



普通高等教育“十一五”国家级规划教材

高等学校计算机科学与技术系列教材

程序设计基础

张长海 陈娟 刘磊 编著



高等教育出版社
Higher Education Press

普通高等教育“十一五”国家级规划教材
高等学校计算机科学与技术系列教材

程序设计基础

张长海 陈 娟 刘 磊 编著



高等教育出版社

内容提要

本书以 C 语言为载体,引进 PAD 表示程序逻辑,阐述基本的程序设计方法。全书共分 14 章,主要内容包括算法、程序设计方法、函数、数据组织、指针、程序开发、结构化程序设计。每章都包含大量例题和习题,以利于读者提高程序设计能力。

本书最大的特点是以“程序设计”为主线,自始至终在讲“程序设计而不是讲“语言”,摒弃了目前各种程序设计书中流行的以“解释程序设计语言”为主的做法。全书以构造算法为基础,全部为“讲述程序设计”,而不是“解释语言概念”。其目的是教会读者怎样编写程序,提高读者的程序设计能力,彻底改变“学了程序设计而不会编程序”的现状。本书整体结构好,图文并茂,力求体现“结构化程序设计”思想,注重培养和训练读者良好的程序设计风格。

本书可作为高等院校计算机及相关各专业“高级语言程序设计”、“程序设计基础”、“C 语言程序设计”等课程的教材和参考书,也可供从事计算机工作的有关人员参考。

图书在版编目(CIP)数据

程序设计基础/张长海,陈娟,刘磊编著. —北京:高等教育出版社,2008.5

ISBN 978 - 7 - 04 - 024115 - 0

I. 程… II. ①张…②陈…③刘… III. 程序设计 - 高等学校 - 教材 IV. TP311.1

中国版本图书馆CIP数据核字(2008)第056769号

策划编辑 武林晓 责任编辑 康兆华 封面设计 于文燕 责任绘图 尹莉
版式设计 马敬茹 责任校对 殷然 责任印制 朱学忠

出版发行 高等教育出版社
社 址 北京市西城区德外大街4号
邮政编码 100120
总 机 010-58581000

经 销 蓝色畅想图书发行有限公司
印 刷 北京明月印务有限责任公司

开 本 787×1092 1/16
印 张 23
字 数 520 000

购书热线 010-58581118
免费咨询 800-810-0598
网 址 <http://www.hep.edu.cn>
<http://www.hep.com.cn>
网上订购 <http://www.landraco.com>
<http://www.landraco.com.cn>
畅想教育 <http://www.widedu.com>

版 次 2008年7月第1版
印 次 2008年7月第1次印刷
定 价 28.70元

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 24115-00

前 言

本书应用于“高级语言程序设计”、“程序设计基础”、“C 语言程序设计”等课程。学好这类课程既可以为后续课程打下良好的基础,又会对学生一生的程序设计技术、技巧、风格和习惯产生深远影响。

随着计算机技术的不断发展和计算机应用的日渐普及,人们对于大学第一门程序设计课程的认识也在不断更新。

最早,在 20 世纪 70 年代及以前,计算机应用范围十分狭窄,所谓“搞计算机”的人也很少。那时的计算机专业是在研究计算机本身,相应的第一门程序设计课程称为“算法语言”。学习这门课程的目的是学习程序设计语言本身(绝大部分学校都选讲 ALGOL₆₀),而对于如何使用这种语言不太关心。因为那时的教学目的是让学生研究计算机本身,是要制造计算机,是要透彻地研究语言本身的语法和语义,并实现它。

时间已经进入 21 世纪,今天计算机已经渗透到各个领域,甚至已经普及到家庭,它像电视机、冰箱一样,是家用电器之一。计算机专业的规模与 30 年前大不一样。据 2005 年统计,全国开设“计算机科学与技术”专业的院校有 741 所,“计算机科学与技术”专业在校生人数超过 45 万;目前保守估计,全国开设“计算机科学与技术”专业的院校超过 800 所,“计算机科学与技术”专业在校生人数超过 50 万。再加上“软件工程”、“计算机网络”、“信息技术”等计算机类的专业,全国的计算机专业数和在校生人数将比 800 所和 50 万多得多。面对如此庞大的队伍,我们的毕业生不可能都去研究计算机本身,社会也不需要这么多人从事计算机研究。所以现在计算机专业的含义已经完全与 30 年前的计算机专业不同了,现在绝大部分计算机专业人员主要是做“计算机应用”,甚至是在“应用计算机”,而不是在研究计算机本身了。

