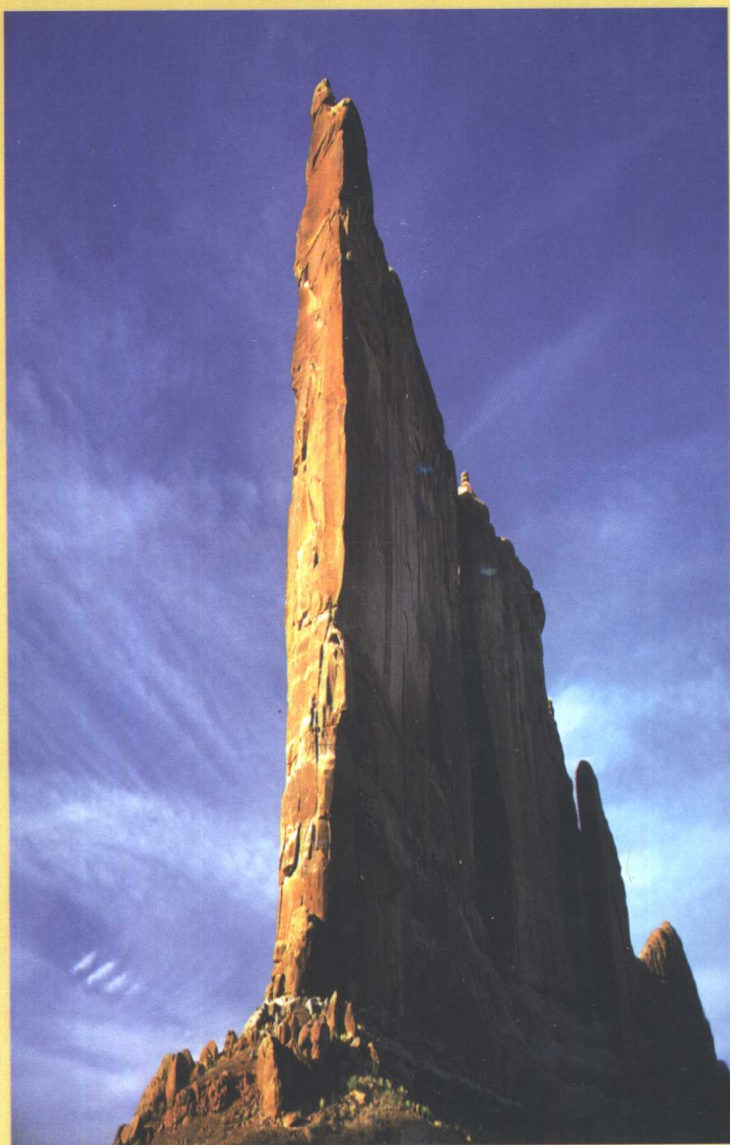




中华人民共和国教育部考试中心
全国计算机应用技术证书考试 (NIT)

程序设计

(QBASIC 语言) 教程



教育部考试中心 组编
清华大学出版社

<http://www.tup.tsinghua.edu.cn>



中华人民共和国教育部考试中心
全国计算机应用技术证书考试(NIT)

程序设计(QBASIC 语言)教程

教育部考试中心 组编

鲍有文 周海燕 编著
崔武子 王梅芬

清华大学出版社

(京)新登字 158 号

内 容 简 介

本书是全国计算机应用技术证书考试(NIT)程序设计模块(QBASIC 语言)的指定教材,是一本面向 QBASIC 语言程序设计初学者的培训与自学用书。全书共分 11 个单元,由浅入深地介绍了程序设计的基础知识、QBASIC 的集成环境、算术表达式、三种基本结构的程序设计方法、字符串、数组和记录、过程、文件、屏幕控制和作图等内容。每一单元均配有单元小结和练习题。书末附有程序设计模块(QBASIC 语言)的培训与考试大纲、作业设计考核表与参考示例以及上机考试题型举例。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

书 名: 程序设计(QBASIC 语言)教程

作 者: 鲍有文 等 编著

出版者: 清华大学出版社(北京清华大学学研楼,邮编 100084)

