全国计算机等级考试应试培训与指导教程

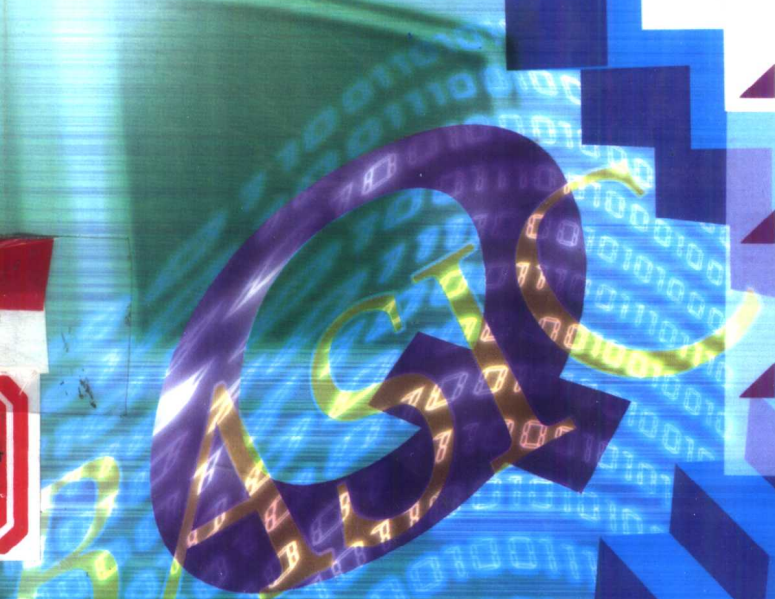
新大纲

二级适用

Quick BASIC 程序设计 应试培训教程

应试培训的首选
轻松过关的导师

- 努力贯彻新的考试大纲，按照新的考试大纲来组织内容
- 兼顾课堂教学和考生考前系统自学和复习的需要
- 这是一个集教师教学、学生自学、考前系统复习为一体的新思维教材
- 本教材特别适合以自学为主的初学者的学习需求



北京工业大学出版社

全国计算机等级考试应试培训与指导教程

二 级 适 用

Quick BASIC 程序设计应试培训教程

本书编写组

北京工业大学出版社

内 容 提 要

本书是《全国计算机等级考试应试培训与指导教程》中的一册，介绍 Quick BASIC 的基础知识和程序设计技术。本书适合于参加各类计算机等级考试的读者自学使用，亦可作为计算机基础教育的入门教材。全书共分 11 章。第 1 章讲解程序设计语言的基础知识、程序设计中常用到的流程图、BASIC 语言的发展历史以及 Quick BASIC 的基本概念；第 2 章讲解 Quick BASIC 的数据与表达式和赋值语句；第 3～6 章讲解了 Quick BASIC 语言中最基础的编程方法，包括输入输出控制、分支结构程序设计、循环结构程序设计、子程序和函数过程；第 7～10 章讲解了 Quick BASIC 的数组、字符串、文件、图形等内容；第 11 章讲解了 Quick BASIC 的集成环境和 Quick BASIC 程序的调试。书末共 5 个附录，分别收录了 ASCII 代码、Quick BASIC 关键字、Quick BASIC 基本语句、Quick BASIC 键盘命令以及 Quick BASIC 内部函数。

图书在版编目（CIP）数据

Quick BASIC 程序设计应试培训教程/《Quick BASIC 程序设计应试培训教程》编写组编. —北京：北京工业大学出版社，1998. 11

全国计算机等级考试应试培训与指导教程

ISBN 7-5639-0744-0

Ⅰ. Q … Ⅱ. Qu … Ⅲ. BASIC 语言-程序设计-教材 Ⅳ. TP312

中国版本图书馆 CIP 数据核字（98）第 33737 号

书　　名	**Quick BASIC 程序设计应试培训教程**
编 著 者	本书编写组
责任编辑	廖晨钟
出 版 者	北京工业大学出版社出版（北京市朝阳区平乐园 100 号　100022）
发 行 者	北京工业大学出版社发行部
印　　刷	徐水宏远印刷厂
开　　本	787×1092 毫米　1/16　21 印张　490 千字
书　　号	ISBN 7-5639-0744-0/T・101
版　　次	1998 年 11 月第 1 版　1998 年 11 月第 1 次印刷
印　　数	0001～5000
定　　价	**31.50 元**

