的宏定义	177	习题 6	182
6.2 文件包含处理	179		

第7章 指 针

7.1 指针的概念与使用	186	7.4.4 指向数组的指针变量 作函数的参数	201
7.1.1 指针的概念	186	7.5 指针的运算	204
7.1.2 指针变量的定义	187	7.5.1 指针变量的赋值运算	204
7.2 指针变量的引用	187	7.5.2 指针变量的 * 运算	204
7.3 指针与函数	191	7.5.3 指针变量的加减运算	204
7.3.1 指针作为函数的参数	191	7.6 指针数组	207
7.3.2 返回指针的函数	193	7.6.1 指针数组的说明及使用	207
7.3.3 指向函数的指针变量	194	7.6.2 指针变量的指针	208
7.4 指向数组的指针变量	195	本章小结	209
7.4.1 指向一维数组的指针	195	习题 7	211
7.4.2 指向多维数组的指针	197		
7.4.3 指向字符数组的指针	199		

第8章 结构体、共用体与枚举类型

8.1 结构体的定义和引用	216	8.3.2 指向结构体类型数据的 指针变量的初始化	232
8.1.1 结构体的定义	216	8.3.3 通过指向结构体变量的指针 访问结构体成员	233
8.1.2 结构体变量的定义	218	8.3.4 指向结构体数组的指针	234
8.1.3 结构体变量的初始化	220	8.3.5 动态存储分配指针所指的 空间	235
8.1.4 结构体变量的引用	221	8.3.6 结构体指针应用实例	240
8.1.5 结构体的嵌套定义	221	8.4 结构体和函数	243
8.1.6 结构体变量应用实例	223	8.4.1 结构体作为函数参数和 返回值	243
8.2 结构体数组	225	8.4.2 结构体指针作为函数参数和 返回值	245
8.2.1 结构体数组的定义与初始化	226	8.5 链表	247
8.2.2 结构体数组的引用	228		
8.2.3 结构体数组应用实例	229		
8.3 指向结构体类型数据的指针 ...	232		
8.3.1 指向结构体类型数据的 指针变量的定义	232		

8.5.1 链表的概念	247	8.6.2 共用体变量的引用	260
8.5.2 建立简单链表	248	8.6.3 共用体类型数据的特点	260
8.5.3 建立动态链表	249	8.7 枚举类型	261
8.5.4 输出链表	251	8.7.1 枚举类型的定义	261
8.5.5 链表的插入操作	253	8.7.2 枚举类型变量的赋值和使用	262
8.5.6 链表的删除操作	256	8.8 用 typedef 定义新的类型名 ...	266
8.6 共用体	259	本章小结	267
8.6.1 共用体类型的说明和 变量的定义	259	习题 8	268

第 9 章 文 件

9.1 文件的概述	272	9.4.3 数据块输入/输出函数 fread() 和 fwrite()	283
9.1.1 引言	272	9.4.4 格式化输入/输出函数 fscanf() 和 fprintf()	285
9.1.2 文件分类	272	9.5 文件的随机存取	287
9.2 文件类型的指针	274	9.5.1 文件定位	288
9.2.1 引言	274	9.5.2 文件的随机写入	290
9.2.2 文件类型 FILE	274	9.5.3 文件的随机读取	291
9.3 文件的打开与关闭	275	9.6 文件的检测	292
9.3.1 文件打开函数 fopen()	275	9.6.1 文件检测函数	292
9.3.2 文件关闭函数 fclose() 和 fcloseall()	276	9.6.2 文件检测应用实例	292
9.4 文件的顺序读写	277	9.7 程序实例	294
9.4.1 字符输入/输出函数 fgetc() 和 fputc()	277	本章小结	295
9.4.2 字符串输入/输出函数 fgets() 和 fputs()	281	习题 9	296

参考文献	297
-------------------	-----

普通高等教育“十一五”
规划教材·计算机、机械
、自动化专业计算机类

第1章 算法与C语言概述

· 计算机的发展· 1
· 计算机系统的组成· 2
· 计算机系统的层次结构· 3
· 计算机系统的组成· 4
· 计算机系统的组成· 5
· 计算机系统的组成· 6

· 计算机系统的组成· 7
· 计算机系统的组成· 8
· 计算机系统的组成· 9
· 计算机系统的组成· 10

· 计算机系统的组成· 11
· 计算机系统的组成· 12

· 计算机系统的组成· 13
· 计算机系统的组成· 14

计算机求解问题的方式就是算法,任何计算任务都会被分解成算法中的若干操作步骤来实现。本章将介绍一些常见的算法,如交换两个变量的值、取绝对值、累加、累乘、求最大公约数、求斐波那契数列、判断质数等。另外,本章还将介绍程序设计的基本概念和C程序的构成。

1.1 程序设计的基本概念

1.1.1 程序设计的基本步骤与程序设计语言

1. 程序与计算机程序

程序的概念自古有之,过去用于表达完成一件事情的操作过程,如一次开会议程、一趟旅行安排、一个宗教仪式等。简言之,程序就是对一系列操作过程的描述。

那么,什么是计算机程序?这种程序与现实生活中的程序有什么相同之处、又有什么不同之处?具体而言,如何借助计算机来求解一个一元二次方程的两个解?