在这种形式下,计算机专业怎么办?作为计算机专业的第一门程序设计课程怎么讲?摆在每个组织计算机专业教学的人的面前,更摆在教授第一门程序设计课程的教师面前。显然,目前计算机教育应该面向应用,第一门程序设计课程也不应该再去讲程序设计语言本身了,而应该讲述程序设计语言的应用,即应该“真正地讲程序设计”,把授课重点从讲授语言的符号、语句等语言成分上转移到讲授程序设计上,教学的目的是教会学生“怎么编程序”,而不是背几个语言符号、说明、语句等,应该把这门课程的名字从“×××语言”改为“高级语言程序设计”或“程序设计基础”,把授课重点真正放在程序设计上。尤其是一些非计算机专业,更应该跳出讲授“语言”的误区,因为他们更是在“应用计算机”,是用计算机解决本专业领域的问题,更没有必要去背那些死的语言符号、语法概念,更应该学会怎么编程序。

本书秉承上述面向应用的主导思想,依托 C 语言讲述“程序设计”,重点在于程序设计。在

这种思想的指导下,本书对C语言本身采取“有所取、有所不取”的策略。对于那些常用的语言成分、直接与讲述程序设计方法有关的语言成分,穿插在程序设计过程中详细、准确地介绍;对于那些与程序设计方法联系不太紧密,但是还常用的部分,放在最后简单介绍;而对于那些与讲述程序设计方法关系不太大,也不常用的部分则根本不涉及。

本书主要作者从事计算机教学30余年,讲授过10多门计算机方面的课程,是吉林大学计算机科学与技术学院“高级语言程序设计”课程梯队负责人;主持和参加过许多大型项目的研发,并亲自参加这些项目的程序编码工作;曾20余次为吉林大学计算机科学系、计算机科学与技术学院、软件学院本科生主讲“程序设计”课程;对程序设计课程的教学内容和讲授方法有独到见解。本书是作者30余年教学实践的总结。

作为大学本科计算机专业基础课教材,本书具有以下特点;

(1) 最大特点是“以程序设计为主线”。按照程序设计的思路组织全书内容,真正在讲授程序设计,而不是讲语言,摒弃了目前各种程序设计书中流行的以“解释程序设计语言”为主的做法。

(2) 全书整体结构好,章节安排合理,注意由浅入深地介绍程序设计知识。比如有关函数的知识,由浅入深地分4章逐步介绍;有关指针的知识也分散到各个章节中介绍。免得集中于一章,使读者学习起来枯燥乏味,难以接受。

(3) 全书自始至终贯穿结构化程序设计思想,所有例题都具有良好的结构和程序设计风格。其目的是给读者一个示范,使读者从开始学习程序设计起就养成一个良好的程序设计习惯和风格。

(4) 本书图文并茂,引进PAD表示程序逻辑。PAD的结构比传统的流程图、NS图等都好,同时也比直接用程序表示算法更为直观,易于理解。

(5) 全书配备大量例题和习题,并且全部为程序设计题目。例题的讲解从构造算法出发,以训练读者的编程能力为目标;概念、语言成分的介绍穿插在程序设计之中。本书中大量的习题可供读者做练习和进一步提高使用。

全书共14章,大致分为4部分。

第一部分基本知识,包括第1章,介绍计算、算法和程序设计的基本概念。

第二部分程序设计,包括第2、3、4、5、9、10、12章。第2章简单程序设计,介绍顺序程序设计、数据及其类型、表达式、赋值、输入输出;第3章介绍分支程序设计;第4章介绍循环程序设计;第5章简单介绍模块化程序设计思想,引进子程序和函数的概念;第9章进一步介绍函数,讲述参数、作用域;第10章介绍递归程序设计;第12章介绍程序开发和结构化程序设计,包括结构化程序设计原则、程序风格、“自顶向下、逐步求精”的程序设计技术。

