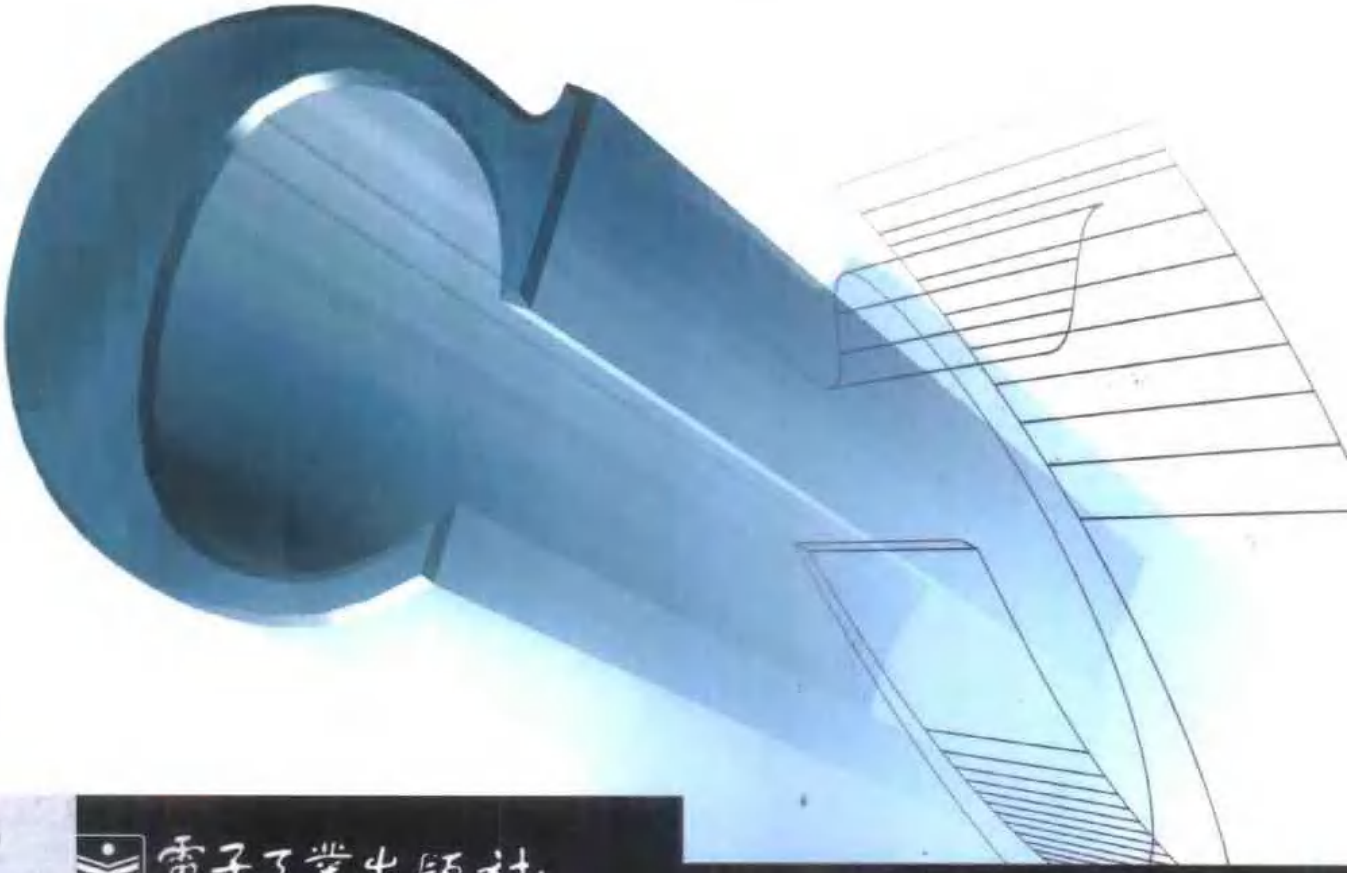
普通高等教育“十五”国家级规划教材

高职高专计算机系列教材

C语言程序设计

徐建民 张 萍
王 苗 杨学全 编著



2C-43

電子工業出版社

普通高等教育"十五"国家级规划教材

高职高专计算机系列教材

C语言程序设计

徐建民　张　萍　王　苗　杨学全　杨著

電子工業出版社
Publishing House of Electronics Industry
北京·BEIJING

内 容 简 介

程序设计是计算机及相关专业学生必备的基本技能之一。因此,“程序设计”课程是计算机及相关专业的一门重要的必修课。根据中国计算机学会高职高专教育学组审定的教学大纲和技能培养的基本方法,本书在介绍C语言基本概念和知识的基础上,重点强调了基本技能的训练。

全书共分13章,分别讲述了程序设计的基础知识、C语言的基本概念、顺序结构程序设计、分支结构程序设计、循环结构程序设计、函数、指针、数组、结构体和共用体、编译预处理、位运算、文件和图形处理等内容。根据技能培养的基本要求,本书给出了比较多的例题、习题,以供学习者模仿和练习。

本书讲述力求准确、简练,强调知识的层次性,例题和习题选用讲究、丰富,强调编程技能的培养。在内容安排上,本书遵循了“难点分解”的原则,即将复杂的内容分解到相关的、不同的章节中讲述,从而减少学生学习的难度。本书既可作为计算机及相关专业本、专科学生的教材,也可以作为编程工作者,尤其是C语言初学者的参考书。

本书被教育部列入普通高等教育“十五”国家级规划教材。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在题编目(CIP)数据

C语言程序设计/徐建民等编著 .—北京:电子工业出版社,2002.8
高职高专计算机系列教材
ISBN 7-5053-7913-5

Ⅰ.C… Ⅱ.徐… Ⅲ.C语言—程序设计—高等学校:技术学校—教材 Ⅳ.TP312

