

高等学校计算机基础教育教材

从QBasic 到Visual Basic 实用教程



刘士杰 主编



清华大学出版社
<http://www.tup.tsinghua.edu.cn>



高等学校计算机基础教育教材

从 QBasic 到 Visual Basic 实用教程

刘士杰 主编

清华大学出版社

(京)新登字 158 号

内 容 简 介

本书由 QBasic 程序设计和 Visual Basic 程序设计两大部分组成。在 QBasic 部分中,重点介绍了结构化程序设计的基础知识,顺序、分支、循环结构的程序设计方法,数组、子程序、过程的应用,作图及文件的相关知识;在 Visual Basic 部分中,重点介绍了面向对象的程序设计的基本概念和基础知识,控件的使用和应用程序的界面设计,菜单、工具栏的使用,图形设计与多媒体应用程序以及文件和数据库的相关知识。

本书避免了繁琐的理论及概念的阐述,突出了基础性、先进性和实用性。本书具有通俗易懂、易学易用、结构合理、图文并茂等特点,既可作为高校非计算机专业学生的计算机语言课教材,也可作为各类工程技术人员、办公人员学习计算机语言的培训教材及参考书。

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: 从 QBasic 到 Visual Basic 实用教程

作 者: 刘士杰 主编

责任编辑: 欧振旭

出 版 者: 清华大学出版社(北京清华大学学研大厦,邮编 100084)

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

印 刷 者: 清华大学印刷厂

发 行 者: 新华书店总店北京发行所

开 本: 787×1092 1/16 印张: 22.5 字数: 542 千字

版 次: 2002 年 1 月第 1 版 2002 年 1 月第 1 次印刷

书 号: ISBN 7-302-04837-1/TP·2866

印 数: 0001~5000

定 价: 29.00 元

前 言

高等学校非计算机专业学生的计算机教育是高等教育的组成部分，是实现高等教育培养目标的一个重要环节。高等学校的学生应具有较高的程序设计能力和应用计算机的能力。

随着计算机教学改革的不深入，我们对传统的计算机语言教学进行了改革，放弃了过去非计算机专业的学生只学一门面向过程的结构化语言，或面向对象的可视化语言的做法，将两者有机地结合起来。我们认为，非计算机专业的学生最好能先学习一门面向过程的结构化程序设计语言，掌握结构化程序设计的基本思想和方法，为进一步学习可视化程序设计和进行计算机应用奠定坚实的基础；然后，学习一门可以开发 Windows 环境下应用程序的可视化语言，如：Visual Basic，Visual C，Visual Foxpro 等。基于这一思想，我们编写了这本《从 QBasic 到 Visual Basic 实用教程》，其目的是满足高等学校非计算机专业学生学习计算机语言的需要。

本书首先讲述了面向过程的结构化程序设计语言 QBasic 的使用方法，并在此基础上，进一步介绍了可视化的程序设计语言 Visual Basic 的使用方法。通过两年的教学实践，我们体会到这样做，不仅克服了学生只学 QBasic 不能掌握和应用先进的可视化程序设计编程的弊病，又克服了学生只学 Visual Basic 带来的不理解程序的结构和编写程序代码困难的问题，收到了良好的教学效果。此外，由于 QBasic 与 Visual Basic 兼容，因此，这样学习，可以将两者有机地结合起来，为本专业的进一步学习和应用程序的开发打下良好的基础。

本书参考了全国计算机等级考试（二级）中 QBasic 的大纲要求，结合 QBasic 与 Visual Basic 中的重点内容，编写了这本具有基础性、先进性、实用性三大特点的教材，可供高等学校非计算机专业，特别是理工科专业的学生，以及各类办公人员，工程技术人员学习和使用。

