

PAD及其在高级语言程序设计中的应用

劉 甲 耀

(电子工程系)

摘 要

当前,软件的发展已进入了软件工程时代,产生以图形为工具表现软件的设计方法,PAD是其中富有生命力的一种软件设计表现法.应用PAD进行结构化程序设计可以大大提高程序的设计、制造、检查、维护等生产效率.本文主要论述计算机软件工程的一种重要方法——PAD及其在高级语言程序设计中的应用.阐述了PAD的基本原理、图式及其写法;PAD描述数据结构的方法、标准图式和应用PAD对诸如BASIC、PASCAL、FORTRAN、COBOL、PL/I以及dBASE II高级语言进行程序设计的方法.

一、 引 言

目前,软件的发展已经进入了软件工程时代.所谓“软件工程”即生产软件的工程,是用工程的观点和方法来设计、制造、检查、维护软件.软件工程与其它工程的主要区别在于它的产品为软件,是一种逻辑型、知识型的产品,具有不可视性和各阶段连贯性的特点.因此,如何表现软件产品及其复杂的设计过程,如何使软件开发、维护各阶段的连贯性与正确性,如何使软件过程(生产)由不可视性变为可视性,如何沟通软件各阶段的生产者、管理者之间的联系和他们能顺利进行信息交换.为了解决上述问题而产生了软件工程方法——软件设计表现法(软件设计图法),是以图形为工具来表现软件设计的一种表示方法.软件设计阶段不同,所用的图法也不同.常用的设计图法(或图形工具)有HIPO(Hierarchy, plus input, Process, Output), SC(Structure Chart), FC(Flow Chart)以及树图(SPD, HCP, PAD)多种.其中, PAD是使用二维树形结构图描述过程的逻辑,使得程序易读、易记、易理解,因而大大提高程序的设计、制造、检查、维护等生产效率,是软件工程的后起之秀,具有强大生命力.为此,本文主要阐述PAD及其在高级语言程序设计中的具体应用.

二、 PAD的基本原理、图式与写法

1. PAD的基本原理

PAD(Problem Analysis Diagram)是在改善Warnier图的基础上产生的一种二维的树形结

本文1986年5月5日收到.

构的软件设计表现法。由于它使用PASCAL的控制结构，可以看作是PASCAL程序的二维展开图，因而也把PAD看作是PASCAL Diagram的缩写，称之为PASCAL的展开图。