第三部分数据组织,包括第6、7、8、11、13章。第6章讲述批量数据组织——数组;第7章介绍指针;第8章讲述复杂的表单数据,引进结构体和共用体;第11章讲述外部数据组织——文件及其操作;第13章讲述动态数据组织及其在程序设计中的应用。

第四部分为第14章,介绍若干较深入的问题及C语言独有的特性。进一步介绍函数,讲述

函数作参数、函数副作用等；以及存储类别、位操作、break 语句、编译预处理等。

本书第 1、2、3、4、5、6、8、10 章由张长海执笔，第 7、12、13、14 章由陈娟执笔，第 9、11 章由刘磊执笔。最后全书由张长海统稿。

在本书的编写过程中，作者参阅并借鉴了国内外诸多同行的文章和著作，在此向他们致以谢意，并恕不一一列举、标明。

由于作者学术水平有限及出版时间仓促，错误和不足在所难免，敬请各位读者批评指正，作者十分感谢。

作者

2007 年 12 月于长春

目 录

第 1 章 绪论	1	2.5.2 常量标识符	27
1.1 求玉米产量——计算	1	2.5.3 变量	27
1.2 算法	3	2.6 数据类型	29
1.3 PAD	4	2.6.1 浮点类型	29
1.4 程序	6	2.6.2 整数类型	30
1.5 运行	8	2.6.3 字符类型	31
1.6 C 语言	10	2.6.4 混合运算	31
1.7 Visual C++ 集成开发环境	11	2.6.5 类型转换	32
1.7.1 启动 Visual C++	11	2.6.6 左值	34
1.7.2 建立环境	12	2.7 输入输出	35
1.7.3 输入、编辑源程序	13	2.7.1 字符输入	36
1.7.4 编译	13	2.7.2 字符输出	36
1.7.5 连接	14	2.7.3 格式输入	36
1.7.6 运行	14	2.7.4 格式输出	37
本章小结	15	2.8 顺序控制结构	41
习题 1	15	本章小结	41
第 2 章 简单程序设计	16	习题 2	42
2.1 求绿化带宽度——简单程序 ..	16	第 3 章 分支程序设计	44
2.2 语句	17	3.1 判断成绩是否及格——双分支	
2.3 表达式	18	程序设计	44
2.3.1 表达式语句	20	3.2 成绩加上获奖信息——单分支	
2.3.2 赋值	20	程序设计	47
2.4 基本符号	21	3.3 关于 if	52
2.4.1 标识符	21	3.4 布尔类型	53
2.4.2 关键字	22	3.4.1 关系运算	53
2.4.3 间隔符	22	3.4.2 布尔运算	54
2.4.4 注释	23	3.5 获奖分等级——多分支	
2.5 数据	24	程序设计	54
2.5.1 字面常量	24	本章小结	60