本书避免了繁琐的理论阐述，强调以应用为主，加强了基础知识、基本技能的训练，加大了实用内容和新内容的篇幅。本书具有内容新颖，易学易用，通俗易懂，图文并茂等特点，并通过典型实例的讲解，使学生能很好地掌握所学的知识和操作要领。各章后附有习题和上机操作题，便于学生课后学习和上机操作。

本书由刘士杰任主编。其中第 1 章、第 5 章由刘士杰编写；第 2 章由张瑾编写；第 3 章、第 6 章由邹光华编写；第 4 章由邹光华、刘士杰编写；第 7 章由孙改平编写；第 8 章由王秀和、刘士杰编写；第 9 章、第 12 章由李耀辉编写；第 10 章由席振元编写；第 11 章由曹德胜编写。参加编写的人员还有赵彦珠、高晓燕。

由于作者水平有限，书中难免存在缺点和错误，殷切希望读者批评指正。

作 者

2001 年 6 月

目 录

第 1 章 QBASIC 程序设计基础	1
1.1 程序设计与 QBASIC 简介	1
1.1.1 程序设计概述	1
1.1.2 算法与流程图	2
1.1.3 QBASIC 简介	6
1.2 QBASIC 程序的基本结构	6
1.2.1 一个简单的 QBASIC 程序	6
1.2.2 QBASIC 程序行	8
1.2.3 QBASIC 的关键字	9
1.2.4 复杂程序的结构	9
1.3 QBASIC 程序设计的基础知识	10
1.3.1 QBASIC 的基本数据类型	10
1.3.2 常量	12
1.3.3 变量	13
1.3.4 表达式与运算符	16
1.3.5 标准函数	21
1.3.6 几个常用的基本语句	28
1.4 QBASIC 的运行环境	34
1.4.1 QBASIC 的启动与退出	34
1.4.2 QBASIC 的功能菜单	35
1.4.3 QBASIC 程序的编辑和运行	37
习题	39
上机操作题	42
第 2 章 基本控制结构的程序设计	44
2.1 顺序结构程序设计	44
2.1.1 顺序结构程序设计的概念	44
2.1.2 读数、置数语句(READ...DATA)	44
2.1.3 恢复数据指针语句(RESTORE)	46
2.1.4 交换变量值语句(SWAP)	48
2.1.5 程序暂停语句(STOP)	49
2.1.6 清屏语句(CLS)	49

2.1.7 注释语句(REM)	49
2.1.8 几种赋值语句的比较	50
2.1.9 程序举例	51
2.2 选择结构程序设计	53
2.2.1 选择结构程序设计的概念	53
2.2.2 单行结构 IF 语句	53
2.2.3 块结构 IF 语句	55
2.2.4 单条件的嵌套语句(IF...ELSEIF...END IF)	57
2.2.5 多分支选择语句(SELECT CASE)	59
2.3 循环结构程序设计	64
2.3.1 循环结构程序设计的概念	64
2.3.2 固定次数的循环语句(FOR...NEXT)	64
2.3.3 条件循环语句(WHILE...WEND)	71
2.3.4 先判断条件的 DO...LOOP 循环语句	74
2.3.5 先执行循环, 再判断条件的 DO...LOOP 循环语句	76
2.3.6 三种循环结构的比较	79
2.3.7 多重循环	80
2.4 综合习题分析	82
习题	85
上机操作题	90
第 3 章 QBasic 数组	92
3.1 数组和数组单元的概念	92
3.1.1 数组的概念	92
3.1.2 数组单元	92
3.1.3 数组的维数	93
3.1.4 数组类型的说明	93
3.2 数组的建立	93
3.3 一维数组和二维数组	95
3.3.1 一维数组	95
3.3.2 二维数组	96
3.4 静态数组和动态数组	97
3.4.1 静态数组和动态数组	97
3.4.2 数组的释放语句(ERASE)	98
3.4.3 重新定维语句(REDIM)	99
3.5 数组应用举例	100
3.5.1 查找应用举例	100
3.5.2 排序应用举例	102