中据版本图书馆 CIP 数据教字(2002)第 059884 号

责任编辑:张孟玮 特约编辑:韩玉彬
印 刷:北京李史山胶印厂
出版发行:电子工业出版社 http://www.phei.com.cn
北京市海淀区万寿路 173 信编 邮编 100036
级 销:各地新华书店
开 本:787×1 092 1/16 印张:17.25 字数:442 千字
题 次:2002 年 8 月第 1 版 2002 年 8 月第 1 次印刷
印 数:10 100 册 定价:20.00 元

凡购买电子工业出版社的图书,如有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系。
联系电话:(010)68279077

前　言

程序设计是计算机及其相关专业学生必备的技能之一，是一门重要的必修课。“程序设计”课程不仅仅应当注重知识的讲授，还应该强调基本技能的训练。为了实现技能训练的基本目标，本书的编写主要遵循了以下原则。

1.作者认为技能训练的过程是一个循序渐进的过程，学生程序设计技能的培养应该分三步走：掌握基本知识，然后从读实例开始，逐步达到能够编写程序的目的。相应地，本教材内容的安排与讲述就遵循这一原则。

2.人们认识复杂事物的一个基本方法是分解，软件开发的过程就遵循了这一原则。对学生来说，一门新课相当于是一个需要接受的新事物，采用“分解与综合”的方法可以使之感觉到学习更容易。因此本教材内容的安排基本遵循“循块大小适中”的原则，即每一章、每一节乃至一个知识点都尽可能保持内容难易程度适中，将难点适当分解，循于学生掌循。

3.软件工程的基本思想之一是规范化文档。一开始就注意培养良好的程序设计风格，对一个学生养成好的习惯十分重要，所以本书遵循的第三个原则是每一道例题程序都尽可能有良好的程序风块。

4.学生是学习的主体，学生主动参与循学过程对学生的学习十分重要。本教材内容的安排强调学生参与循学过程，希望使用该循材的循师能注意这一点，适当安排学生一些自己思考和分析的时间。