<http://www.tup.tsinghua.edu.cn>

印刷者: 世界知识印刷厂

发行者: 新华书店总店北京发行所

开 本: 787×1092 1/16 印张: 15.25 字数: 343 千字

版 次: 2000 年 5 月第 1 版 2000 年 5 月第 1 次印刷

书 号: ISBN 7-302-03853-8/TP·2250

印 数: 0001~6000

定 价: 21.50 元

JS71712

第一届全国计算机应用技术证书考试 委员会名单

(以姓氏笔画为序)

主任委员：杨学为 谭浩强

副主任委员：王建军 刘瑞挺 吴文虎 潘桂明

委 员：王成钧 王 耆 王景新 毛汉书 边奠英

刘百惠 刘长占 任威烈 求伯君 吴立德

吴功宜 苏运霖 陈 禹 杨一平 杨明福

杨炳儒 林毓材 周明德 张基温 张 森

孟志华 高 林 徐士良 徐惠民 赵鸿德

侯炳辉 裴纯礼 潘 阳

秘 书 长：潘 阳

全国计算机应用技术证书考试教材编审 委员会名单

(以姓氏笔画为序)

主任委员：杨学为 谭浩强

副主任委员：王建军 刘瑞挺 吴文虎 潘桂明

委 员：王成钧 王 耆 吴功宜 赵鸿德

侯炳辉 姜春红 高 林 徐士良

徐海涛 韩庆久 熊燕清 潘 阳

“全国计算机应用技术证书考试(NIT)”系列教材

序

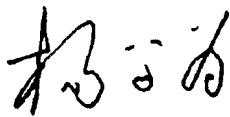
人类社会已经进入了信息时代。计算机的应用日益成为人类生活、工作、学习所必备的一种基本能力,愈来愈多的人迫切希望掌握计算机的应用技术,以符合信息时代的要求。毫无疑问,中国需要一批人掌握深奥的信息技术理论与复杂的信息技术,但是对于大多数人来说,只需要掌握实用技术就足够了。在几年前我们就注意到这种趋势,并开始了这种变革。在借鉴英国剑桥大学考试委员会举办的剑桥信息技术(CIT)的成功经验的基础上,实行以实践为主的操作培训和技能考试,这就是全国计算机应用技术证书考试(NIT)。它在系统设计上采取了一种全新的思路,首次将考试分为过程式考核、作业设计及上机考试三个阶段,以实际应用为目的,培养和测试考生在计算机应用领域的独立操作能力和应用技能。根据计算机技术发展的特点和学习者在应用领域中的需要,它采用模块化结构,在培训内容设置上紧跟计算机技术的发展,在教学过程中充分体现考生的个性,侧重于考生应用技能的培养;采用指导评估的方式进行能力考核,对考生的独立操作能力和独立解决问题能力进行综合测试。

为了规范培训和考试,我们决定选择最新和最流行的计算机应用软件,编写系列丛书,作为全国计算机应用技术证书考试的指定教材。为了体现NIT侧重培养和测试考生在计算机应用领域的独立操作能力的特点,我们改变了以往同类教材的传统写法,采用以任务驱动的方式,引导读者在完成每个任务的过程中学会相应的操作,并希望通过培训来帮助大多数人掌握计算机的应用技能。这套丛书图文并茂、循序渐进,易学易懂,有的还配有多媒体教学光盘,以帮助读者的学习。

我们邀请国内一些著名的专家编写这套丛书,他们夜以继日地紧张工作,圆满完成了任务,在此谨向他们致以衷心感谢。

由于我们缺乏经验,书中不足之处在所难免,敬请各位读者及关心我们的同志批评指正。

教育部考试中心 主任



1999年3月

前 言

本书是根据教育部颁布的《全国计算机应用技术证书考试(NIT)培训与考试大纲》编写的,是全国计算机应用技术证书考试程序设计模块(QBASIC 语言)的指定教材。

全国计算机应用技术证书考试(NIT)是教育部考试中心主办的计算机应用技能的培训与考试系统,它采用了系统化的设计、模块化的结构、个性化的教学、规范化的考试和国际化的标准,为用人单位提供了一个客观、统一和规范的标准。QBASIC 语言是学习程序设计的入门模块,是进一步学习和掌握计算机应用及程序设计(如 Visual BASIC)的基础。