编写说明

计算机等级考试始于1994年。原国家教育委员会于1994年1月颁布了《全国计算机等级考试考试大纲》，《大纲》的颁布带来了全社会学习计算机，使用计算机的高潮。时至今日，计算机应用技能已经成为下个世纪人们的基本技能之一。尤其对于当代大学生，计算机知识已经成为他们知识结构的重要组成部分。目前普遍开展的计算机等级考试正有效地推动这一目标的实现，与此有关的教材和参考资料的需求与日俱增。

然而，计算机科学在不断地发展，计算机等级考试也以更加猛烈之势席卷全国，各系统、行业、组织的等级考试已有10余种之多。如今，奔腾II计算机已成为主流，多媒体、Windows操作系统、计算机网络等原来的考试大纲中没有的内容已成为学习计算机的基本要求，汇编语言的应用更加普遍，数据库技术的发展也导致了FoxBASE与FoxPro成为学习计算机的基本要求之一。计算机硬件和计算机软件的理论也成为计算机应用人员的必学科目。许多教育工作者都认为：跟上时代前进的步伐，保持计算机等级考试的权威性、全面性、时效性，是刻不容缓的。

我们很高兴地看到，教育部已于1998年9月颁布了新的计算机等级考试大纲，为此，本书的主编约请国内部分高等学校从事计算机等级考试教学第一线工作的教师和一些对计算机普及教育有经验的同仁，根据新的大纲编写了《全国计算机等级考试应试培训与指导教程》。本丛书有让人耳目一新的感觉，它浅显易懂、循序渐进、深入浅出。在编写的过程中我们主要注意了以下几点：

一、新大纲与新思维相结合

在努力贯彻新的考试大纲、按照新的考试大纲来组织内容的同时，编者兼顾课堂教学和考生考前系统自学或复习的需要，在讲解基本知识的同时，注意分析难点，着力解决易混淆的概念，纠正错误的观点——这是一本集教师教学、学生自学、考前系统复习为一体的新思维教材。

二、应试与实际应用相结合

考试是为了考核对知识的掌握程度与灵活应用所学知识的能力，本丛书的教材均分为应试培训教材和考试辅导书两部分，配套使用。应试培训教材特别适合初学者，它是自学为主的读者的良师益友。全书在培养读者实际上机操作能力方面的指导意义较为突出。考试辅导书则直接针对考试的知识点、题型进行练习与指导，可在临考前阅读，用来检验复习的效果。

三、本书简介

本书是《全国计算机等级考试应试培训与指导教程》中的一册，介绍Quick BASIC的基础知识和程序设计技术。本书适合于参加各类计算机等级考试的读者自学使用，亦可作为计算机基础教育的入门教材。全书共分11章。第1章讲解程序设计语言的基础知识、程序设计中常用到的流程图、BASIC语言的发展历史以及Quick BASIC的基本概念；第2章讲解Quick BASIC的数据与表达式和赋值语句；第3～6章讲解了Quick BASIC语言中最基础的编程方法，包括输入输出控制、分支结构程序设计、循环结构程序设计、子程序和函数过程；第7～10章讲解了Quick BASIC的数组、字符串、文件、图形等内容；第11章讲解了Quick BASIC的集成环境和Quick BASIC程序的调试。书末共5个附录，分别收录了ASCII代码、Quick BASIC关键字、Quick BASIC基本语句、Quick BASIC键盘命令以及Quick BASIC内部函数。

虽然编委会作了大量细致的工作，但肯定还有不少谬误之处，欢迎广大读者多提意见，以利再版更正。

编者

1998年10月

目　录

第 4 章 分支结构程序设计

第 5 章 循环结构程序设计

第 6 章 过程程序设计

第7章 数 组

第8章 文件系统

第9章 字符串处理

第 10 章 图形程序设计

第 11 章 Quick BASIC 系统简介

附 录

第 1 章　程序设计与 Quick BASIC

在本章中，即将接触到程序设计的基本概念。那么什么是程序？什么是计算机语言？这些最基本名词的含义是读者阅读本章时必须首先解决的问题。如何进行程序设计？程序与算法之间有什么关系？弄懂这两个问题有助于读者从宏观上掌握程序设计的一般思路。另外，我们还介绍了在程序设计和算法表达时都要用到的“流程图”。本章的后一部分主要介绍 Quick BASIC 的主要特点、Quick BASIC 中的若干基本概念和 Quick BASIC 的启动与退出。