编写一本优秀的教材是一件非常不容易的工作，涉及到诸多因素，比如内容安排的科学性、编写者的文字表达能力、作者之间的协作精神等。尽管本书三易其稿，每写一稿编者都考虑本书和其他循材相比有什么特色、有什么优点；尽管本书的定稿经过了不止一人的审阅，但编者仍感觉不尽人意，惟恐对不起孜孜求学的学子们，对不起那些关心这本教材编写工作的朋友们。由于作者水平、循力所采，再加上时间紧迫，也只好暂且这样了。

衷心感谢南京师范大学的俞光昀老师，块阳大学的朱乃立老师，电子工业出版社的张孟玮先生。他们的循奋与敬业使我们受益匪浅。

衷心感谢参加这本书审阅的王长仑、采晓丽等老师；他们的许多中肯意见对本书的编写质量起到了很大作用。

衷心感谢所有关心本书的师长和朋友。

编著者

2002 年 4 月

目　录

此为试读，需要完整PDF请访问：www.ertongbook.com

第 1 章　程序设计基础

1.1　程序设计的基本概念

1.1.1　程序

我们做任何事情都遵循着一定的程序，如开会的议程、做饭的食谱、演奏的乐谱等都是程序，在程序的指导下，人们可以有秩序地、有效地完成一项工作。随着计算机的问世和普及，“程序”逐渐被专业化，它通常特指：为让计算机完成特定任务（如解决某一算题或控制某一过程）而设计的指令序列。

程序是让计算机运行的。程序就是对解决问题所需要的数据的描述和对数据处理的描述。对数据的描述称之为“数据结构”，对数据处理的描述称之为“算法”，而编写程序所用的计算机语言，称之为“程序设计语言”。

1.1.2　算法与数据结构

著名计算机科学家沃斯（N.Wirth）提出过一个经典公式：

算法 + 数据结构 = 程序

算法反映了计算机的执行过程，是对解决特定问题的操作步骤的一种描述。数据结构是对参与运算的数据及它们之间的关系进行的描述，算法和数据结构是程序的两个重要方面。

1. 算法

【例 1.1】　输入 3 个数，求其最大值。

问题分析：设 num1, num2, num3 存放 3 个数，max 存放其最大值。

为求其最大值，就必须对 3 个数进行比较，可按如下步骤去做：

① 输入 3 个数 num1, num2, num3；

② 先把第 1 个数 num1 的值赋给 max；

③ 将第 2 个数 num2 与 max 比较，如果 num2＞max，则把第 2 个数 num2 的值赋给 max（否则不做任何工作）；

④ 将第 3 个数 num3 与 max 比较，如果 num3＞max，则把第 3 个数 num3 的值赋给 max（否则不做任何工作）；

⑤ 输出 max 的值，即最大值。

从例 1.1 可以看出，首先分析题目，然后寻找一种实现这个问题所要完成功能的方法，这种方法的具体化就称做为算法。因此可以说，算法是由一套明确的规则组成的一些步骤，它指定了操作顺序并通过有限个步骤将问题解决、得出结果。

一个算法应具有以下 5 个特性：

（1）有穷性

一个算法必须总是在执行有限个操作步骤和可以接受的时间内完成其执行过程。也即

此为试读，需要完整PDF请访问：www.ertongbook.com

是说，对于一个算法，要求其在时间和空间上均是有穷的。例如：一个采集气象数据并加以计算进行天气预报的应用程序，如果不能及时得到结果，超出了可以接受的时间，就起不到天气预报的作用。

（2）确定性

算法中的每一步都必须有明确的含义，不允许存在二义性。例如："将成绩优秀的同学名单打印输出"，在这一描述中"成绩优秀"就很不明确，是每门功课均为 95 分以上？还是指总成绩在多少分以上？