为了适应 NIT 这种新型的考试模式,本书改变了传统教材的写法,采用任务驱动,任务就是学习的重点,引导学生在具体任务的操作中掌握知识,充分体现了 NIT 的教学思想。除作为 NIT 的教学及培训教材外,本书还可作为 QBASIC 语言程序设计的自学和参考用书。

本书由 11 个单元构成,每个单元又包括若干个任务。其中第一单元为程序设计的入门知识,介绍了算法与程序设计的基本概念和应用程序设计全过程的基础知识;第二单元为 QBASIC 集成环境,介绍了 QBASIC 的界面、操作及如何建立、编辑、调试、保存一个 QBASIC 的源程序(包括主程序、SUB 过程和 FUNCTION 过程);第三单元为 QBASIC 语言的算术表达式,介绍了 QBASIC 的基本数据类型及算术表达式;第四单元为顺序结构,第五单元为选择结构,第六单元为循环结构,这三个单元介绍了使用 QBASIC 语言的三种基本结构进行程序设计的方法;第七单元为字符串,介绍了字符串类型及包含字符串处理的程序设计;第八单元为数组和记录,介绍了两种用户定义的数据类型及包含应用数组与记录的程序设计;第九单元为过程,介绍了模块化程序设计的方法,全面地应用了第一至第八单元的知识;第十单元为屏幕控制与作图,介绍了包含控制屏幕和图形的程序设计;第十一单元为文件,介绍了从实用角度应用顺序文件和随机文件的程序设计。最后,在附录中给出了程序设计模块(QBASIC 语言)的培训与考试大纲和作业设计等内容,以供准备参加考试的读者参考。

本书由教育部考试中心教育测量学术交流中心组织编写,由鲍有文副教授主编。本书的第六、七单元和附录由鲍有文编著,第二、四、五单元由周海燕编著,第八、十一单元由崔武子编著,第一、三、九、十单元由王梅芬编著。在编写过程中,我们得到了谭浩强教授、高林教授的指导,还得到了教育部考试中心和清华大学出版社的许多同志的帮助,在此一并表示诚挚的谢意。

由于编者的知识及水平有限,本书一定存在不少缺点和不足,敬请广大读者指正。我们将在不断积累经验的基础上,根据 NIT 考试的发展情况,适时修改再版。

编 者

1999 年 11 月

目 录

第一单元 程序设计的入门知识	1
任务一 算法设计与描述.....	1
一、算法设计	1
二、常用的算法描述方法	3
任务二 程序设计及其基本原则.....	6
一、程序与程序设计	6
二、先算法设计与描述,后程序设计	6
三、模块化	6
四、结构化	7
任务三 一个小应用程序的程序设计过程.....	8
一、问题分析	8
二、算法设计	9
三、程序设计	10
任务四 程序的调试和测试	10
一、查错与纠错	10
二、测试与调试	11
任务五 建立程序文档	12
一、用户使用说明书	12
二、技术文档	12
任务六 程序设计风格	12
一、清晰第一,效率第二	12
二、变量使用得当	13
三、适当的注释	13
四、锯齿形的书写格式	13
五、友好的用户界面	13
六、有一定的容错功能	13
单元小结	13
练习题	13
第二单元 QBASIC 集成环境	15
任务一 启动和退出 QBASIC	15

一、启动条件	15
二、启动方法	15
三、工作窗口的组成	15
四、窗口操作	17
五、退出 QBASIC	17
任务二 程序的输入、保存和运行	18
一、输入一个新程序	18
二、保存源程序	19
三、程序的运行	21
任务三 源程序的打开、编辑和打印输出	22
一、打开一个已存在的源文件	22
二、源文件的编辑	23
三、源文件的打印输出	25
任务四 QBASIC 中有关独立模块的操作	25
一、建立一个 SUB 过程	25
二、建立一个 FUNCTION 过程	27
三、主程序和过程之间的切换	27
四、在屏幕上同时显示两个模块的方法	28
任务五 简单程序调试	29
一、利用联机帮助查找信息	29
二、借助 PRINT 或 STOP 语句查错	29
三、单步执行	30
四、设置断点	30
单元小结	31
第三单元 QBASIC 语言的算术表达式	32
任务一 QBASIC 语言的特点和 QBASIC 源程序的结构	32
一、QBASIC 语言的特点	32
二、QBASIC 源程序的结构	33
任务二 QBASIC 的数据描述	35
一、QBASIC 的数据类型	35
二、常量和算术常量	35
三、变量和算术变量	37
四、符号常量	40
任务三 标准函数	41
一、标准函数	41
二、算术标准函数	41
任务四 运算符和表达式	43
一、运算符和算术运算符	43