本章涉及概念较多，有的又比较抽象，读者应该仔细揣摸，彻底弄懂，因为它们是学习全书的基础。

1.1　程序设计概述

1.1.1　程序的概念与计算机的工作原理

电子计算机是一个能高速运算，具有存储与记忆能力的用程序控制的电子装置。计算机工作时，只有当操作者向计算机输入一定的信息（这种信息必须是计算机能接受的），它才能按照操作者的要求进行工作并且得到所需的结果。目前的计算机的运行机制基本是这样的：

（1）将需要计算机完成的任务编成一条一条的指令，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”）。

计算机通过一条一条的指令来完成工作。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。程序是人们意志的体现，它表明了要计算机“做什么”和“怎么做”。同时，程序又是计算机处理问题的灵魂，只有当操作者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工作。失去了程序的控制，计算机便无法发挥其作用。

1.1.2　计算机语言

程序是计算机指令的集合。我们做一件事时，总要按照一定的步骤一步一步地进行。计算机也是一样的，执行一个程序时总是一条指令一条指令地执行。例如，要求两个数 A、B 之和 S，就可以按照以下的“指令”来进行：

（1）输入 A；

（2）输入 B；

（3）求和 S=A+B；

(4) 输出 S 的值。

那么,怎样把以上(1)～(4)所确定的步骤以一种计算机能够接受的形式输入计算机呢?这就必须克服以下问题:

首先,计算机并不认识汉字或者其他的自然语言,如英语、法语等。若强行把以上(1)～(4)以某种自然语言形式输入计算机,计算机无法"执行"这样的"程序"。

其次,自然语言的意义往往和具体的语言环境有关。一个句子,一个词语在不同的场合的含义往往不同,具有"二义性",而计算机不能容忍这种"二义性"。因为它根本不具备智能,只能机械地、古板地按照你的规定步骤去做,如果你的程序中存在着"二义性",计算机将不知所措。

正是因为以上的原因,我们期待着一种能够准确无误地表达我们编制的程序含义的、一种能够被计算机和人们所接受的、一种相当严谨,不具有二义性的表达方法的出现。这种表达方法正是我们所说的"计算机语言"。

"计算机语言"是人与计算机之间进行通信的工具,是一种计算机能够接受的信息,它由一些简单的单词符号、数字和严谨的语法组成,能够准确无误地表达要完成的具体操作。它是专门用于人们与计算机之间信息交流的一种特殊语言,在人与计算机之间建立起了一座信息的桥梁。目前计算机语言的种类很多,总的来说,它可以分成机器语言、汇编语言和高级语言三大类。

1.1.3 程序设计的基本步骤与任务

人们通过程序让计算机工作起来,以便处理各种信息,解决各种问题,让计算机为人类服务,所以"程序"是人们意志的体现。反过来说,要让计算机按自己的"意志"进行工作,就必须编制程序。编制程序是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。编制程序是本书的核心内容,下面简要地说明编制程序的过程。

(1) 分析问题　这是编程序的第一步,因为任何程序都是为了解决一定的实际问题。编程序时不能无的放矢,而要认真考察实际问题,找出解决问题的大致思路。

(2) 提出算法　把第(1)步中的分析问题的思路进一步明确化、详细化,建立解题需要的数学或物理模型。这也就是把解题步骤一步一步详细地写出来,为下一步用计算机语言来表达这些方法奠定基础。

(3) 编写程序　根据第二步的方案用一定的计算机语言把程序写出来。

(4) 上机调试　对编好的程序进行实际检验,发现其中的错误之处,不断加以改正,直到程序能达到预期目的。

(5) 运行程序将程序投入运行,并输出结果。

程序设计的基本步骤如图 1-1 所示。

1.2 算法与流程图

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

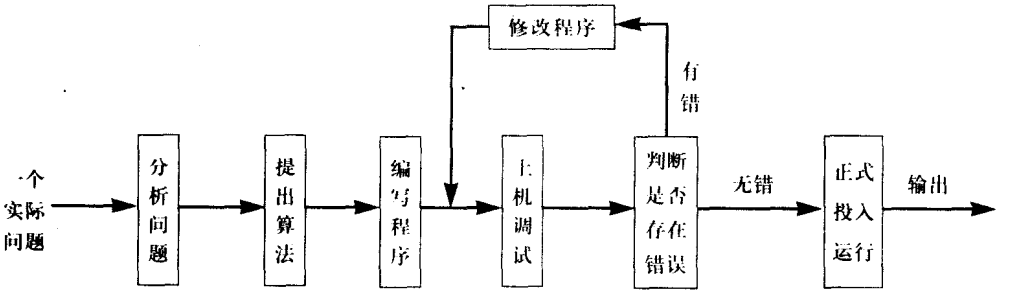


图 1-1

1.2.1　算法

算法就是对问题求解方法的精确描述。在进行程序设计时，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。如果没有认真地研究实际的问题，就草率地提出一些不成熟的算法，那么编写出来的程序就可能出现错误或疏忽。算法作为对解题步骤的精确描述，应具备如下性质：