（3）有效性

算法中描述的每一步操作都应该能有效地执行，并最终得到确定的结果。例如：当 Y=0 时，X/Y 是不能有效执行的。

（4）输入

一个算法有零个或多个输入数据。例如：计算 1～10 的累计和的算法，是无须输入数据，而对 10 个数据进行排序的算法，却需要从键盘上输入这 10 个数据。

（5）输出

一个算法应该有 1 个或多个输出数据。执行算法的目的是为了求解，而"解"就是输出，因此没有输出的算法是毫无意义的。

2. 算法的表示方法

算法的表示方法很多，常见的有：自然语言、传统流程图、N-S 结构图、伪代码、PAD 图等。

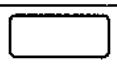
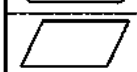
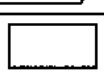
（1）用自然语言表示

自然语言就是人们日常使用的语言，可以是中文、英文等。用自然语言表示的算法通俗易懂，但一般篇幅冗长，表达上往往不易准确，容易引起理解上的"歧义性"。所以，一般用于算法较简单的情况，如例 1.1 所示。

（2）用传统流程图表示

传统流程图是用规定的一组图形符号、流程线和文字说明来表示各种操作算法的表示方法。传统流程图常用的符号如表 1.1 所示。

表 1.1　传统流程图常用符号

符　号	符号名称	含　义
	起止框	表示算法的开始和结束
	输入/输出框	表示输入/输出操作
	处理框	表示对框内的内容进行处理
	判断框	表示对框内的条件进行判断
	流程线	表示流程的方向
	连接点	表示两个具有同一标记的"连接点"应连接成一个点

【例 1.2】　用传统流程图描述算法如图 1.1 所示。

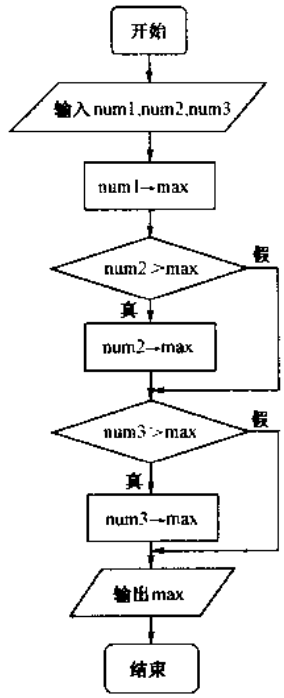


图 1.1

用传统流程图表示算法直观形象，算法的逻辑流程一目了然，便于理解。但占用篇幅较大，画起来比较麻烦，而且又由于允许使用流程线，使用者可以随心所欲，使流程可以任意转移，从而造成阅读和修改上的困难。

（3）用 N-S 结构图表示

针对传统流程图存在的问题，美国学者 I.Nassi 和 B.Shneiderman 于 1973 年提出一种新的结构化流程图形式，即简称为 N-S 结构图。Chapin 在 1974 年对其进行了进一步的扩展，因此，N-S 结构图又称为 Chapin 图或盒状图。

N-S 结构图的目标是开发一种不破坏结构化基本构成元素的过程设计表示。其主要特点是完全取消了流程线，不允许有随意的控制流，全部算法写在一个矩形框内，该矩形框以 3 种基本结构（顺序、分支、循环）描述符号为基础复合而成。

N-S 结构图表示的 3 种基本结构如下：

① 顺序结构。顺序结构是最简单的基本结构。在顺序结构中，要求顺序地执行且必须执行由先后顺序排列的每一个最基本的处理单位。图 1.2（a）所示是用传统流程图表示的顺序结构，图 1.2（b）所示是用 N-S 结构图表示的顺序结构，先执行处理 A，然后再顺序执行处理 B。

② 分支结构。分支结构又称做选择结构。在分支结构中，要根据逻辑条件的成立与否，分别选择执行不同的处理。如图 1.3 所示，当逻辑条件成立时，执行处理 A，否则执行处理 B。