PAD的核心思想是利用自顶向下设计和反复的逐步求精的基本概念，把人们头脑中解决问题的粗略、模糊的想法，转换成计算机能进行处理的明确而详尽的过程。

PAD使用六种符号(图1)，用于描述处理、重复、选择、语句标号、定义和过程。

与其它规定的软件设计方法类似，PAD为软件设计者提供了设计系统或程序时所遵循的步骤：

(1) 从设计程序或系统的一种模糊的概念开始，标识这种自由地定义所存在的过程的各个部分的顺序，即在时间顺序上模模糊糊部分的前后关系——顺序(连接)的细化。

(2) 标识将要重复的部分以及重复的初始、终止条件，即主要循环过程的开始和结束条件等——循环的细化(重复的细化)。

(3) 标识将要执行的各个过程的条件，即主要选择的条件——选择的细化。

(4) 如果需要平行处理、则标识将要并行执行的那部分——平行性的细化。

(5) 重复(1)~(4)步，直到过程完全确定和完全精确为止。

上述过程如图2所示。

重复执行，直到完全精确为止

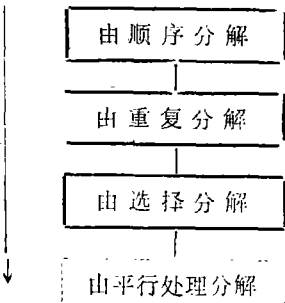


图2 PAD方法的处理流程

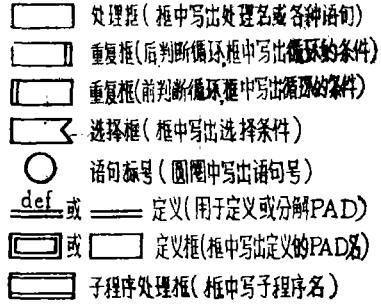


图1 PAD中使用的符号

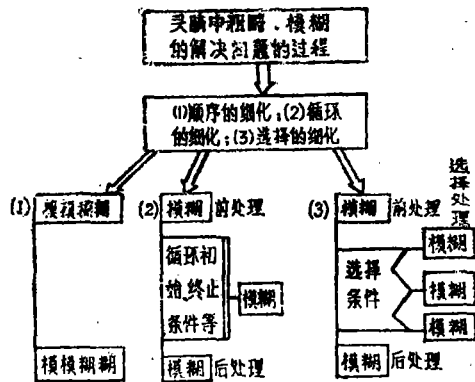


图3 PAD程序开发原理

。归结起来，PAD程序开发的原理可用图3描述。

2. 程序结构和PAD的基本图式

任何程序都是由顺序、循环(重复)、选择三种基本形式组成。

(1) 顺序结构——处理两个事情以上的时间序列。

(2) 循环结构——凡条件成立就反复执行。

(3) 选择结构——处理两个以上的事情时，选择满足条件的处理。

这三种形式的顺序结构及其对应的流程图(Flow Chart, 简写FC)与PAD图如表1所示，而在循环型中又分前判断型与后判断型以及面对问题型三种，它们的关系如表2所示。

正如前面所指出,除顺序、循环、选择三种基本型式外,还有一种平行结构型式,用于进行平行性处理,如图4所示。

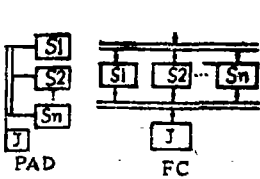
表1 程序结构及PAD的基本形式

表2 前判断循环与后判断循环的关系

型式	顺序型	循环型	选择型	代号	循环图式	与它 其它循 环的关 系	流 程 图	代号
PAD描述								
FC描述								

说明: (1) y表示条件成立, N表示条件不成立。
(2) R—1图,读“Q条件成立时”,称“前判断循环”。
(3) S—1图,表示Q条件成立时执行上方的处理,否则执行下方的处理。

说明: (1) R—2图,读“直到Q条件成立为止”,称后判断循环。
为R—3图,读“从M到N,步长K”,称面对问题循环。



PAD

FC

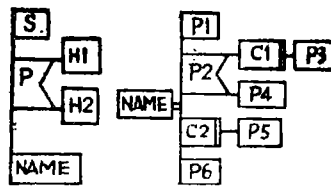


图5 定义上下页连接的用法

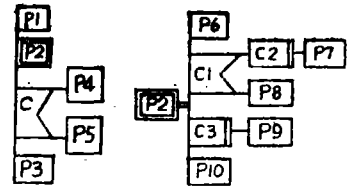


图6 定义PAD细部的用法

图4 平行处理图式(含义: (a)上页的PAD (b)下页的PAD 平行处理S1, S2, ..., Sn, 然后再处理J)

3. PAD的写法及注意事项

(1) PAD的写法及def的使用

PAD是一种二维的树形结构的程序表示法,它表示图形/概念的一种唯一的组合。纵向表示顺序信息,横向描述分支和嵌套层,两者组合即构成表示程序的PAD图。

写PAD时,用户可从定义一个程序或程序组的顶层图开始,然后再用def符号添加详图。

def或 \equiv 符号用于定义PAD,其写法为

$S \equiv PAD$ 或 $S \equiv PAD$ 或 $S \equiv PAD$

其中, S为所定义PAD的名字。

def 用法有三:

- (i) 用于表示改进阶段的各个详细步骤,即逐步求精。
- (ii) 用于把较大的PAD分解成较小的PAD。

(iii) 用于对某一处理添加 PAD，或连接上下页，即在本页末尾和下页开始定义之。

例 1 定义上下页的连接 (图 5)。

例 2 定义 PAD 细部的用法 (图 6)