习题	103
上机操作题	105
第 4 章 子程序与过程	107
4.1 自定义函数	107
4.1.1 定义格式	107
4.1.2 自定义函数的调用	108
4.2 子程序	109
4.2.1 子程序的定义与调用	109
4.2.2 开关语句(ON GOSOB-RETURN 语句)	110
4.3 过程	111
4.3.1 变量的作用域	111
4.3.2 说明类语句	114
4.3.3 SUB 过程	115
4.3.4 FUNCTION 过程	117
4.3.5 过程的参数传递和递归调用	120
习题	123
上机操作题	126
第 5 章 屏幕控制与作图	128
5.1 屏幕显示模式与屏幕控制	128
5.1.1 屏幕显示模式简介	128
5.1.2 屏幕控制语句	129
5.1.3 显示模式控制语句(SCREEN)	130
5.1.4 设置颜色语句(COLOR)	132
5.2 画图	135
5.2.1 几种基本画图语句	135
5.2.2 填充颜色语句(PAINT 语句)	142
5.2.3 综合举例	143
习题	144
上机操作题	146
第 6 章 文件	149
6.1 文件基本概论	149
6.1.1 文件说明	149
6.1.2 记录	149
6.1.3 文件的分类	150
6.1.4 文件号	151

6.2 文件操作语句和函数.....	151
6.2.1 打开文件语句(OPEN).....	151
6.2.2 关闭文件语句(CLOSE).....	152
6.2.3 文件操作语句.....	152
6.2.4 文件操作函数.....	154
6.3 顺序文件.....	155
6.3.1 顺序文件的输出(顺序写文件).....	155
6.3.2 顺序文件的输入(顺序读文件).....	156
6.3.3 顺序文件中数据的添加、检索、排序和修改.....	158
6.4 随机文件.....	161
6.4.1 随机文件的输出(随机写文件).....	161
6.4.2 随机文件的输入(随机读文件).....	162
习题.....	164
上机操作题.....	165
第 7 章 Visual Basic 编程基础.....	168
7.1 Visual Basic 概述.....	168
7.1.1 什么是 Visual Basic.....	168
7.1.2 Visual Basic 的特点.....	169
7.1.3 软硬件要求.....	170
7.1.4 Visual Basic 版本简介.....	170
7.2 Visual Basic 的集成开发环境.....	171
7.2.1 运行 Visual Basic 中文版.....	171
7.2.2 Visual Basic 的窗体设计器.....	171
7.2.3 工程资源管理器.....	172
7.2.4 工具箱窗体.....	174
7.2.5 属性窗口.....	174
7.2.6 代码窗口.....	175
7.2.7 窗体布局窗口.....	178
7.3 建立第一个 Visual Basic 应用程序.....	178
7.3.1 简单应用程序设计.....	178
7.3.2 Visual Basic 的相关概念.....	186
7.3.3 输入、输出对话框.....	189
7.3.4 QBasic 与 Visual Basic 中的部分区别.....	192
7.4 调试与错误处理.....	198
7.4.1 程序错误.....	198
7.4.2 Visual Basic 调试工具.....	200
7.4.3 错误捕获及处理.....	201