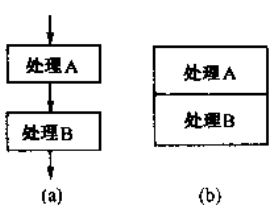


图 1.2　顺序结构

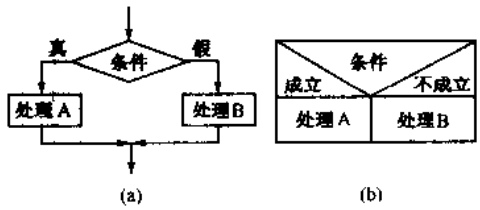


图 1.3　分支结构

③ 循环结构。循环结构一般分为当型循环和直到型循环。

（a）当型循环。在当型循环结构中，当逻辑条件成立时，就反复执行处理 A（称为循环体），直到逻辑条件不成立时结束，如图 1.4 所示。

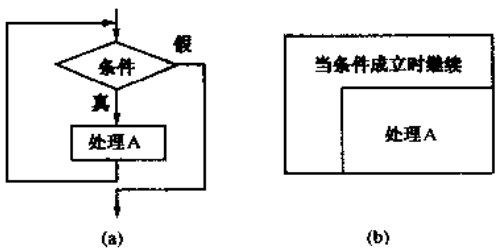


图 1.4　当型循环结构

（b）直到型循环。在直到型循环结构中，反复执行处理 A，直到逻辑条件成立时结束（即逻辑条件不成立时继续执行），如图 1.5 所示。

【例 1.3】 用 N-S 结构图描述的例 1.1，求 3 个数中的最大值的算法，如图 1.6 所示。

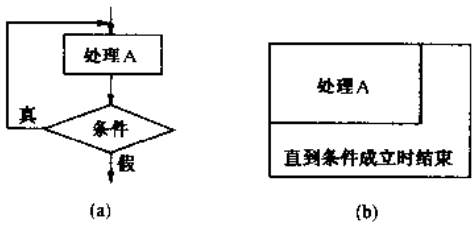


图 1.5 直到型循环结构

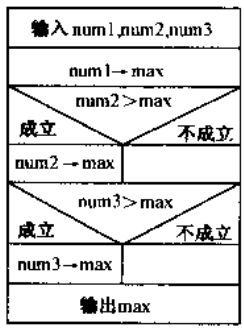


图 1.6

（4）用伪代码表示

伪代码是用一种介于自然语言和计算机语言之间的文字和符号来描述算法。伪代码的表现形式比较灵活自由，没有严谨的语法格式。

【例 1.4】 用伪代码描述例 1.1，求 3 个数中的最大值的算法。

```
input num1,num2,num3
num1→max
if num2 > max then num2→max
if num3 > max then num3→max
print max
```

理论已经证明，任何复杂的算法均可以用顺序、分支、循环这 3 种基本结构组合、嵌套进行描述。由于 N-S 结构图无箭头指向，而局限在一个个嵌套的框中，最后描述的结果必然是结构化的，因此 N-S 结构图描述表示的算法，适用于结构化程序设计（有关结构化程序设计的概念将在后面介绍），本书将主要采用 N-S 结构图表示算法。

3. 数据结构

计算机处理的对象是数据，数据是描述客观事物的数、字符以及所有计算机能够接受和处理的信息符号的总称。数据结构是指数据的类型和数据的组织形式。数据类型体现了数据的取值范围和合法的运算，数据的组织形式体现了相关数据之间的关系。

数据结构与算法有着密切的关系，只有明确了问题的算法，才能更好地构造数据结构；但要选择好的算法，又常常依赖于好的数据结构。事实上，程序就是在数据的某些特定的表示方式和结构的基础上对抽象算法的具体表述。因此，编写一个程序的关键就是合理地组织数据和设计好的算法。

1.1.3 程序设计语言

程序设计语言是人们用来编写程序，为计算机能够接受并理解的计算机语言。它通常是一个能够完整准确地表达人的意图，并按以控制计算机实现给定运算的符号系统。它是人与计算机进行信息交流的重要工具。