(2) 注意事项

(i) 当处理框为空时，可省略该框及其分支。例如图 7 所示。

(ii) 在 PAD 图外侧可写上注解，如图 8 所示。

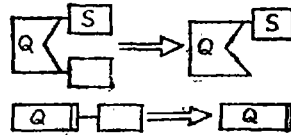


图 7

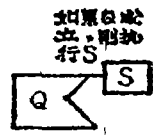
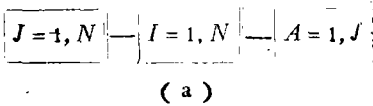


图 8

三、用 PAD 描述数据结构的方法

数据结构也可用象程序结构一样的那三种基本形式来描述，相当直观简单，如表 3 所示。

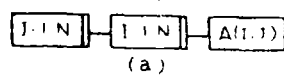
例如， $N \times N$ 的二维矩阵 $A(1,1), A(2,1), \dots, A(N,1), A(1,2), \dots, A(N,N)$ 可用 PAD 描述为图 9 (a) 所示。又如，BASIC 的变量名可用 PAD 描述如图 9 (b) 所示。



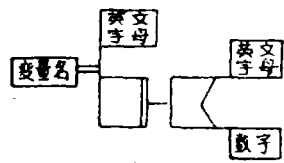
(a)

表 3 数据结构的基本形式

基本形式	PAD	含义
顺序	<div><div>A</div><div>B</div></div>	数据 A 后为数据 B
循环	<div><div>N</div><div>A</div></div>	重复操作数据 A 为 N 次，如果 N 为未知数则可省略，并且循环条件可写成 $I=1, N$
选择	<div><div>A</div><div>B</div></div>	根据条件选择 A 或 B，选择条件可写成 $P < Q$ 或 $I=1, N$



(a)



(b)

图 9

四、PAD 在高级语言程序设计中的应用

PAD 写法基本上与语言无关，除标准图式外，使用者可根据需要进行不同的扩充。

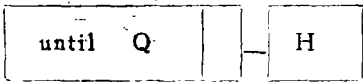
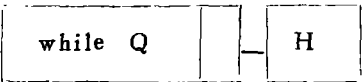
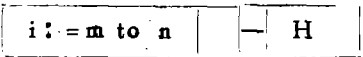
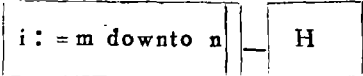
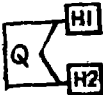
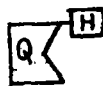

BASIC、PASCAL、FORTRAN、COBOL、PL/I 以及 DBASE II 所用的 PAD 标准图式如表 4—7 所示，表 5 中 PASCAL 的 PAD 图的对应流程图如图 10 所示，而表 7 中 dBASE II 的 PAD 图的对应流程图如图 11 所示。

由于 PAD 是用于表示程序的一种二维树形结构，因而，根据 PAD 能容易转换为源程序。

表 4 BASIC 用的 PAD 标准图式

基 本 型 式		程 序	
顺 序	图 式	正 规 型	简 略 型
前 判 断		H1 H2	如果H1和H2写在一行上, 则写成 H1: H2
后 判 断		L1 REM IF NOT(Q) THEN L2 H	用WHILE语句则写成 WHILE Q H WEND
问 题 对 环		L REM H IF NOT(Q) THEN L	
单 分 支		FOR I = M TO N STEP K H NEXT	如果步长K为1则STEP K可省略
二 分 支		IF NOT(Q) THEN L H REM L1 REM H2 GOTO L2 L2 REM	如果把H写成一行则可写成 IF Q THEN H 如果IF语句含有H1, 把H1和H2写在1行, 则可写成 IF Q THEN H1 ELSE H2
多 分 支		ON I GOTO L1, L2, ..., Ln L1 REM H1 GOTO L Ln REM Hn L REM	