习题 3	60	7.1.1 指针类型和指针变量	150
第 4 章 循环程序设计	62	7.1.2 指针所指变量	152
4.1 计算平均成绩——循环程序	62	7.1.3 空指针和无效指针	154
4.1.1 后判断条件的循环	63	7.2 指针运算	155
4.1.2 先判断条件的循环	67	7.3 指针与数组	158
4.1.3 for 语句	71	7.3.1 用指针标识数组	158
4.2 打印 99 表——多重循环	75	7.3.2 多维数组与指针	162
4.3 程序设计实例	78	7.3.3 指针数组	165
本章小结	87	7.3.4 指针与数组总结	169
习题 4	87	7.4 指针与字符串	170
第 5 章 模块化程序设计——函数	92	7.5 指向指针的指针	174
5.1 重写例 1.1——模块化程序 设计	92	7.6 命令行参数	176
5.2 函数	94	本章小结	178
5.2.1 函数定义	96	习题 7	178
5.2.2 函数调用	97	第 8 章 表单数据组织——结构体	181
5.2.3 先调用后定义	100	8.1 图书卡管理	181
5.3 程序设计实例	100	8.2 保存图书卡——结构体	184
本章小结	116	8.2.1 定义结构体类型	185
习题 5	116	8.2.2 结构体类型名	187
第 6 章 批量数据组织——数组	119	8.2.3 结构体变量	187
6.1 整数排序——数组类型	119	8.2.4 指向结构体变量的指针	188
6.1.1 数组声明	121	8.2.5 访问结构体变量的成分	189
6.1.2 下标表达式	122	8.3 表示语言种类——枚举 类型	190
6.1.3 应注意的问题	122	8.4 职工登记卡——共用体	192
6.2 矩阵乘积——多维数组	123	8.4.1 共用体类型	193
6.3 程序设计实例	125	8.4.2 限制	194
6.4 带序号的排序——数组 初值	137	8.4.3 switch 语句与共用体	195
6.5 输出回文字——字符串	140	8.5 程序设计实例	195
6.6 类型定义	144	本章小结	202
本章小结	145	习题 8	202
习题 6	145	第 9 章 再论函数	206
第 7 章 指针	149	9.1 参数	206
7.1 指针与变量	149	9.1.1 C 参数传递规则	206
		9.1.2 指针作参数	208

9.1.3 数组作参数	215	12.3 程序风格	279
9.1.4 结构体作参数	219	12.3.1 行文格式	280
9.2 返回指针的函数	220	12.3.2 标识符	281
9.3 返回结构体值的函数	223	12.3.3 注释	281
9.4 作用域	224	12.3.4 程序说明	281
9.4.1 作用域	224	12.4 八皇后——穷举法与试	
9.4.2 生存期	225	探法	282
9.4.3 局部量和全局量	226	本章小结	289
本章小结	229	习题 12	289
习题 9	229	第 13 章 动态数据组织	292
第 10 章 递归程序设计	232	13.1 打印法雷序列——动态	
10.1 计算 $n!$ ——递归程序		数据结构	292
设计	232	13.2 动态变量	295
10.2 计算算术表达式的		13.3 链表	297
值——间接递归	237	13.3.1 单向链表	298
10.3 递归程序执行过程	239	13.3.2 栈	301
本章小结	250	13.3.3 队列	303
习题 10	251	13.4 树	305
第 11 章 外部数据组织——文件	253	13.5 图	311
11.1 重新考虑图书卡管理		13.6 程序设计实例	312
问题	253	本章小结	318
11.2 文件概述	254	习题 13	318
11.3 文件操作	257	第 14 章 若干较深入的问题和	
11.3.1 打开和关闭文件	257	C 语言独有的特性	322
11.3.2 字符读/写	259	14.1 不定方向的数组排序——函数	
11.3.3 字符串读/写	259	指针	322
11.3.4 数据块读/写	259	14.2 计算定积分——函数作	
11.3.5 格式化读/写	260	参数	324
11.3.6 文件定位	261	14.3 函数副作用	327
11.4 程序设计实例	262	14.4 运算	329
本章小结	269	14.4.1 赋值运算	329
习题 11	269	14.4.2 顺序表达式	329
第 12 章 程序开发	271	14.4.3 条件表达式	330
12.1 验证三角形外心定		14.4.4 位运算	330
理——自顶向下、逐步求精 ..	271	14.5 语句	333
12.2 结构化程序设计原则	278	14.5.1 break	333

14.5.2	continue	334	14.7.7	类型定义符	343
14.5.3	for 的延伸	334	14.8	编译预处理	344
14.5.4	goto 和标号	335	14.8.1	宏定义	344
14.6	位段	336	14.8.2	文件包含	347
14.7	存储类别	337	14.8.3	条件编译	348
14.7.1	数据在内存中的存储	337	本章小结	350	
14.7.2	自动存储类别	338	习题 14	351	
14.7.3	寄存器存储类别	339	附录一 ASCII 字符集	352	
14.7.4	变量的静态存储类别	340	附录二 标准库头文件表	355	
14.7.5	变量的外部存储类别	341	参考文献	356	
14.7.6	函数的存储类别	342			