习题	203
上机操作题	204
第 8 章 设计应用程序的界面	205
8.1 设计窗体	205
8.1.1 窗体的建立、添加和删除	205
8.1.2 窗体的属性和事件	206
8.1.3 向窗体中添加控件	210
8.1.4 启动窗体的设置	211
8.1.5 窗体的生命周期	213
8.2 Visual Basic 中常用的控件	214
8.2.1 Visual Basic 的控件简介	214
8.2.2 Label(标签)	215
8.2.3 TextBox(文本框)	217
8.2.4 CommandButton(命令按钮)	222
8.2.5 CheckBox(复选框)	226
8.2.6 OptionButton(单选钮)	228
8.2.7 Frame(框架)	229
8.2.8 ScrollBar(滚动条)	232
8.2.9 ListBox(列表框)	236
8.2.10 Combox(组合框)	242
8.2.11 Image(图像框)	244
8.2.12 PictureBox(图片框)	245
8.2.13 Timer(定时器)	246
8.3 Visual Basic 的 ActiveX 控件	248
8.3.1 如何向工具箱中添加 ActiveX 控件	248
8.3.2 用户 ActiveX 控件的创建	249
8.4 设计多文档界面(MDI)	251
8.4.1 界面样式	251
8.4.2 多文档界面(MDI)中窗体间的关系	251
8.4.3 创建 MDI 应用程序的方法与步骤	251
8.5 发行 Visual Basic 应用程序	253
8.5.1 创建安装程序	253
8.5.2 使用 Visual Basic 安装向导	254
习题	256
上机操作题	256

第 9 章 设计菜单和工具栏	260
9.1 菜单编辑器窗口简介	260
9.2 设计菜单	262
9.2.1 菜单栏设计	262
9.2.2 添加代码	264
9.2.3 快捷菜单的建立	266
9.2.4 运行时操作菜单	270
9.3 设计工具栏	272
9.3.1 工具栏的常用属性、方法和事件	272
9.3.2 创建工具栏	274
9.3.3 为工具栏的各按钮编写处理过程	276
习题	277
上机操作题	278
第 10 章 设计图形和多媒体程序	279
10.1 图形控件介绍	279
10.2 线条和形状控件	279
10.2.1 线条控件(Line)	279
10.2.2 形状控件 (Shape)	281
10.3 图像框和图片框控件	282
10.3.1 图像框控件	282
10.3.2 图片框控件	287
10.4 图形控件的应用	292
10.4.1 建立绘制正弦曲线的程序	292
10.4.2 图形查看实例	293
10.5 使用 Animation 控件	296
10.5.1 Animation 控件的相关属性、方法和事件	297
10.5.2 使用 Animation 播放 AVI 文件的方法	297
10.6 多媒体集成控件 MCIWnd	298
10.6.1 MCIWnd 控件的属性和事件	298
10.6.2 使用 MCIWnd 控件制作播放器	299
10.6.3 完善媒体播放器	300
10.7 利用 API 函数编写多媒体应用程序	301
10.7.1 如何添加 API 函数	301
10.7.2 利用 mciExecute() 函数建立简单的多媒体程序	302
习题	303
上机操作题	304

第 11 章 创建文件应用程序	305
11.1 文件系统概述	305
11.2 文件的读写	305
11.2.1 顺序文件	305
11.2.2 随机文件	309
11.2.3 二进制文件	318
11.3 与文件相关控件的介绍	319
11.3.1 驱动器列表框	320
11.3.2 目录列表框	320
11.3.3 文件列表框	321
习题	322
上机操作题	323
第 12 章 数据库技术	324
12.1 用数据控件访问数据库	324
12.1.1 数据库的基本概念	324
12.1.2 使用数据控件访问数据库	325
12.1.3 用 Data 控件创建简单的数据库程序	326
12.1.4 使用 DbGrid 浏览数据库	327
12.1.5 记录的自动添加	329
12.2 使用可视化数据管理器	330
12.2.1 创建新的数据库及表	330
12.2.2 数据库记录的添加、删除和查询	333
12.3 用程序代码操作和创建数据库	334
12.3.1 用程序代码操作数据库	335
12.3.2 用程序代码创建数据库	338
12.4 结构化查询语言 (SQL) 简介	340
12.4.1 SQL 语言的组成	340
12.4.2 利用 SQL 语言进行查询	341
习题	343
上机操作题	343
附录 常用字符与 ASCII 码对照表	344