用户使用计算机,就是要利用计算机处理现实中的各种问题。但是,计算机只是一种计算工具,本身并没有自行处理问题的能力,如同算盘一样,都需要人进行控制才能处理问题。而计算机与算盘的工作方式完全不同,其处理问题的过程需要用户写出代码形式的操作步骤。下面来看一下计算机是如何求解一元二次方程的。

【例 1.1】 求解一元二次方程。

用计算机求解一元二次方程的操作步骤如下。

步骤 1:为计算机提供一元二次方程的 3 个系数。

步骤 2:用数学公式计算方程的第一个解。

步骤 3:用数学公式计算方程的第二个解。

步骤 4:输出方程的两个解。

步骤 5:程序结束。

下面来观察一下计算机程序的执行过程。计算机在执行一项用户安排给它的计算任务时,将完全按照用户指定的操作指令去做,比如用户的操作指令是“相加”,它不能进行“相减”,更不能计算错误。在计算机中,操作指令对应着计算机能够执行的一个基本动作。为求解一个计算任务,用户会告诉计算机按照特定的操作顺序完成一系列指令,这一系列指令的集合就是计算机程序。

2. 程序设计的基本步骤

在用计算机求解一个计算任务时,需要描述解决计算任务的过程,这是软件构造活动中的重要组成部分之一。程序设计就是以一种程序设计语言为描述工具,给出相应语句序列的过程。

基于求解问题本身的复杂性和构造程序的特点,通常用户不可能直接编写程序,而是要将其分解成若干步骤。程序设计的基本步骤包括分析问题、设计算法、编写程序、运行程序、分析结果、编写文档等。

(1) 分析问题

对于计算任务要进行认真分析,研究已知条件,分析最后应得到的结果,找出求解问题的规律,选择合理的解题思想。

(2) 设计算法

设计解题过程的具体方法和相关操作步骤。

(3) 编写程序

根据实际情况选择合适的程序设计语言,按算法要求编写计算机程序,对源程序进行编辑、编译、连接、修改等。

(4) 运行程序

使用特定的计算机执行系统来运行程序,其中需要提供合理的输入数据,并最终得到正确的运行结果。

(5) 分析结果

通过运行程序得到运行结果后,并不一定表示程序完全正确。所以,需要对结果进行分析,检查是否正确。若不正确,则要对程序进行修改并调试。

(6) 编写文档

由于程序最终都是要提供给他人使用的,因此,程序开发者在提供给用户所用程序的同时,还必须向用户提供说明书和用户手册,主要内容包括程序功能、运行环境要求、程序的安装与启动过程、输入数据要求、使用注意事项等。另外,由于程序调试和维护需要,也应该在程序清单中添加许多注释方面的信息。

3. 程序设计语言

自 20 世纪 50 年代末期以来,计算机行业内出现了数千种程序设计语言,但目前只有数十种语言得到了广泛应用。从历史发展历程来看,程序设计语言可以分为 4 代。

1) 第一代——机器语言

机器语言是由二进制数码 0、1 构成的指令。不同的计算机系统具有不同的指令系统,如 PC 中表示相加运算的指令代码是 01001。

机器语言中的指令可以由计算机识别和执行,运行速度快,且可以实现对硬件的控制,但使用机器语言编写程序非常困难,程序的修改与维护也极不方便,所以编程效率极低,目前这种语言已经逐渐被其他语言代替了。

2) 第二代——汇编语言

汇编语言中的语句是机器指令的符号化形式,即使用英文缩写词和十进制数据来取代二进制形式的机器指令,如指令代码 01001 可改写成 ADD,二进制数据 1001 可改写成 9。汇编语言的语句与机器语言的指令存在着直接的对应关系,所以汇编语言也同样存在难学、难用、易出错、维护困难等缺点。但是,汇编语言也有其优点,如可以直接访问硬件、比机器语言的编程效率高等。不过,计算机不能直接识别汇编语言中的语句,所以需要翻译成二进制形式的机器指令。这种翻译一般称为“汇编”,所用软件称为“汇编程序”,如图 1-1 所示。