1. 有穷性

一个算法必须在有限步骤之后结束，而不能无限制地进行下去。因此，在算法中必须给出一个结束的条件。

我们不能指望计算机算出圆周率的精确值，因为 π 是一个无穷不循环小数，无法求出它的精确值。同理，我们也不要让计算机计算诸如 $\lim\limits_{n\to\infty}\dfrac{n^2+1}{n^2-1}$ 的精确值，因为计算机根本无法使“$n\to\infty$”这个条件成立。

2. 明确性

一个算法中的任何步骤都必须意义明确，不能模棱两可、含混不清，即不允许有二义性，不能在计算机中使用诸如“老张对老李说：他的儿子考上了清华大学”这种有歧义的表达方法，到底是老张的儿子上了清华大学还是老李的儿子上了清华大学？无法确知。

3. 可执行性

所采用的算法必须能够在计算机上执行，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。要计算机执行的步骤，计算机应该能够实现，不能提出像“让计算机去煮饭，煮完饭之后再炒菜”之类的算法，至少它在目前无法实现。

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的数据；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。

以上是算法的基本性质，在此基础上就可以学习描述算法的基本工具——流程图。

1.2.2　流程图

流程图是一种能够比较形象地描述“算法”的工具，它对于编制程序很有帮助。流程图又称为框图，是由几种不同的图形组合而成的，如图 1-2 所示。

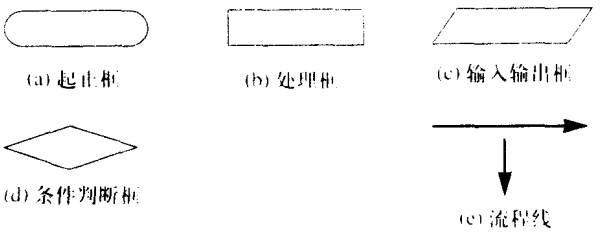


图 1-2

(1) 起止框，如图 1-2(a)，它代表一个算法的开始与结束之处。

(2) 处理框，如图 1-2(b)，它可以表示算法中的一个或若干个步骤，这些步骤不涉及输入与输出。

(3) 输入输出框，如图 1-2(c)，它表示一个算法中需要进行输入或输出处理的步骤。为了与一般的处理步骤区别开来，输入输出框采用了平行四边形的形式。

(4) 条件判断框，如图 1-2(d)，当一个算法中需要依据某一条件来决定后续操作时，采用此框。

(5) 流程线，如图 1-2(e)，它表明每一步骤之间的先后顺序，标识着一个算法的走向。

【例 1.1】用流程图来描述如下算法：向计算机输入一个数 X，若 X≥0 则显示 X 的值。

解：流程图如图 1-3 所示。开始时先遇到起始框，表示算法的开始，之后随着箭头指向输入框，要求你从键盘上向计算机输入一个数 X。再向下，遇到判断框，判断“X≥0”这个条件是否成立，若“X≥0”成立，则打印出 X 的值。否则，沿着标有“不成立”的那条流程线到达终止框，该算法结束。