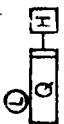




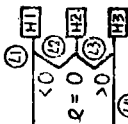
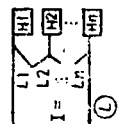
表 5 PASCAL用的PAD标准图式

		PAD	PASCAL
循 环	UNTIL		repeat H until Q;
	WHILE		while Q do H;
	DO		for i := m to n do H;
	DOWNTO		for i := m downto n do H;
选 择	①		if Q THEN H1 ELSE H2;
	②		if Q THEN H;
	③		case i of L1 : H1; : Ln : Hn; end;

注：图中PAD对应的流程图见图10

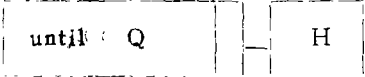
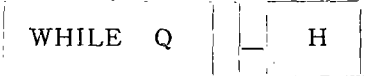


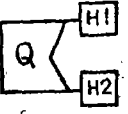

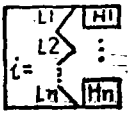
- ① = IF THEN ELSE
- ② = IF THEN
- ③ = CASE

表 6 FORTRAN、PL/I、COBOL用的PAD标准图式

	PAD	FORTRAN	PL/I	COBOL
后判断		L CONTINUE H IF (Q) GOTO L	DO UNTIL Q; H; END;	PERFORM H UNTIL Q. (H是段名, 以下同)
前判断		L CONTINUE H IF (Q) GOTO L1 H	DO WHILE Q; H; END;	PERFORM H UNTIL (NOT Q).
面向问题		DO L I=M,N,K H CONTINUE	DO I=M TO N BY K; H; END;	PERFORM H VARYING I FROM M BY K UNTIL I>N.
二支分型		IF (Q) O GOTO L1 H1 GOTO L2 H	IF (Q) THEN H1; ELSE H2;	IF Q H1. ELSE H2.
单纯型		IF (Q) GOTO L H	IF Q THEN H;	IF Q THEN H.
计算分支型		IF (1) L1, L2, L3 L1 CONTINUE H1 GOTO L4 L2 CONTINUE H2 GOTO L4 H3 L4 CONTINUE H	SELECT (1) WHEN(1) H1; WHEN(2) H2; WHEN(n) Hn END;	GOTO L1, L2, ..., Ln DEPENDENT ON I. L1. H1 GOTO L. Ln. Hn. L.
多分支型		GOTO (L1, L2, ..., Ln), I L1 CONTINUE H1 GOTO L Ln CONTINUE Hn L CONTINUE		

注: Q表示Q的非(否定); O是编程时根据需要写上的实际标号。

表 7 dBASE II 用的PAD标准图式

		PAD	DBASE II
循 环	U N T I L		STORE "Y" TO B DO WHILE Q="Y" H IF Q STORE "N" TO B ENDIF ENDDO
	W H I L E		DO WHILE Q H ENDDO
	D O		STORE M TO I DO WHILE I<=N H STORE I+K TO I ENDDO
	D O W N T O		STORE M TO I DO WHILE I>=N H STORE I-K TO I ENDDO
选 择	IF THEN ELSE		IF Q H1 ELSE H2 ENDIF
	IF THEN		IF Q H ENDIF
	CASE		DO CASE CASE I=L1 H1 : CASE I=L Hn ENDCASE

注，图中PAD对应的流程图见图11.