图 1-1 汇编过程

3) 第三代——高级语言

高级语言是面向用户的、独立于计算机硬件的编程语言。其主要优点是书写形式接近于数学语言和自然语言,思维形式与传统人类求解问题的方法比较接近。高级语言中的一条语句可以代替数条、数十条甚至数百条汇编语言中的指令。所以,高级语言通用性强,应用广泛。但是计算机不能直接识别高级语言中的语句,所以需要翻译成二进制形式的机器指令。这种翻译一般分为“编译”和“解释”两种方式,所用软件称为“编译程序”和“解释程序”,如图 1-2 所示。

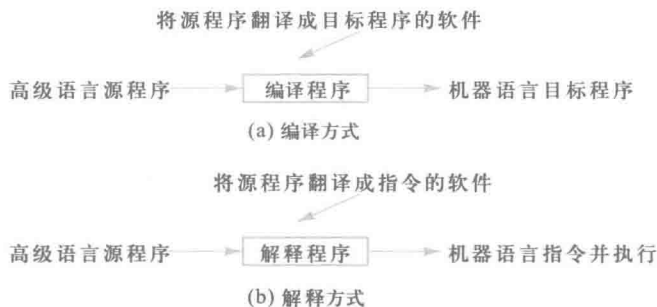


图 1-2 两种翻译方式

如图 1-2 所示,编译是把源程序中的每条语句都编译成机器语言后,保存为二进制的目标文件,这样运行时计算机可以直接以机器语言来运行此程序,运行速度很快;而解释则是只在执行程序时,才一条一条地解释成机器语言并由计算机来执行,所以运行速度较慢。

高级语言种类繁多,可以从应用角度和描述客观系统两个方面进行分类。

从应用角度来看,高级语言可以分为 3 类:基础语言、专用语言和结构化语言。

(1) 基础语言

基础语言也称为通用语言。这种高级语言历史悠久,目前有大量的软件库,为众多用户所熟悉和使用。属于这类语言的有 FORTRAN、BASIC、COBOL、ALGOL 等。其中,BASIC 语言是 20 世纪 50 年代末期为适应分时系统而研制的一种人机交互式的程序设计语言,可用于一般的程序设计教学、数值计算、事务处理等。

(2) 专用语言

专用语言是为某种特殊应用而专门设计的语言,通常具有特殊的应用需求和语法形式。一般而言,这种语言的应用范围狭窄,移植性和可维护性都较差。属于这类语言的有 LISP、APL、Forth 等。其中,Forth 语言是 20 世纪 60 年代出现的基于堆栈、人机交互等功能的编程语言,主要应用于有数千行代码规模的嵌入式系统。该语言被广泛应用于军事、航天、航空、工业自动化、工作站、图形、仪器仪表、天文等领域,目前已研制出以 Forth 为体系结构的处理器芯片。

(3) 结构化语言

从 20 世纪 70 年代初期开始,结构化程序设计思想就日益为人们所接受和使用,其间先后出现了许多结构化语言。这些结构化语言支持结构化的控制结构,具有完备的过程结构和数据结构的描述能力,属于这类语言的有 Pascal、C、Ada 等。其中,C 语言具有

使用方便、应用面广、功能强大、描述能力强、运算符与数据类型丰富、易移植、编译质量高等优点。同时,C 语言还具有机器语言的许多优点,如允许直接访问物理地址、能进行二进制位操作、可直接访问硬件等。

从描述客观系统来看,程序设计语言可以分为两类:面向过程语言和面向对象语言。

(1) 面向过程语言

面向过程语言是以“程序=数据结构+算法”的程序设计范式构成的编程语言。属于这类语言的有 BASIC、FORTRAN、C、Pascal 等。

面向过程的核心是自顶向下、逐步求精的功能分解方法。在进行软件设计和编码时,一般需要进行如下 4 个步骤:将实际应用问题分解成若干个称为模块的功能块;根据指定的模块功能来设计一系列用于存储数据的数据结构;绘制多个求解模块功能的算法框图;编写对这些数据结构进行操作的过程、函数、子程序等。最终的程序就是由这些过程、函数和子程序构成的。显然,这种方法是将数据结构和过程实现作为两个实体分别对待的,其重点在于过程实现,而不是对数据结构进行描述。程序员在设计程序时,首先要考虑的是如何进行功能分解,在每一个过程实现中着重安排程序的操作序列;另一方面,由于操作是作用于特定的数据结构上,因此程序员在编程时必须经常考虑相应的数据结构。