二、表达式和算术表达式	43
任务五 掌握 QBASIC 语言的算术表达式的设计	44
单元小结	45
练习题	46
第四单元 顺序结构	48
任务一 掌握为变量提供数据的语句	48
一、赋值语句(LET 语句)	48
二、键盘输入语句(INPUT 语句)	49
三、读数/置数和恢复数据指针语句(READ/DATA/RESTORE 语句)	51
四、三种提供数据的语句的比较	53
任务二 掌握数据输出语句的使用	54
一、PRINT 语句	54
二、LPRINT 语句	57
任务三 了解其他常用语句的功能	57
一、终止程序执行语句(END 语句)	57
二、暂停程序执行语句(STOP 语句)	58
三、变量内容交换语句(SWAP 语句)	58
四、注释语句	58
任务四 掌握顺序结构程序设计的方法	59
单元小结	61
练习题	61
第五单元 选择结构	63
任务一 掌握程序控制条件的描述方法	63
一、逻辑量的概念	63
二、关系运算符和关系表达式	63
三、逻辑运算符和逻辑表达式	64
任务二 设计两分支的选择结构	65
一、使用行 IF 语句	65
二、使用块 IF 语句	67
任务三 设计多分支的选择结构	68
一、使用嵌套的行 IF 语句	68
二、使用嵌套的块 IF 语句	69
三、使用带 ELSEIF 子句的块 IF 语句	70
四、使用 SELECT CASE 语句	71
任务四 掌握选择结构的程序设计方法	75
单元小结	78
练习题	78

第六单元 循环结构	81
任务一 FOR-NEXT 语句的使用	81
任务二 WHILE-WEND 语句的使用	84
任务三 DO-LOOP 语句的使用	85
一、DO-LOOP 语句的格式和执行过程	85
二、DO 循环的应用举例	88
任务四 设计多重循环	89
任务五 使用循环结构实现常用算法的程序设计	91
单元小结	95
练习题	95
第七单元 字符串	100
任务一 字符串运算量的使用	100
一、字符串常量	100
二、字符串变量	101
任务二 字符串的基本操作	101
一、字符串变量的赋值	102
二、字符串的比较	103
三、字符串的连接	104
四、字符的置换	104
任务三 掌握常用的字符串处理库函数	105
一、求字符串长度的函数(LEN)	105
二、子串处理函数(LEFT\$、RIGHT\$、MID\$)	106
三、字符串与数值之间的转换函数(VAL、STR\$)	107
四、字符与 ASCII 码之间的转换函数(ASC、CHR\$)	108
五、大写字母与小写字母之间的转换函数(LCASE\$、UCASE\$)	109
六、产生由相同字符组成的字符串函数(STRING\$、SPACE\$)	109
任务四 掌握包含字符串处理的程序设计	110
单元小结	113
练习题	113
第八单元 数组和记录	116
任务一 数组的定义	116
一、一维数组的定义	116
二、二维数组的定义	118
任务二 数组元素的引用	119
一、一维数组元素的引用	119
二、二维数组元素的引用	121
任务三 字符串数组的使用	123

一、字符串数组的定义	124
二、字符串数组元素的引用	124
任务四 记录的使用	125
一、记录类型的定义	125
二、记录类型变量的定义	125
三、记录类型变量的引用	126
任务五 掌握包含数组和记录的程序设计	127
一、包含一维数组的程序设计举例	127
二、包含二维数组的程序设计举例	130
三、包含字符串数组的程序设计举例	133
四、包含记录类型数据的程序设计举例	134
单元小结	135
练习题	136
第九单元 过程	139
任务一 函数过程(FUNCTION 过程)	139
一、函数过程的定义	139
二、函数过程的引用	141
任务二 子程序过程(SUB 过程)	142
一、子程序过程的定义	142
二、子程序过程的调用	143
三、函数过程与子程序过程的比较	144
任务三 关于过程调用的讨论	144
一、虚实结合	144
二、过程的嵌套调用	151
三、过程的递归调用	154
四、全局变量和局部变量	155
任务四 模块内函数(DEF 语句)	157
任务五 掌握包含过程调用的程序设计	160
单元小结	163
练习题	165
第十单元 屏幕控制与作图	168
任务一 屏幕控制的两个要素	168
一、屏幕显示模式	168
二、屏幕坐标系	169
任务二 屏幕控制的基本操作	169
一、定义屏幕显示模式(SCREEN 语句)	169
二、设置显示颜色(COLOR 语句)	170