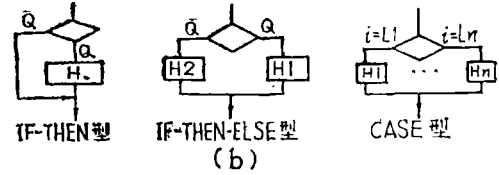
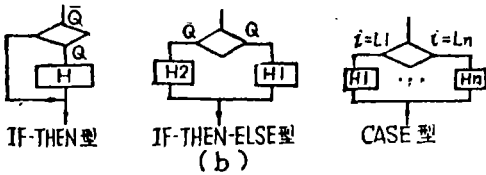
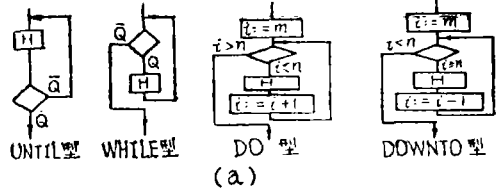
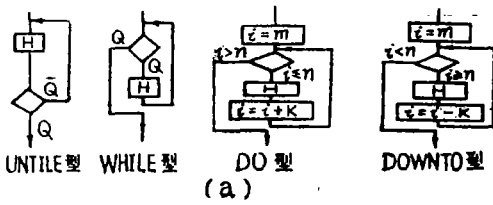


图10 对应PASCAL用的PAD标准图式的流程图
(a)循环型; (b)选择型

图11 对应dBASE II用的PAD标准图式的流程图
(a)循环型; (b)选择型

要借助PAD编程,首先要写出PAD,然后再按该图式进行编程,编程有两种方法:一种是由PAD人工地化为源程序;另一种是在有PAD系统支持的计算机上,直接输入PAD,机器即自动编码(自动写出源程序)。人工地根据PAD编程的步骤如下:

(1) 写出PAD图,并标上语句标号。根据算法画出PAD图,然后在IF、GOTO、FOR、WHILE、ON语句的相应处记入○,再把PAD看作横向生长的树,沿着树迹前进,以10为单位递增地在○标记中写上语句标号(这种沿着树形结构一步一步前进查找的方法称为树迹法或沿树形前进法(Tree Walk)),这样画出的图称为树迹图。

- (2) 编写源程序。沿着树迹前进(箭头所示方向),写出源程序,其原则是:
- (i) 遇到处理部分,写出其处理内容;
 - (ii) 遇到语句标号,如10,则写10 REM(对BASIC),或10CONTINUE(对FORTRAN);
 - (iii) 遇到循环R—1图或R—2图或选择S—1图,则根据表4—7所示的相应高级语言的标准的PAD图式写上FOR、IF、GOTO等语句。

就这样,沿树迹结束程序即告完成。

示例

1. 在IBM-PC机上(配置PAD系统)直接输入PAD图,由机器自动编出BASIC源程序

例 要求对任意排列的n个数按它们的数值由大到小顺序排列。

(1) 根据题目要求写出PAD(注:图中的树迹及语句标号记号○是为了说明树迹法而人为地写上的,实际上可以不写),如图12所示。

(2) 根据PAD写出源程序(注:这里是机器自动编出的程序,实际上也可方便地由PAD转换成源程序):

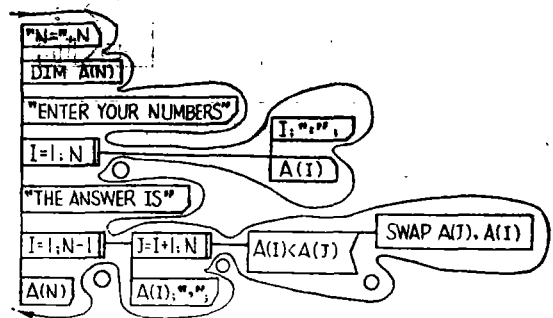


图 12

10 INPUT "N=" ; N	RUN
20 DIM A(N)	N=? 10
30 PRINT "ENTER YOUR NUMBERS"	ENTER YOUR NUMBERS
40 FOR I=1 TO N	1: ? 10.98
50 PRINT I, ":" ;	2: ? 11.0
60 INPUT A(I)	3: ? 22
70 NEXT	4: ? 34
80 PRINT "THE ANSWER IS"	5: ? -57
90 FOR I=1 TO N-1	6: ? -57
100 FOR J=I+1 TO N	7: ? 34.5
110 IF A(I)<A(J) THEN 130	8: ? 19.2
120 GOTO 140	9: ? 22.4
130 SWAP A(J), A(I)	10: ? 10.981
140 NEXT	THE ANSWER IS
150 PRINT A(I); " , " ;	34.5, 34, 22.4, 22, 19.2,
160 NEXT	11, 10.981, 10.98, -57, -57
170 PRINT A(N)	
180 END	