第1章 QBasic 程序设计基础

1.1 程序设计与 QBasic 简介

1.1.1 程序设计概述

1. 计算机的工作原理与程序的概念

计算机工作时，只有当用户向计算机输入一定的信息，它才能按照用户的要求进行工作，并得到所需要的结果。目前的计算机的运行机制基本是：

(1) 将需要计算机完成的任务编成一条条指令，输入计算机，存放在计算机的内存储器中，称为编程序。

(2) 计算机工作时，从内存中取出指令，然后再执行它们，称为运行程序。

因此，计算机是通过一条一条的指令来完成工作的。人们以一条一条的指令来控制计算机，使它按照人们的要求工作。用计算机术语来说，指令的序列被称为程序。也可以说，程序是为解决某一问题或完成某项任务的指令的有序集合。只有当用户向计算机输入一定的程序，计算机才能按照程序规定的步骤进行工作。失去程序控制，计算机将无法发挥其作用。

2. 程序设计的基本步骤

人们通过程序让计算机工作，以便处理各种信息，解决各种问题，让计算机为人们服务，所以“程序”是人们意志的体现。编程序中用户通过程序设计语言对信息进行加工、处理并输出预期的结果。编制程序的基本过程如下：

(1) 分析问题。为了解决一定的实际问题，编程序时不能无的放矢，而要认真考察和分析实际问题，确定有哪些给定的数据，需要进行哪种处理，需要输出哪些数据等等，找出解决问题的大致思路。

(2) 提出算法。把第(1)步中分析问题的思路进一步明确化、详细化，建立解决问题所需要的数学或物理模型。

(3) 编写程序。根据第(2)步的方案用一定的计算机语言把程序编写出来。

(4) 上机调试。先录入程序，然后对编好的程序运行调试，进行实际检验。调试过程中，对其中的错误不断加以改正，直到程序能达到预期的结果为止。

(5) 运行程序。将程序投入运行，并输出结果。

1.1.2 算法与流程图

在系统地学习 QBasic 语言程序设计知识之前,先介绍算法与流程图,它们是程序设计的重要工具。

1. 算法

算法就是对问题求解方法的精确描述。在程序设计时,最关键的问题是算法的提出,因为它直接关系到编写出来的程序的正确性、可靠性和完整性。一个计算机能实现的算法应具有以下特征:

(1) 有穷性。一个算法应包含有限的操作步骤,而不能是无限的。因此,在算法中必须给出结束的条件。

(2) 确定性。一个算法中的任何步骤都必须意义明确,不能模棱两可、含糊不清,即不允许有二义性。

(3) 可行性。所采用的算法必须能在计算机上执行,因此,在算法中所有的运算必须是计算机能够执行的基本运算。例如, $A \neq 0$, $B = 0$, 那么“ A / B ”就是一个不可执行的操作,算法中不应出现。

(4) 有一定的输入与输出。要计算机解决问题时,总是需要输入一些原始数据;计算机向用户报告结果时,总是要输出一些信息。因此,一个算法中必须有一定的输入与输出。

2. 流程图

流程图是一种用图形、文字和符号来比较形象地描述“算法”的工具。它对于编制程序很有帮助。

(1) 流程图的图形符号

流程图又称为框图,它是由几种不同的图形符号组合而成的。流程图常用图形符号如图 1-1 所示。

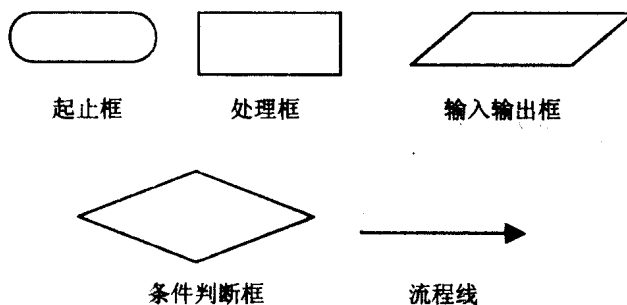


图 1-1 流程图常用图形符号

图中各图形表示的意义如下:

① 起止框,它代表一个算法的开始与结束之处。

② 处理框, 它表示算法中对问题处理的一个或若干个步骤, 如: 赋值、计算等, 这些步骤不涉及输入与输出。它用矩形框表示。

③ 输入输出框, 它表示算法中需要进行输入或输出处理的步骤。它用平行四边形表示。

④ 条件判断框, 它表示算法中需要根据条件判断的步骤。它用菱形框表示。

⑤ 流程线, 它表示算法的每一个步骤之间的先后顺序, 标示着程序的走向。

(2) 流程图的作用

流程图在程序设计中起着重要的作用, 主要有以下几点:

① 流程图既是程序设计者解决问题时逻辑思维的具体反映, 也是计算机算法的形象描述。

② 用流程图描述程序的设计思想, 可以使设计思想明确、具体、直观和更精确化。

③ 对于程序设计人员来说, 有了流程图, 思路清楚、层次分明, 可以减少编程中出现的错误, 而且也有利于程序的检查和修改。

④ 有利于程序的交流。由于流程图直观性强, 便于阅读(一般来说, 看懂一个程序的流程图, 要比看懂一个程序容易得多), 因此, 借助于流程图, 可以较快地弄清别人编制的程序。

⑤ 适应性强。流程图所表示出来的是程序的计算步骤和流程, 各框内都是用文字或符号表示程序的功能。它只是程序的骨架, 并不是程序的一部分。因此, 它适于在不同的机器上用不同的语言进行程序设计。

对于复杂问题, 在编写程序前都要先画出流程图。正确地使用流程图, 可以使程序的设计思想更加清晰和条理化, 从而有助于开发出高质量的程序来。

例 1. 用流程图描述如下算法: 向计算机输入两个不相等的数 A、B, 判断两数中哪个大, 输出大数。

本算法流程图如图 1-2 所示。

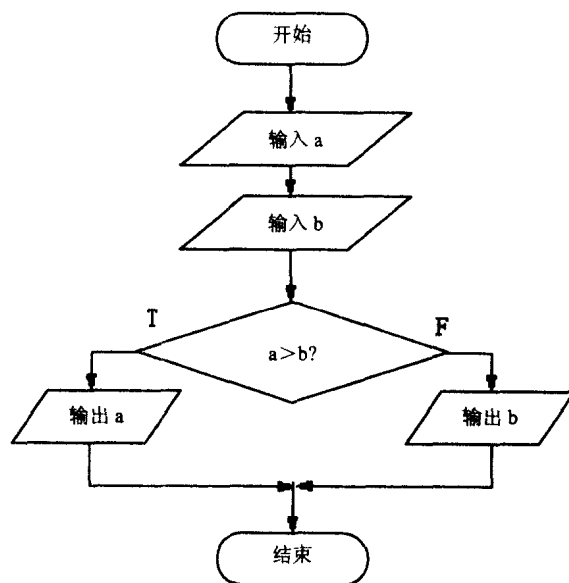


图 1-2 例 1 算法的流程图

从图 1-2 中可以看出, 本流程图从起始框开始, 表示算法开始; 用输入框输入两个不相等的数 A、B; 向下用条件判断框判断“ $A > B$ ”这个条件是否成立, 若成立, 则 A 为大值, 输出 A 的值, 否则不成立, B 为大值, 输出 B 的值; 最后流程线达到终止框, 表示算法结束。

3. N-S 流程图

前面介绍的流程图是传统流程图。由于它有一个流程线($\rightarrow$), 用它可以表示 GOTO 语句。但程序在使用 GOTO 语句时, 有很大的随意性, 使用多了也可使程序结构混乱, 可靠性降低。因此, 对于结构化程序设计, 传统流程图不太适用。1973 年, I.Nassi 和 B.Schneiderman 推出了盒图, 也称 N-S 图。按盒图规定画出的算法图都是结构化的, 所以也称为结构化盒图。N-S 流程图的主要特点是全部算法由一些基本的矩形框图顺序排列而组成的一个大矩形表示, 即不允许程序任意转移, 只能顺序执行, 从而使程序结构化。