三、清除整个屏幕(CLS 语句).....	172
四、确定屏幕的行列数(WIDTH 语句)	172
五、控制光标定位(LOCATE 语句)	172
六、保存当前光标的位置(CSRLIN 和 POS 函数)	173
任务三 基本作图类型.....	173
一、画点(PSET 语句和 PRESET 语句)	173
二、画线、矩形框、矩形块(LINE 语句)	175
三、连续画直线(DRAW 语句)	176
四、画圆、椭圆、圆弧、扇形(CIRCLE 语句)	177
任务四 图形的着色(PAINT 语句)	180
任务五 视窗的设计.....	181
一、开视窗(VIEW 语句)	181
二、重新定义坐标系(WINDOW 语句)	182
任务六 设计作图程序.....	183
单元小结.....	186
练习题.....	187
第十一单元 文件	189
任务一 了解文件的概念.....	189
一、文件的概念	189
二、文件的分类	189
三、文件的读写和文件缓冲区	190
四、文件的打开与关闭	190
五、文件指针和文件结束标志	190
任务二 顺序文件的使用.....	190
一、顺序文件的建立、打开与关闭	191
二、顺序文件的读和写	192
任务三 随机文件的使用.....	197
一、随机文件的建立、打开与关闭	198
二、随机文件的读和写	198
三、记录变量的读写操作	201
任务四 与文件和目录有关的操作语句的使用.....	203
任务五 掌握包含文件的程序设计.....	205
一、包含顺序文件的程序设计举例	205
二、包含随机文件的程序设计举例	206
单元小结.....	207
练习题.....	207

附录	全国计算机应用技术证书考试(NIT)培训与考试大纲	
	——程序设计模块(QBASIC 语言)	208
附件一	全国计算机应用技术证书考试(NIT)学员评估记录表	
	——程序设计模块(QBASIC 语言)	212
附件二	全国计算机应用技术证书考试(NIT)作业设计考核表	
	——程序设计模块(QBASIC 语言)	214
附件三	全国计算机应用技术证书考试(NIT)作业设计参考示例	
	——程序设计模块(QBASIC 语言)	216
附件四	全国计算机应用技术证书考试(NIT)上机考试题型举例	
	——程序设计模块(QBASIC 语言)	222

第一单元

程序设计的入门知识

计算机的应用技术越来越普及,许多应用软件的使用已非常方便,这是因为计算机软件厂商已事先编好了程序。如果一个计算机应用人员有了程序设计的概念,懂得了程序设计的原理,并且会编写程序,那么他就能更好地使用计算机。所以,程序设计是一个计算机应用人员的基本功,而 QBASIC 语言又是初学程序设计者的最佳选择。

学习本单元可帮助初次接触程序设计的读者对程序设计有些入门的认识。

任务一 算法设计与描述

一、算法设计

1. 算法

算法这个概念实际上是被广泛应用的。例如,某班学生星期一上课的问题,课程表告诉学生这样做:1、2 节课到 101 教室上“数据结构”=>3、4 节到 102 教室上“C 语言”=>5、6 节到微机实验室做实验=>7、8 节自由活动=>晚上没有课。其实课程表就是解决上面这个问题的算法。当然,我们还可以把上面的问题再进一步细化,例如,如何到 101 教室的问题等等。算法就是解决一个特定问题所采用的确定的、有限的方法和步骤。解决一个问题的方法和步骤也可以多种多样,就是说算法一般不是惟一的,例如,到 101 教室的路线就可以有好几种方案,但总有一个最好的方案。也就是说人们在解决问题时总是力图去寻找一个最好的算法。某个学生提出了到 101 教室的路线图,我们就说该学生设计了一个算法,学生按这个路线图走到了 101 教室,我们就说学生实现了算法。在本书中,我们讨论的是让计算机可以执行的算法,这种算法称为计算机算法。例如,计算机求阶乘、求若干实验数据的平均值等等。在有关计算机的书中,我们常把计算机算法简称为算法。在本书以后的单元中,将介绍许多典型的算法,请读者掌握这些算法。