2. 对同一问题(求和数及最大值),由PAD人工地转换成BASIC、FORTRAN、PASCAL、COBOL、PL/I源程序。

(1) BASIC/PAD:

(i) PAD如图13所示。

(ii) PAD⇒源程序:

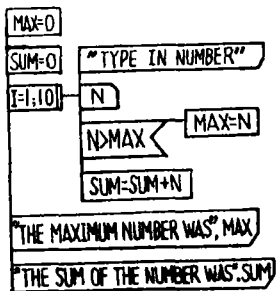


图 13

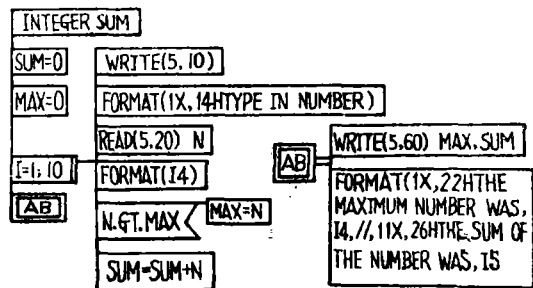


图 14

10 MAX = 0	70 GOTO 90
20 SUM = 0	80 MAX = N
30 FOR I=1 TO 10	90 SUM = SUM + N
40 PRINT "TYPE IN NUMBER"	100 NEXT
50 INPUT N	110 PRINT "THE MAXIMUM NUMBER WAS" ,
60 IF N>MAX THEN 80	MAX

120 PRINT "THE SUM OF THE NUMBER
WAS", SUM

130 END

(2) FORTRAN/PAD;

(i) PAD如图14所示。

(ii) PAD⇒源程序

INTEGER SUM

SUM = 0

MAX = 0

DO 50 I = 1, 10

WRITE(5, 10)

10 FORMAT(1X,

* 14H TYPE IN NUMBER)

READ(5, 20)N

20 FORMAT(14)

IF N.GT.MAX GOTO 30

GOTO 40

30 MAX = N

40 SUM = SUM + N

50 CONTINUE

WRITE(5, 60)MAX, SUM

60 FORMAT(1X, 22H THE

MAXIMUM NUMBER WAS, 14, //,

* 11X, 26H THE

* SUM OF THE NUMBER

* WAS, 15)

STOP

END

(3) PASCAL/PAD;

(i) PAD如图15所示。

(ii) PAD⇒源程序;

PROGRAM LJY;

VAR I, N, MAX, SUM;

INTEGER;

BEGIN

SUM := 0;

MAX := 0;

FOR I := 1 TO 10 DO

BEGIN

WRITELN(' TYPE IN

NUMBER');

READ(N);

IF N > MAX THEN

MAX := N;

SUM := SUM + N;

END;

WRITELN(' THE MAXIMUM

NUMBER WAS, MAX);

WRITELN(' THE SUM OF THE

NUMBER WAS, SUM);

END.

(4) COBOL/PAD

(i) PAD如图17所示。

(ii) PAD⇒源程序;

IDENTIFICATION DIVISION.

PROGRAM LJY.

AUTHOR.LGI.

DATA DIVISION.

WORKING-STORAGE

SECTION.

77 N PIC 9999.

77 SUM PIC 99999,

VALUE ZERO.

77 MAX PIC 9999,

VALUE ZERO.

PROCEDURE DIVISION.

MAIN.

PERFORM LOOP 10 TIMES.

DISPLAY "THE MAXIMUM

NUMBER WAS", MAX.

DISPLAY "THE SUM OF

THE NUMBER

WAS", SUM.

STOP RUN.

LOOP.

DISPLAY "TYPE IN

NUMBER".