绪 论

现代计算机从出现至今不过 60 多年的时间,但其发展速度却是任何一种新技术都不可比拟的,目前计算机已经渗透到社会的各个领域。本书以 C 语言为载体向读者揭示如何编制计算机程序,即如何使用计算机解决科技、生产、事务处理等方面的问题,介绍程序设计的基本方法、技术和技巧。

1.1 求玉米产量——计算

【例 1.1】 已知玉米每亩(本书采用实际耕作者所用的 1 亩 = 1 000 m²;作为练习,读者可以以 1 市亩 = 666.67 m² 自行编程练习)产量为 950 kg。如图 1.1 所示,现有一块近似四边形的地块,位于南北方向路东侧,东西方向路北侧。其中一个顶点距离南北方向路 547 m,距离东西方向路 411 m;另一个顶点距离南北方向路 804 m,距离东西方向路 77 m;第三个顶点距离南北方向路 39 m,距离东西方向路 208 m;第四个顶点距离南北方向路 116 m,距离东西方向路 332 m。若该地块种植玉米,求这块地玉米产量。

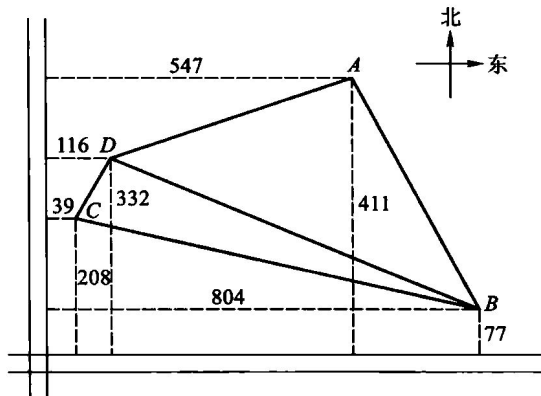


图 1.1 四边形地块

解:

第一步分析,解决该问题应该首先把问题数学化。求总产量应该用总面积乘以单位面积产

量,关键问题是怎样求总面积。把南北方向路定义为 y 方向坐标轴,北为正;把东西方向路定义为 x 方向坐标轴,东为正;把4个顶点分别记为 A, B, C, D ;把4个顶点到道路的距离分别定义为相应点的坐标值。经过分析,最后此问题大致可以经过以下步骤实现。

- (1) 计算该地块的面积。
- (2) 折合成亩。
- (3) 求总产量。

其中,“折合成亩”和“求总产量”两步的计算很简单。

第二步分析,计算地块面积。根据数学知识,可以把四边形的 B, D 两个顶点相连,构成两个三角形 ABD 和 BCD ,然后:

- (1) 计算三角形 ABD 的面积,设为 S_1 ;
- (2) 计算三角形 BCD 的面积,设为 S_2 ;
- (3) 把 S_1 和 S_2 相加,得到四边形面积(也就是地块面积)。

第三步分析,计算三角形 ABD 的面积(计算三角形 BCD 的面积算法与计算三角形 ABD 的面积是一样的,可以参照进行)。根据数学知识,计算三角形面积有很多方法,选择一种合适的即可。可以选择公式

$$S = \sqrt{s(s-a)(s-b)(s-c)}$$

其中,

S 为三角形面积;

a, b, c 分别为3条边的边长;

$$s = \frac{a+b+c}{2}。$$

第四步分析,按照第三步所选定的方法计算三角形 ABD 的面积。应该按如下步骤计算:

- (1) 求边 AB 的长度,记为 ab ;
- (2) 求边 BD 的长度,记为 bd ;
- (3) 求边 AD 的长度,记为 ad ;
- (4) 计算 s ;
- (5) 最后计算 S_1 。