程序设计语言一般分为3大类：面向机器的语言、面向过程的语言和面向对象的语言。

1. 面向机器的语言

面向机器的语言主要是针对特定的机器而设计的计算机语言。它依赖于具体的计算机，如：机器语言、汇编语言。

2. 面向过程的语言

面向过程的语言又称为结构化语言，用这类语言进行程序设计时，必须用语言的语句、函数、命令描述一步一步地解决问题的步骤，即告诉计算机“如何做”。这类语言独立于计算机，如：Pascal语言、C语言等。

3. 面向对象的语言

面向对象的语言适于解决大型的任务。用这类语言设计程序时，只需告诉计算机“做什么”，而不用说明“怎么做”。这类语言也独立于计算机，如C++等。

程序是用程序设计语言编写的，不同的程序设计语言有其不同的特点。因此，在开始编程之前要根据实际情况选择适合本系统的程序设计语言。通常可以按照如下原则选择编程语言：

① 程序设计语言是否能充分描述软件的处理功能，其中包括算法的复杂度、数据结构的要求、运行效率和性能要求等；

② 程序设计语言的适用性，其中包括语言的复杂程度及程序员和用户对这种语言的熟悉情况等。

1.2 程序设计方法

1.2.1 程序设计的一般步骤

程序设计就是针对给定问题进行设计、编写和调试计算机程序的过程。

程序设计是一种构造性技术。作为一名程序设计者，要想设计好一个程序，不仅要掌握程序设计语言本身的基本结构和语句，还要学习程序设计的方法和技巧，并通过程序设计的实践，不断地发现总结其规律性东西，从面进一步提高程序设计的能力。

一般程序设计方法步骤如下：

① 分析问题，确定解题方案。首先根据用户的具体要求进行：用户需求分析（详细而具体地理解用户要解决的问题）；数据及处理分析（已知或需要的输入数据、需要输出的数据、需要进行的处理）；可行性分析（用户提出的问题是否可解？可解的价值如何？）；运行环境分析（硬件环境和软件环境分析）。

然后在以上分析的基础上，将实际问题抽象化建立相应的数学模型，并确定解题方案。

② 确定算法。根据选取的模学模型和确定的解题方案，设计出具体的操作步骤，并可以通过流程图将确定的算法清晰、直观地表示出来。

③ 编写程序。选用较为适宜的程序设计语言，以书面形式将算法描述出来，即形成用程序设计语言编制的源程序。

④ 调试运行程序。对编写好的计算机程序进行试运行和检验，发现问题即对程序进行修改，直至得出正确的结果。

⑤ 建立文档资料。整理分析计算结果，并建立相应的文档资料（程序技术说明书、用户使用说明书等），以便维护和修改。

1.2.2 结构化程序设计方法

结构化程序设计思想产生于 20 世纪 60 年代，是随着计算机的发展，硬件成本的急剧下降，软件规模和复杂性的不断增加，而提出的一种至今仍广为使用的计算机软件开发技术。其目的是为了增强程序的易读性（容易理解），保证程序的质量，降低软件成本，从而提高软件的生产和维护的效率。

所谓结构化程序设计方法，就是“按照一组能够提高程序的易读性和易维护性的规则，进行程序设计的方法。”其要点如下。

① 程序的质量标准是“清晰第一，效率第二”。

② 程序的设计采用“自顶向下，逐步求精，模块化”的方法。

模块化，是指将一个复杂的问题或任务，分解成若干个功能单一、相对独立的小问题来进行设计，每个小问题就是一个模块。每个模块是一组由 3 种基本结构（顺序、分支、循环）组成的程序，模块一定要简单、功能独立，这样能使程序具有一定的灵活性和可靠性。

自顶向下，是指模块的划分要从问题的顶层向下逐层分解、逐步细化，直到最底层达到最简单的功能模块。

逐步求精，是指在将抽象问题分解为若干个相对独立的小问题时，要逐级地由抽象到具体、由粗到细、由表及里进行细化，直到将问题细化到可以用程序的 3 种基本结构来实施为止。