人们经过长期的实践, 将程序的基本流程归纳为三种基本的简单结构, 即顺序、选择和循环。这三种基本结构的流程图称为结构流程图的基本图或基本结构元素。对于一个算法也可认为, 无论多么简单或多么复杂, 都可以由这三种基本结构组合构造而成。

N-S 盒图的上边框表示入口, 下边框表示出口, 它的每个基本图形都符合“单入口, 单出口”的要求, 而且和结构化程序设计的基本控制结构的约定一样。本书将用 N-S 图表示算法, 下面对流程图的基本图作进一步说明。

(1) 顺序结构

顺序结构中, 各语句是按照在算法中排列的先后次序执行的。N-S 流程图的顺序结构用图 1-3b 形式表示, 它与传统流程图的图 1-3a 对应。块 a 和块 b 两个框组成一个顺序结构。它表示程序执行顺序是从块 a 到块 b。每个块可以是某个语句或语句序列。

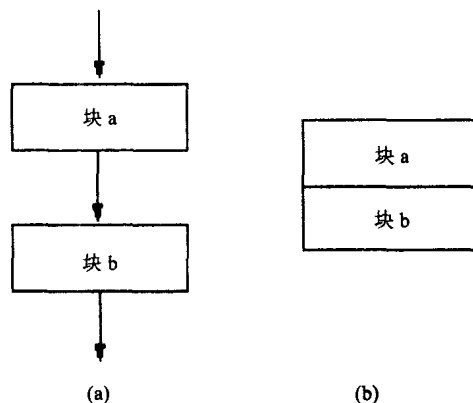


图 1-3 顺序结构流程图

(2) 选择结构

选择结构是根据某种条件选择性地执行算法的某一部分。在选择结构中先判断条件是否成立, 如果成立则执行某个语句或语句序列, 否则执行另一个语句或语句序列。选择结

构 N-S 流程图用图 1-4b 形式表示，它与传统流程图的图 1-4a 对应。当“条件”成立时执行块 a 操作，否则执行块 b 操作。注意，在执行一次选择结构时，块 a 和块 b 只有一个块被执行。

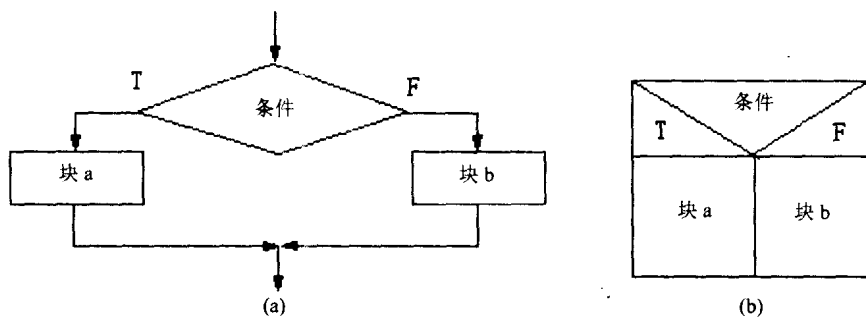


图 1-4 选择结构流程图

(3) 当型循环结构

循环结构根据某种条件重复性地执行算法所规定的某一部分，重复执行的语句或语句序列称为循环体。在进入循环前要判断所说明的条件是否成立，只有在条件成立时才执行循环体；每次循环体执行后再判断条件是否成立，条件成立则再一次执行循环体，如此循环重复直到条件不成立为止，从而达到重复性地执行算法所规定的某一部分的目的。N-S 当型循环结构用图 1-5b 形式表示，它与传统流程图的图 1-5a 对应。当“条件”成立时，重复执行块操作，直到“条件”不成立为止。它的特点是：先判断，后执行块。