第五步分析,求边 AB 长度(求边 BD, AD 长度的算法与求边 AB 长度的算法是一样的,可以参照进行)。选择如下两点间距离公式,并按公式计算。

$$L = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

如上已经把解决该问题的各个步骤分析清楚。实际计算过程应该是:

- (1) 按公式

$$L = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

计算边 AB 长度 ab 。

- (2) 用同样的公式计算边 BD 长度 bd 。
- (3) 用同样的公式计算边 AD 长度 ad 。
- (4) 按公式

$$s = \frac{a + b + c}{2}$$

计算 s 。

- (5) 按公式

$$S = \sqrt{s(s-a)(s-b)(s-c)}$$

计算三角形 ABD 的面积 S_1 。

- (6) 重复上述(1)~(5)步计算三角形 BCD 的面积 S_2 。
- (7) 计算总面积 $S = S_1 + S_2$ 。
- (8) 把面积 S 折合成亩 $SS = S/1\ 000$ 。
- (9) 计算总产量 $M = SS \times 950$ 。

选择一种程序设计语言,把上述计算过程用该程序设计语言表示出来,就是计算机程序。这个过程就是“程序设计”。

这就是一个简单的实际计算问题,也是一个简单的程序设计过程。一般的,一个现实问题要用计算机来解决,大致经过如下步骤。

- (1) 建立数学模型——把实际问题转化为数学问题。
- (2) 找出计算方法——为数学问题的求解找出合适的方法。
- (3) 进行算法分析——为实现计算方法给出具体的算法。
- (4) 选择一种程序设计语言,编写计算机程序——写程序。
- (5) 调试程序——保证程序运行的正确性。
- (6) 测试程序的正确性——组装测试、确认测试。
- (7) 最后,程序交付使用并维护。

在上述例 1.1 简单问题的解决过程中,

第一步分析是建立数学模型,建立了坐标系,把实际求玉米产量的问题转化为数学问题;

第二步、第三步分析是找出计算方法,为计算四边形面积给出方法;

第四步、第五步分析是算法分析,得到计算四边形面积的具体计算步骤。

以下写程序、调试程序、测试程序本例中没有给出过程。

最后是交付使用并维护。

1.2 算 法

在 1.1 节中最后给出的计算步骤(1)~(9)是解决例 1.1 问题的算法。“程序设计”的任务

就是找出算法(算法分析)、编出计算机程序、调试测试程序和运行程序。找出算法,本课程要解决一部分,将来还要学习“算法分析”课。但是,由于现实世界是缤纷复杂的,真正解决算法问题还需要靠长期的学习、积累和悟性。选择一种程序设计语言,要根据具体问题来决定,本书以C语言为背景讲授“程序设计”,也就是说选择了C语言。千万不可忽略上机调试阶段,因为一方面任何程序不经过调试是不能保证其正确性的,另一方面程序设计是实践性很强的课程,计算机专业更是实践性极强的学科。

算法是一个计算过程,具体指明所应进行的操作,描述解决问题的方法和途径,它是程序设计的基础和精髓。一个有效的算法具有如下特点。

1. 有穷性

一般情况下,一个算法应该在有限的时间内终止,而不应该是无限的。进一步,有穷性往往是指在合理的时间范围内终止。比如后文讲述的“Hanoi塔”问题的循环迭代算法,若1s计算一次,大约需要耗费5849亿年,虽然是有穷的,但是显然是毫无意义的。

2. 确定性

算法中的每一个步骤都应该是确定的,含义是唯一的,而不应该是模糊的、模棱两可的。例如,求解一元二次方程,如果算法直接写成

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

显然是不确定的,不能构成确定的算法步骤。比如当 a 等于0时怎么办,当 $b^2 - 4ac$ 小于0时如何操作? 这些都没有确定的描述。

3. 有效性

算法中的每一个步骤都应该是有效的,能够被有效地执行并得到确定的结果,而不应该存在无效的操作。比如在一个算法中存在操作“ $x \rightarrow x$ ”,显然无意义,也是无效的;又比如操作“ $a \div 0$ ”也是不能有效执行的。

算法由某些基本成分所组成,这些基本成分是一些基本的操作及控制结构。构成算法的基本操作包括:表达式及为变量赋值;