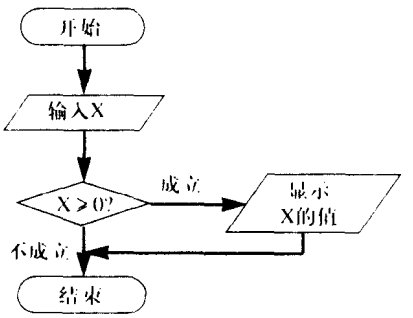


图 1-3

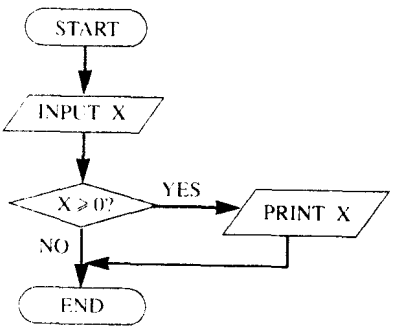


图 1-4

有时候，往往用英文来标注流程图，因为这样与计算机的语言更加接近。于是，图 1-3 可以改画为图 1-4。

1.3 语言的识别与程序的执行

前面叙述仅说明了计算机的程序要通过一定的计算机语言来表达，而未说明计算机是

如何识别用某种语言编写的程序，如何执行程序。这是本节将要讨论的内容。

1.3.1　计算机最终能执行的是机器语言程序

计算机不能识别与执行人类自然语言。计算机内部存储数据和指令是采用二进制("0"和"1")方式的。计算机只能接受和识别由"0"和"1"组成的二进制信息。每一类型的计算机都分别规定了由若干个二进位的信息(即若干个"0"或"1"组成的信息)组成一条指令。例如，某种计算机规定以 1011011000000000 这样的指令作为"加法"指令，遇到这样的指令计算机就执行一次加法。

这种计算机能直接识别和执行的二进制形式的指令称为"机器指令"。例如前面介绍的 1011011000000000 就是一条机器指令。一条机器指令产生一个相应的机器操作。每一种计算机都确定有若干种指令(例如加法指令、减法指令、传送指令、取数指令、存数指令、输出指令……)以实现不同的操作。一种计算机的指令的集合称为该计算机的机器语言(machine language)，或者说该计算机的指令。正如同用算盘算题一样，每一条珠算口诀就是一条"指令"，算盘全部口诀之和就是"珠算语言"。也就是说，"语言"是全部指令的总和。人们为了解决某一问题，可以从该"语言"中选择所需的指令，组成一个指令序列，这个指令序列称为"机器语言程序"。

由于计算机系统的电路逻辑设计上的不同，即使是执行同一种操作(例如在两类不同的计算机上都执行一次加法操作)，不同的计算机系统中的指令是不同的。因此说，机器语言是依赖于具体计算机的(而不是各类计算机都通用的)，它是"面向机器"的低级语言。

用机器语言编写出的程序，计算机能直接识别和执行，执行效率比较高。但是，难学、难记、难写、难检查、难调试、难以推广也是它显而易见的缺点。

1.3.2　BASIC 语言是一种高级语言

为了弥补机器语言的上述缺点，人们创造了一种各类计算机都通用的、接近于人类"自然语言"和"数学语言"的程序设计语言。譬如写出以下一条 Quick BASIC 指令：

```
PRINT SIN(x+y)+COS(x-y)+3.14
```

其中，"PRINT"是一个英文单词，意思是"打印"。"SIN(x+y)+COS(x-y)+3.14"是一个数学式子，它的数学含义是"分别计算(x+y)的正弦值和(x-y)的余弦值，把它们相加之后再加 3.14"。以上是一条接近自然语言(英文)和数学语言的指令。如果计算机能识别这样的指令，将为使用者提供极大的方便。

这种人工创造的语言称为"高级语言(High-level Language)"，相对而言机器语言称为"低级语言(Low-level Language)"。所谓"低级"，指它直接贴近机器。"高级"指离机器远一些，不是直接面向机器的。高级语言在各种计算机都通用。

1.3.3　翻译程序

计算机不能直接识别高级语言，需要有一个"翻译"，把用高级语言编写的程序翻译成用二进制形式表示的机器语言程序(即由若干条机器指令组成的指令序列)，如图 1-5 所示。这个"翻译"工作由一个计算机软件实现。人们在"创造"一种语言的同时，必须设计出一个翻译系统。高级语言程序在运行之前，通过这个翻译系统逐条被翻译成机器语言程序，然后再由

计算机执行。

高级语言程序称为“源程序”，翻译后得到的机器语言程序称为“目标程序”(或“目的程序”)。

目前，国内外使用的高级语言种类很多，不下几百种。最常用的也有十几种，它们运用的范围各不相同。每一种高级语言都有自己相应的翻译程序，翻译程序属于系统软件范畴。

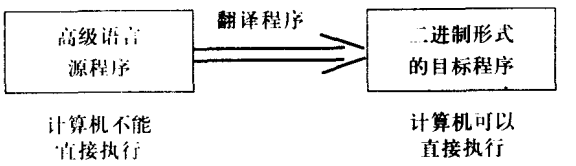


图 1-5

以 BASIC 语言为例，图 1-6 便是一个用户所能看到的 BASIC 程序的执行过程：先将 BASIC 程序输入计算机，发命令让计算机运行此程序。在运行时若需要向计算机提供一些数据，则从输入设备上将数据输入计算机。在 BASIC 语言系统软件的控制下，计算机按照程序一步一步地执行，直至程序运行结束。