(2) 面向对象语言

面向对象语言是以“程序=对象+消息”的程序设计范式构成的编程语言。属于这类语言的有 Visual Basic、C++、Java、Delphi 等。

面向对象程序设计技术通过增加软件的可扩充性和可重用性来提高程序员的编程能力,它的最大优点是软件具有可重用性,更接近人的思维活动。人们利用面向对象思想进行程序设计时,可以很大程度地提高编程效率,减少软件维护的开销。

4) 第四代——非过程化语言

使用非过程化语言编程时,只需告诉计算机“做什么”而不是“怎么做”,即不需要描述算法实现的细节,其中两个典型应用是数据库查询和应用程序生成器。属于这类语言的有 System Z、PowerBuilder、FOCUS 等。其中,PowerBuilder 语言是 Sybase 公司研发的一种新型快速开发工具,使用客户机/服务器结构,基于 Windows 环境的集成化开发工具。主要特点包括:有可视化与多特性的开发工具、使用面向对象技术、支持高效的复杂应用程序、提供数据库的连接能力、查询、报表和图形功能等。

1.1.2 程序设计方法

本小节介绍两种程序设计方法,即结构化程序设计方法和面向对象程序设计方法。

1. 结构化程序设计方法

结构化程序设计是进行以模块功能和处理过程设计为主的详细设计的基本原则,其概念由 E.W.Dijkstra 于 1965 年首次提出,是软件工程领域的一个重要里程碑。它的主要思想是采用自顶向下、逐步求精的功能分解方法,只使用 3 种基本控制结构来构造程序,即任何程序均由顺序结构、选择结构和循环结构来构造。

1) 结构化程序设计原则

结构化程序设计原则包括自顶向下、逐步细化、模块化和限制使用 goto 语句。

(1) 自顶向下

在编程时,绝对不能直接书写代码,而是应该首先考虑总体结构,然后考虑实现细节。换言之,首先考虑整体目标,然后考虑局部目标,不要一开始就过多追求细节,而是要首先从最上层整体目标开始进行设计,然后逐步将计算任务的操作过程具体化。正如写文章时需要确定章节划分一样。

(2) 逐步细化

在处理复杂问题时,要将其先分解成若干子问题,然后再分解成若干小问题,直到将复杂问题转换成可以处理的简单问题为止,这就是逐步细化。

(3) 模块化

众所周知,任何复杂问题都是由一系列较简单的问题构成的。模块化就是将要解决的总目标分解为若干子目标,再将子目标进一步分解为具体的小目标,这些小目标就称为模块。

(4) 限制使用 goto 语句

由于 goto 语句可以直接改变程序的运行过程,容易造成程序的逻辑混乱,与人类的线性思维方式相冲突,因此在结构化程序设计中限制使用 goto 语句。取消或限制使用 goto 语句后,程序易于理解、易于排错、易于维护。

2) 3 种基本结构

结构化程序由 3 种基本结构构成:顺序结构、选择结构和循环结构。

(1) 顺序结构

顺序结构表示程序中的各操作是按照语句出现的先后顺序执行的,如常见的输入—计算—输出的程序就是顺序结构。例如,求解一个一元二次方程时,首先输入 3 个系数,然后用数学公式计算得到两个解,最后输出这两个解。当然,大多数程序不会如此简单,通常情况是顺序结构作为程序的一部分,与其他结构一起构成一个复杂的程序,例如选择结构中的复合语句、循环结构中的循环体等。

(2) 选择结构

顺序结构的程序虽然能够解决输入、计算、输出等问题,但不能做出判断并进行选择,对于要先做判断后进行选择的问题就要使用选择结构。选择结构的执行是依据指定条件选择执行路径,而不是严格按照语句出现的物理顺序。关键在于构造合适的分支条件和分析程序流程,根据不同的程序流程选择适当的选择语句。

选择结构一般分为 3 种形式:单分支选择、双分支选择和多分支选择。

(3) 循环结构

循环结构表示程序要重复执行某些操作,直到某条件为假(或为真)时才可以终止循环。循环结构可以减少源程序重复书写的工作量。在循环结构中最主要的是哪些操作需要循环执行,什么情况下执行循环,什么情况下退出循环。

循环结构分为两种形式:当型循环和直到型循环。

① 当型循环:表示先进行条件判断,当满足给定的条件后才执行循环体,并且在循环终端处流程自动返回到循环入口处;如果条件不满足,则退出循环体,直接到达循环出口处。因为是“当条件满足时执行循环体”,即先判断后循环,所以称为当型循环。

② 直到型循环:表示从循环入口处直接执行循环体,在循环终端处判断条件。如果