2. 算法设计

我们来看一个求 n 的阶乘 $n!$ 的算法设计。

例 1.1 计算 $5!$,即计算 $1 \times 2 \times 3 \times 4 \times 5$ 。

问题分析:在计算 $5!$ 的过程中,可以发现有两个特点:

(1) 连续进行乘法(在算法中称作“累乘”),即前一次的乘积乘以一个数得本次乘积,

然后继续重复这个乘法过程；

(2) 乘数的变化是有规律的，每做一次乘法后，乘数就增加 1 成为新的乘数。

为解决这个问题，我们设计的算法可以描述为：

(1) 用 fact 表示累乘的积，用 n 表示乘数；

(2) 给 fact 赋初值为 1，给 n 赋初值为 2，常表示为 $1 \Rightarrow \text{fact}, 2 \Rightarrow n$ ；

(3) 进行 $\text{fact} * n$ 的乘法运算，并把乘积赋给 fact，表示为 $\text{fact} * n \Rightarrow \text{fact}$ ；

(4) 使乘数增 1，表示为 $n+1 \Rightarrow n$ ；

(5) 判断 n 的值，当 $n < 5$ 或 $n = 5$ 时，返回到第 (3) 步，重复执行 (3)~(5) 步，当 $n > 5$ 时，执行第 (6) 步；

(6) 输出 fact 的值，此时 fact 的值就是 5!。

以上是计算 5! 的算法，而解决这个问题的算法不止这一种，例如，我们可以通过计算 $5 \times 4 \times 3 \times 2 \times 1$ 或计算 $5 \times 4!$ 等方法来计算 5!。所以说，一个问题的算法不是惟一的。设计算法的过程称为算法设计。

3. 算法的特性

从例 1.1 中我们可以看到，算法有以下的特性：

(1) 确定性。算法中的每一步骤必须是明确的做法，而不是随便怎么做都行，即不可以有“多义性”或“歧义性”。例如，将第 (4) 步的“使乘数增 1”改为“使乘数增值”，那么如果增的值是多少没有确定，就会导致多种可能性。

(2) 有效性。算法中的每一步骤必须是计算机可以有效执行的，并能产生有效的结果。例如，我们不能让计算机表达一个它表达不了的数。如果一台计算机可表示的最大正整数为 32767，那么用整数表示阶乘的结果，就只可以表示到 7!，表示 8 或 8 以上的阶乘就会出现一些无效的结果。又例如，计算机只能表示有限的数，被零除在数学上是成立的，然而，在计算机上是无法实现的。

(3) 有穷性。算法中的步骤必须是有限次数的。例如，我们可以计算 5! 或 7!，但是，我们不能计算 $n!$ ，因为前者的计算步骤是有限次数的，后者的计算步骤是无限次数的。

(4) 一个或多个输出。算法的最终目的必须是把一个或多个结果输出给用户，否则算法没有什么实际意义。例如，计算机必须把 5! 是 120 作为结果输出给用户看，否则，用户怎么知道算法计算 5! 后有没有结果？结果又是多少呢？

(5) 零个或多个输入。在执行算法时，可以从程序外获取数据，称为输入。有的算法也可以不必从程序外获得什么值。例如，例 1.1 中没有输入的值。但如果我们修改一下例 1.1 中算法的几个步骤：

在第 (1) 步里加上“用 m 表示求某数的阶乘的某数”；

在第 (2) 步之后增加上一步“从键盘上输入一个值给 m，表示为 $\text{值} \Rightarrow m$ ”；

在第 (5) 步里把“ $n < 5$ 或 $n = 5$ ”改为“ $n < m$ 或 $n = m$ ”；

在第 (6) 步里把“5!”改为“m 的阶乘”。