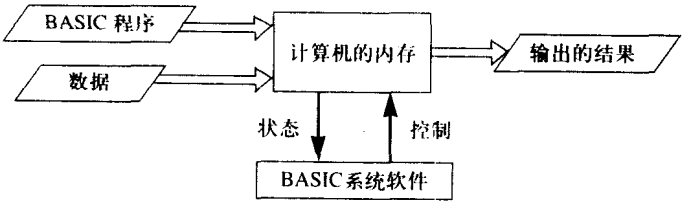


图 1-6

1.3.4 翻译程序的分类

目前的翻译程序有解释与编译两种方式。

1. 编译(Compile)方式

用一个称为“编译程序”的软件进行编译工作。编译过程包括翻译和查错两个功能。它包括：词法分析、语法和语义分析、生成目标程序、优化目标程序。如果发现语法有错，就报告出错信息，不生成目标程序。用户必须修改程序，再次进行编译。如还有错，还要修改和编译，直到不出现语法错误为止，此时生成目标程序(见图 1-7)。

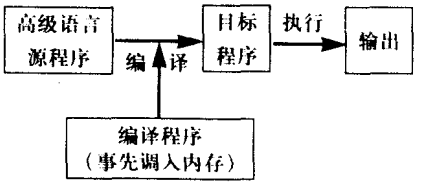


图 1-7

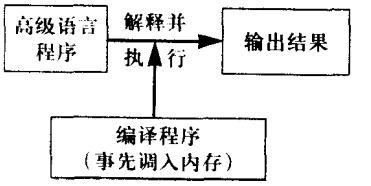


图 1-8

2. 解释(Interpret)方式

解释也是将高级语言源程序翻译为机器指令的一种方式，但它与编译方式不同，不是把

整个高级语言源程序一起翻译成一个目标程序。而是翻译一句，执行一句，不产生整个的目标程序。如果程序没有错误，则一直进行到全部执行完毕。如果在运行过程中发现程序有错，则马上中止“解释”工作，需要修改程序后再重新运行（见图 1-8）。

3. 解释与编译的区别

编译方式相当于笔译，得到一篇完整的译文。解释方式相当于口译，说一句译一句，发现说得不对就停止翻译并且前功尽弃。解释方式使用方便，便于逐条语句调试，但占机器时间多，效率较低。编译方式得到的目标程序经过优化，执行效率高，但占内存多，使用不大方便。如果一个程序要反复多次运行，则用解释方式很不经济，因为每次都要重新进行翻译工作。而用编译方式能得到一个可以保存的目标程序，且编译完后再次执行程序时，编译系统可以撤出内存，只需直接运行该目标程序即可，不必再重新编译，也就是“一次编译，多次运行”（当然，如果源程序有修改，就需要重新编译以便得到一个新的目标程序）。

编译或解释工作都是在操作系统的管理下进行的。图 1-9 表示计算机、操作系统、高级语言翻译程序、用户源程序之间的关系。

用户要计算机运行自己编写的程序，先要编写好源程序，再经过编译（或解释），得到机器指令，在操作系统的统一调度与安排下由计算机执行。从表面上，好像用户在直接操纵计算机，其实，用户与计算机隔了两层，用户是通过操作系统与计算机打交道的，也就是说，在操作系统的统一安排下执行机器指令。

现在，绝大多数语言的翻译工作都采用编译方式。早期的 BASIC 采用解释方式（例如 MS BASIC），后来的 BASIC 采用的是编译方式（例如 Ture BAISC 与 Turbo BASIC 等），有的 BASIC 同时提供解释方式和编译方式（如 Quick BASIC）。