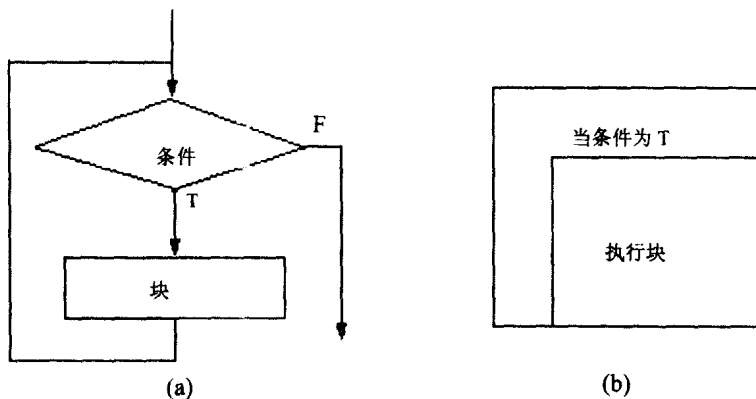


图 1-5 循环结构流程图

例 2. 用 N-S 流程图作例 1。

本例算法的 N-S 流程图如图 1-6 所示。

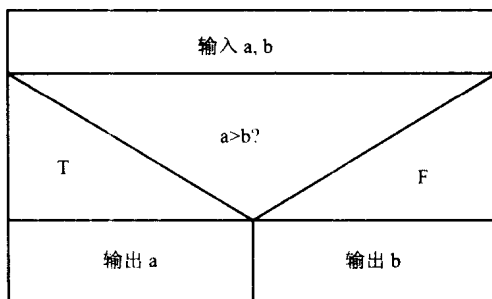


图 1-6 例 2 算法的 N-S 流程图

1.1.3 QBasic 简介

QBasic 程序语言是美国微软(Microsoft)公司开发的一种现代化、结构化的 Basic 程序设计语言。QBasic 程序设计语言随微软公司 MSDOS 5.0 及以上版本提供给用户。QBasic 语言与其他高级语言相比并不逊色，它既吸取了其他高级语言的诸多特点，又很好地保持了 Basic 语言易于学习和使用的特点，尤其适用于初学者，同时又具有很强的实用价值，这对许多熟悉 Basic 语言的用户提供了很大的方便。QBasic 是全国高等院校计算机基础教育研究会提出推广的语言，也是目前全国计算机等级考试的二级考试中规定的五种语言中的一种。

QBasic 和其他语言相比，具有以下特点：

(1) 功能很强，编程效率高。它是模块化结构程序设计语言，支持递归。它适用于科学计算，又有丰富的画图功能，用它可以编写各种数据处理程序及高效率的管理程序等。

(2) 用户界面好，使用方便。它有当代流行的菜单式图形用户界面，有会话式的编程方式和使用方便的联机帮助，易编程，易调试。

(3) QBasic 采用先进的解释程序，使程序运行比一般 Basic 快很多。当输入一行 QBasic 语句时，系统会立即将该行上所有的关键字改为大写，并进行语法检查，如有错，会立即给出出错信息，因此，特别适合初学者。

(4) QBasic 系统涉及的知识广度和深度适中，有利于普及和提高。学会了 QBasic，再去学习 Visual Basic 是水到渠成的事。否则，直接学习 Visual Basic 是比较困难的。

还应指出，在 Windows 环境下运行的 Visual Basic 与 QBasic 基本兼容，也就是说，用 QBasic 所写的程序一般能在 Visual Basic 环境下顺利运行。因此，掌握了 Qbasic，再过渡到学习 Visual Basic 是很方便的。本书将在学习 QBasic 后介绍 Visual Basic 的有关知识。

1.2 QBasic 程序的基本结构

1.2.1 一个简单的 QBasic 程序

首先用一个简单的 QBasic 程序实例来说明 QBasic 程序的基本结构。