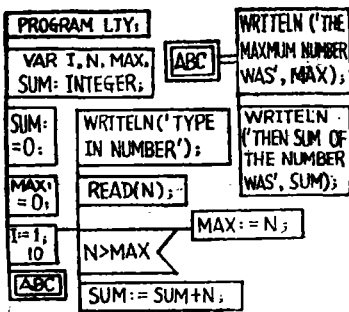


图 15

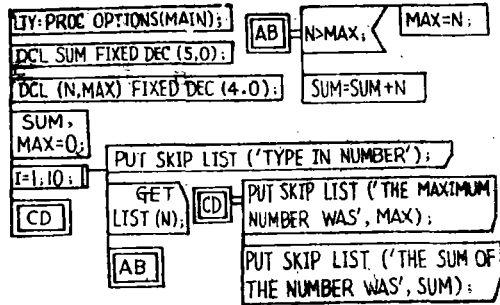


图 16

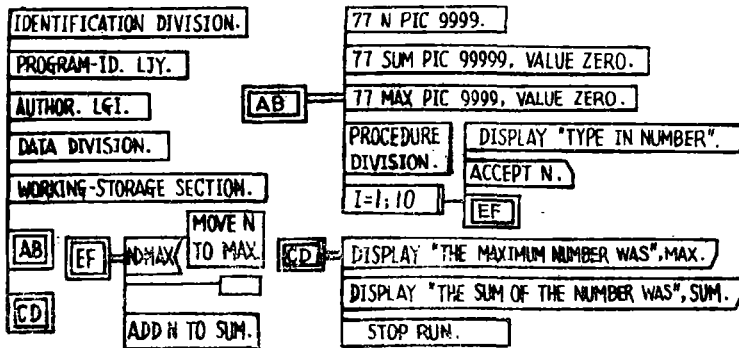


图 17

ACCEPT N.

IF N > MAX MOVE N TO MAX.

ADD N TO M.

(5) PL/I/PAD

(i) PAD如图16所示。

(ii) PAD⇒源程序:

LJY, PROC OPTIONS(MAIN);

DCL SUM FIXED DEC(5,0);

DCL (N,MAX) FIXED DEC(4,0);

SUM, MAX = 0;

DO I = 1 TO 10;

PUT SKIP LIST('TYPE IN NUMBER');

GET LIST(N);

IF N > MAX THEN MAX = N;

SUM = SUM + N;

END;

PUT SKIP LIST('THE MAXIMUM NUMBER WAS', MAX);

PUT SKIP LIST('THE SUM OF THE NUMBER WAS', SUM);

END;

五、结 束 语

1. 用PAD表现程序,使得程序结构清晰,层次分明。
2. 便于和易于由PAD转换为源程序(人工或机器自动编程),且对程序设计风格有较强的制约性。

3. 采用PAD进行编程,能大大提高程序设计效率。

因此,应大力开发把源程序自动转换为图形以及由图形直接转换为源程序的PAD工具,大力推广使用PAD技法。

参 考 文 献

- 〔1〕二村良彦, プログラム技法——PADによる構造化プログラミング——, オーム社, (1985).
- 〔2〕何克清, 计算机软工工程学, 武汉大学出版社, (1983).
- 〔3〕Lawrence J. Peters Foreword, L.A. Belady, Software Design, Methods & Techniques, Yourdon Inc. (1981).
- 〔4〕严桂兰、刘甲耀, 软件工程讲座, 福建电脑, 3 (1986).

PAD and Its Applications in High-level Language Programming

Liu Jiayao

Abstract

Currently, the development of software has stepped in an age of software engineering. It brings forth the software designing method of using graph as tool, of which PAD is a representation full of vitality. By using PAD in structured programming, the efficiency of design, manufacture, testing and maintenance of programs can be improved greatly.

This paper describes PAD and its applications in high-level language programming. It describes the fundamental principles, graph, and written form of PAD, and it introduces the method for describing the data structure, and finally, it shows the standard graph of PAD and its applications in BASIC, FORTRAN, COBOL, PL/I, dBASE II, and other high-level language programming.