③ 程序的结构仅由顺序、分支、循环 3 种基本结构的组合、嵌套而成，且满足：

（a）每个程序模块只有一个入口和一个出口；

（b）没有死语句（永远执行不到的语句）；

（c）没有死循环（永远执行不完的无终止的循环）。

④ 程序的书写必须按一定的规范和格式进行，不能随心所欲地拼凑。应当按照“工程化”生产方式来组织软件生产，每个人都必须按照同一规划、同一方法进行工作，使生产的软件有统一的标准、统一的风格，成为“标准产品”，不仅便于推广，而且便于生产和维护。

⑤ 程序的设计风格要以好的可读性为标准，以使用程序的用户为中心，外表美观、结构清楚、语句简洁。

1.3 程序设计风格

程序设计风格是指一个程序员在程序设计过程中所表现的特点、结构、逻辑思路等习惯和技术的总称。它应包括：程序结构形式、程序正文格式（源程序文件、清楚说明、输入和输出等）、行文格式等。

程序的设计风格直接影响到程序的可读性。好的程序设计风格对于好的程序设计具有关键性作用，它不但可以提高程序设计的质量，而且可以提高程序设计的效率。

良好的程序设计风格来源于大量的实践和经验，其总的原则是清晰、易读。具体表现

在以下几个方面。

① 程序中算法。程序中采用的算法要直接了当、尽可能简单，避免使用过于复杂和技巧性强的算法。描述算法的流程图要尽可能是结构化流程图，注意保持程序结构的清晰、直观。

② 程序正文格式：程序正文格式（源程序文件编排格式）根据清晰、易读的原则，应注意以下几点。

（a）名字命名。文件名、数据名、过程名、函数名、变量名等命名，要尽可能采用有实际意义的名字命名，即达到“见名知意”。使人看了一目了然，不仅能帮助理解和记忆，而且还有助于提高程序的可读性。

例如：distance=speed*time 要比 d=s*t 更容易理解。

命名要有统一标准，如所有命名都采用汉语拼音或英文单词及单词的缩写，如：xm/name（姓名）、xb/sex（性别）、nl/age（年龄）等，但不能混合使用。不要采用过于相似的命名，以避免引起误解或输入错误。同一名字不要赋予多种含义，以避免增加阅读和修改上的困难。

（b）程序注释。程序中使用适当的注释，有助于对程序的理解，提高程序的可读性。最好的注释是简洁明了地指出程序突出的特点、功能，或提供一种概念，有助于别人理解程序。注释分为首部注释和功能注释。

首部注释：位于每个模块的前面，用于说明整个模块的功能、接口信息、程序开发状况、使用方法等。

功能注释：位于程序的内部，用于说明处理的功能。

（c）程序书写格式。程序书写格式要注意清晰、美观，具有整体效果和层次结构。通常采用一致的缩格书写形式，反映出嵌套的深度和层次，以突出程序的逻辑结构，提高程序的可读性。

③ 数据说明与语句构造。数据说明应当标准化，并按一定的次序对不同的数据结构和类型进行定义与说明。

语句构造应以简洁、直接为原则，尽量避免使用复杂的条件判断语句和多层次的循环嵌套与条件判断嵌套。表达式要采用自然形式，尽可能一目了然。同时，可以使用加括号的方式排除歧义性。

④ 输入/输出格式。输入/输出格式的设计要以方便用户使用为原则，简单、通用、规范。具体包括：术语、格式、排布方式、字体、颜色等，根据用户的不同类型、不同要求、不同特点，设计不同格式的输入/输出形式。

⑤ 文档类理。程序文档尽量使用利于阅读的行文格式，逻辑简单、条理清楚。

习　题

1.1　简述题：

（1）什么叫算法？它具有什么特性？

（2）程序设计的一般步骤是什么？

（3）提出结构化程序设计的目的是什么？其要点有哪些？

1.2　用传统流程图和 N-S 结构图分别表示以下算法：

（1）已知圆的半径，求其圆周长和面积？

此为试读，需要完整PDF请访问：www.ertongbook.com