读(输入);

写(输出)。

基本的控制结构包括:顺序控制结构;

分支控制结构;

循环控制结构;

函数调用;

函数返回。

1.3 PAD

描述算法有多种多样的方法。例如,流程图、NS图、程序等。本书采用所谓的问题分析图

(Problem Analysis Diagram)来描述算法。问题分析图简称 PAD, PAD 有许多优点:

PAD 使用二维的树形结构描述程序的逻辑,比直接用程序(可以说程序的表现形式是一维的)表示算法更加清晰、直观;PAD 使用结构化的、概括的抽象记号系统,比用流程图表示算法更简练、紧凑、层次分明;PAD 是开放式的,比封闭式的 NS 图更清晰、分明,也更便于修改。

为简单明了起见及出于印刷上的原因,也为了适应 C 语言的特点,本书将标准 PAD 的记号系统做了一定的修正。这里先给出 PAD 记号系统的基本格式,再给出例 1.1 算法的 PAD 描述。更复杂控制结构的 PAD 表示将在下文涉及的地方逐步给出。

PAD 把基本操作序列用方框括起来,表示成图 1.2 的形式。

PAD 把顺序执行的操作用一条竖线连接起来,表示成图 1.3 的形式。竖线从上向下表示程序的执行顺序,竖线所连接起来的是一个操作成分。

【例 1.2】 例 1.1 问题算法(计算步骤(1)~(9))的 PAD 描述如图 1.4 所示。

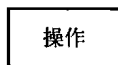


图 1.2 基本操作序列

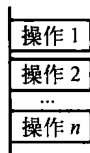


图 1.3 顺序控制结构

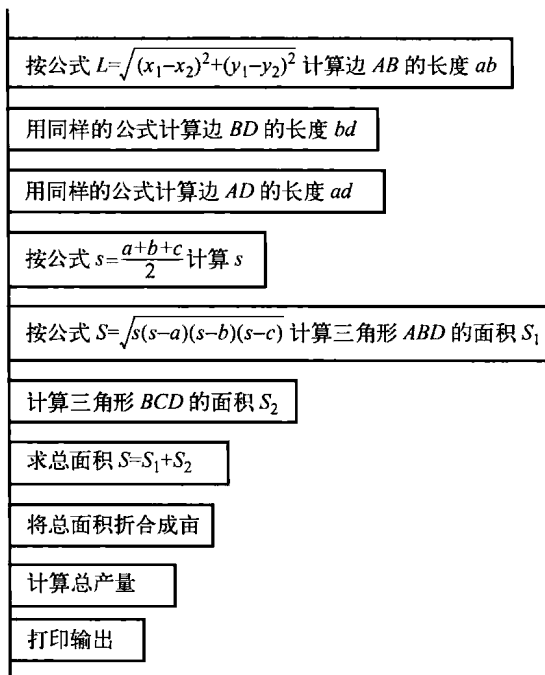


图 1.4 例 1.1 算法的 PAD 描述

1.4 程 序

一个庞大的计算机系统是怎样有条不紊地工作的呢?答案是:计算机系统的工作是由事先设计好的程序来控制的。人们首先按照需要把让计算机做的工作编排成计算机程序,并把程序送入计算机,然后启动计算机执行程序。计算机的控制器从程序地第一条指令开始,顺序地逐条取出指令进行解释,然后按照指令的规定和要求指挥整个计算机系统工作,从而完成人们设想的要求计算机完成的工作。

程序是一个指令序列,也就是用指令排成的一个工作顺序、工作步骤。人们平常也使用程序这个名词,如运动会程序。计算机程序是用计算机指令为计算机排定的工作顺序、工作步骤。为计算机编排程序的过程称为程序设计。