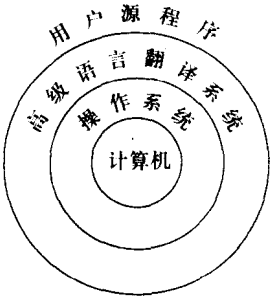


图 1-9

1.4　BASIC 语言的发展历程

1964 年，美国两位计算机科学家 John G. Kemeny 和 Thomas E. Kurtz 在 Dartmouth 学院创造的 BASIC（Beginner's All-Purpose Symbolic Instruction Code——初学者通用符号指令代码）语言问世了。它的诞生是计算机语言发展史上的一件大事。BASIC 语言虽然不是世界上第一个出现的计算机高级语言（第一个出现的高级语言是 1954 年创造的、1956 年正式使用的 FORTRAN 语言），但是 BASIC 语言却为计算机的推广使用立下了汗马功劳。

1.4.1　BASIC 语言产生的背景

FORTRAN 语言是为工程计算设计的，不易掌握，所以 John G. Kemeny 和 Thomas E. Kurtz 决心发明一种新的计算机语言以方便初学者学习，这种语言是一种通用语言，采用人机对话的方式进行工作，用户无需对硬件深入了解。BASIC 语言很好地实现了上述思想，广受欢迎，直至风靡全世界。目前程序设计的语言已有数百种之多，但是“大浪淘沙，唯剩赤

金”，BASIC 语言和其他为数不多的几种高级语言一起，经久不衰，顽强地生存下来，成为目前几种流行的程序设计语言和重要的教学语言之一。

BASIC 语言的发展经历了以下 4 个阶段。

1. 初期的 BASIC

初期的 BASIC 语言功能弱，语句少，只有 14 条语句，后来发展到 17 条语句，这就是所谓的“基本 BASIC”。这个时期的 BASIC 主要在小型机上使用，以编译方式执行。

2. 微机的 BASIC

70 年代，BASIC 语言发展成为一种广泛使用的通用语言，也正是这个年代，微型计算机诞生了，做为必备的软件，各种微机上都配备了 BASIC 语言，例如：Apple BASIC、IBM BASIC(BASICA)等。由于各机型不同，它们对基本 BASIC 语言的扩展也不相同，导致了同是 BASIC 语言程序却不能互相兼容的局面(例如各种 BASIC 语言的绘图语句差别很大)，即所谓“方言性 ”问题。这个时期的 BASIC 开始采用解释方式执行，它方便了用户调试程序。

3. 结构化的 BASIC

结构化程序设计的思想从 70 年代开始萌芽，其主要思想是尽量使程序依书写顺序执行，减少语言之间的跳转，采用模块化设计，各模块完成一定的相对简单的功能，且每个模块只有一个入口，一个出口。结构化程序设计能增加程序的可读性。

在 80 年代中期，美国国家标准化协会(ANSI)根据结构化程序设计的思想，提出了一个新的 BASIC 标准草案。

在此前后，出现了一些结构化的 BASIC 语言，主要有 True BASIC、Quick BASIC、Turbo BASIC 等。True BASIC 对 BASIC 语言作了重大改进和发展，它严格遵循 ANSI BASIC，不仅完全适应结构化和模块化程序设计的要求，而且保留了 BASIC 语言的优点——易学易懂，程序易编易调试，它同时提供了解释工作方式和编译工作方式。它的两位创始人宣称，True BASIC 的出现将开始 BASIC 的新纪元。

Quick BASIC 是 Microsoft 公司 1987 年推出的。Quick BASIC 提供了一个开发程序的集成环境，用户在编程序、修改、编译、调试、运行时均可通过菜单进行操作，十分方便；它与 GWBASIC 和 BASICA 高度兼容，并且提供了全局变量和局部变量；程序模块化；编译后能产生可执行文件，提高执行效率。

4. 在 Windows 环境下运行的 BASIC

80 年代中期，Microsoft 公司推出的 Windows 操作系统又一次给广大计算机用户带来福音。它提供了图形方式的用户界面，通过鼠标、窗口、菜单等操作计算机，使操作变得直观、简单，令人神往。人们不再需要去记忆那些枯燥、复杂的 DOS 命令了。Windows 迅速赢得广大用户，使人们使用计算机变得更加容易了。

最早的基于 Windows 系统的 BASIC 语言是 Visual BASIC(意为“可视的 BASIC”，即图形界面的 BASIC)，它是 Microsoft 公司在 1991 年开发推出的。Visual BASIC 用于开发 Windows 应用软件，用它可以设计出具有良好用户界面的应用程序。Visual BASIC 与 Quick BASIC 兼容，用 Quick BASIC 编写的程序可以不加修改地运行于 Visual BASIC 环境下。Visual BASIC 是一个强有力的软件开发工具 。它一出现就受到普遍重视，是很有前