描述程序必须使用一种语言。程序设计语言是指用于编写、描述计算机程序的语言。C 语言是使用最广泛的程序设计语言,本书以 C 语言为载体讲述程序设计。

本章的例 1.1 计算过程,按图 1.4 的 PAD,写出 C 程序,如例 1.3 所示。

【例 1.3】 例 1.1 的 C 程序。

```
#include "stdio. h"      //括入标准 I/O 函数库头文件
#include "math. h"      //括入标准数学函数库头文件
#define xa 547          //以下分别表示 4 个顶点的 x、y 坐标
#define ya 411
#define xb 804
#define yb 77
#define xc 39
#define yc 208
#define xd 116
#define yd 332
float ab, bd, ad, bc, cd;    //分别表示两个三角形的 3 条边,以对应的顶点命名
float s, s1, s2, ss, m;     //计算所用变量:ss 表示总面积;m 表示玉米总产量
void main( void ) {        //主函数
    //计算三角形 ABD 的面积
    ab = sqrt( ( xa - xb ) * ( xa - xb ) + ( ya - yb ) * ( ya - yb ) ); //边 AB 长度
    bd = sqrt( ( xb - xd ) * ( xb - xd ) + ( yb - yd ) * ( yb - yd ) ); //边 BD 长度
    ad = sqrt( ( xa - xd ) * ( xa - xd ) + ( ya - yd ) * ( ya - yd ) ); //边 AD 长度
    s = ( ab + bd + ad ) / 2; //求 s 值
    s1 = sqrt( s * ( s - ab ) * ( s - bd ) * ( s - ad ) ); //面积 s1
    //计算三角形 BCD 的面积
    bc = sqrt( ( xb - xc ) * ( xb - xc ) + ( yb - yc ) * ( yb - yc ) ); //边 BC 长度
```



```

cd = sqrt((xc - xd) * (xc - xd) + (yc - yd) * (yc - yd)); //边 CD 长度
s = (bc + bd + cd)/2; //边 BD 长度前面已经计算
s2 = sqrt(s * (s - bc) * (s - bd) * (s - cd));
ss = s1 + s2; //总面积 s = s1 + s2
ss = ss/1000; //折合成亩
m = ss * 950; //计算玉米总产量
printf("m = %10.3f\n", m); //打印输出
}

```

这是一个完整的 C 程序,它全部存放在一个文件中。在这个程序中,从“void main(void) {”一行到最后是主函数,构成一个顶层声明,该行之前的每一行均为一个顶层声明,包括括入文件的、常量定义的、变量声明的等。

在实际应用中,若程序规模较大,往往把一个程序分别存放在多个不同的文件中,每个程序文件称为一个“编译单元”。如图 1.5 所示是由两个编译单元组成的一个完整的 C 程序,每个编译单元保存在一个源程序文件中。执行此程序将打印字符串

Hello!

文件 hello.c 的内容:

```

#include "stdio.h"
int m;
void hello(void){
    printf("Hello!\n");
}

```

文件 startup.c 的内容:

```

extern void hello(void);
int main(void) {
    hello();
    return 0;
}

```

图 1.5 一个完整的 C 程序

图 1.5 程序的执行过程是:函数 main 调用函数 hello;函数 hello 打印字符串“Hello!”。

图 1.5 程序由两个源程序文件 hello.c、startup.c 组成,每个源程序文件称为一个编译单元。

文件 hello.c 的第一行、第二行、第三行~第五行以及文件 startup.c 的第一行、第二行~第五行分别是顶层声明。

文件 hello.c 的第二行声明 int 类型变量 m;第三行~第五行定义函数 hello,打印字符串“Hello!”;第一行括入一个头文件“stdio.h”,此头文件中包含所有 I/O 函数的定义。

文件 startup.c 的第二行~第五行定义主函数 main,这是必需的,任何 C 程序必须包含且仅包含一个以 main 命名的函数,称为 C 程序的主函数,C 程序从这个函数开始执行。main 调用函数 hello,hello 的声明不在本文件中,而在另一个文件 hello.c 中,因此使用一个外部声明来指明。

概括起来

- 一个 C 程序由一个或若干编译单元组成,每个编译单元是一个源程序文件;
- 一个编译单元由若干顶层声明组成,每个顶层声明是一个声明或函数定义,其中主要为函数定义;