

1. 序

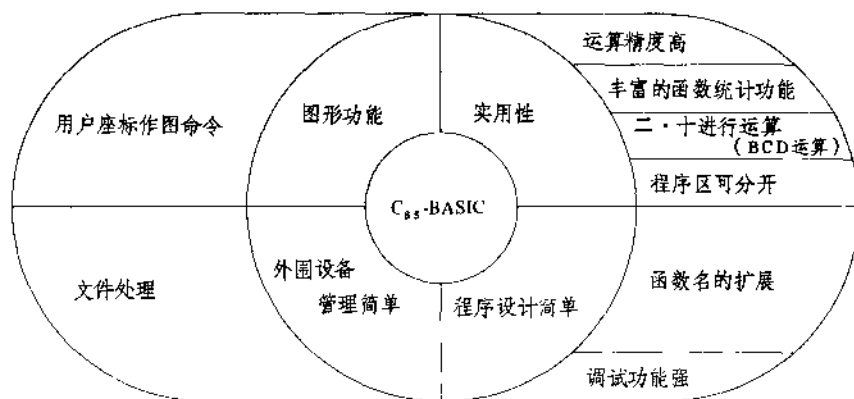
1-1 前 言

本书讲解了FP-200的程序设计语言C_{ss}-BASIC和CETL的文法。

除本书外FP-200还有一本「FP-200使用手册」,建议读者首先阅读「使用手册」掌握基本的使用方法和练习编制CETL、C_{ss}-CASIC简单程序。

1-2 C85-BASIC的特长

C85-BASIC是在最新制定的JIS基本BASIC(C6207)的基础上,增强了运算功能、文件处理功能,并增加了图形处理功能,是一种高性能的BASIC语言。



C85-BASIC

• 与一般的BASIC相同,有以下特长。

(1) 容易理解

与FORTRAN相比,文法的约束较少。因此初学者也能简单地使用。

(2) 编制程序简单

因为可以和计算机采用对话形式编制、修改、执行程序,故简单、快速。

(3) 运算精度高

为确保不同运算的精度,提供了以下2种数值型。

① 单精度实数型 — 内部9位·显示6位。

② 双精度实数型 — 内部19位·显示16位。

另外,各个实数型的指数部分可取-99~99。所以可以处理实际应用的所有实数。

(4) 丰富的函数·统计功能

① 17种标准的数学函数都能双精度地加以使用。

SIN, COS, TAN, ASN, ACN, ATN, LOG, LGT, EXP, SQR, ABS, SGN, INT, FIX, FRAC, ROUND, RND*

有*号的RND是单精度。

② 强力的字符控制函数

CHR\$, STR\$, MID\$, LEFT\$, RIGHT\$, ASC, VAL, LEN

③ 包括直线回归的统计功能。

(5) 二、十进制运算

可以在事务处理、科学技术计算中安心地使用二、十进制(BCD)运算功能。

(6) 程序区可分

在主存储装置内可同时存入10个程序。因此,当处理变换时,不需要每次都输入程序。

(7) 函数名的扩展

函数名最长可达到255字符。同时还可以使用假名及小写字符。用符合函数内容的函数名,可便于理解程序的内容。

(8) 强力的调试功能

为了检查程序执行的情况,可在液晶显示的语句号中加进TRON命令。

(9) 统一的外围设备控制

为了便于控制外围设备(软磁盘装置等),引入了「文件处理」的概念。

这样以来,与所有外围设备的数据交换,都可以通过文件来进行。

结果,无论使用什么设备,都可用同样的方法处理。

(10) 加强了图形处理功能

① 使用了用户座标

画面上的位置可以根据用户能任意决定动用户座标来确定。

这样以来,免去了烦琐的座标计算,可以进行顺利的图形处理。

② 强力的作图指令

具备直线、方形等作图指令。

1-3 键盘

1-3-1 键盘的构成与动作

本计算机键盘是标准 J I S 键盘。另外具有将一系列操作集中成一次键操作的程序功能键、和独立的经常使用的功能键。是一种使用方便的键盘。

1-3-2 方式选择键



<控制键>

按住这个键再去按其它键，便能以“控制方式”进行输入。



<图形键>

按住这个键再去按其它键，便能以“图形方式”输入。



<首字母变换键>

首字母变换方式的设定和解除。每按一次这个键，可以设定或解除。首字母变换方式设定时，光标从“一”变成了“二”。当输入小写字母时要设定。



<转换键>

按住这个键再去按其它键时，以各方式的转换状态输入。但是在控制方式和图形方式时，本键不起作用。



<转换封锁键>

封锁或开放转换方式。每按一次此键，移位方式就变化一次。当移位方式被封锁时，光标显示从“一”变成“二”或从“二”变成“三”。

1-3-3 程序功能键

键盘最上一列键 P F 0、~ P F 9 叫作程序功能键。将一系列键操作编目在功能键上，这样只要按一次功能键便可完成一系列键操作。

最初接通电源时，功能键形式如下。

BASIC 方式

PF 0 : EDIT
PF 1 : PROG
PF 2 : SYSTEM^C_R
PF 3 : LIST^C_R

PF 4 : RUN^C_R
PF 5 : S₀
PF 6 : FILES^C_R

PF 7 : LOAD"
PF 8 : SAVE"
PF 9 : P. DATE\$, TIMES

CETL方式

PF 0: D₃

PF 1: FILE

PF 2: SYSTEM^C_R

PF 3: S_H

PF 4: P: FRE^C_R

PF 5: S₀

PF 6: FL(

PF 7: RC(

PF 8: IT(

PF 9: P. DATE\$, TIME\$

* PF 5 (即 SHIFT + PFO) 可用作输入数字的变量键。也可作为输入 + 和 * 的 ; 和 : 键。

1-3-4 字母数字记号键

按确定的方式, 输入字母、数字、图形记号等。另外, 在控制方式时, 可进行特别规定的动作。但是, 为了使键盘看起来方便, 没有写图形记号及控制方式的动作。

① 正常方式的动作

(1) 不按转换键时

输入从 A 到 Z 的大写字母。输入其它键下方的文字记号。

<例>



A



8

(2) 按转换键时

输入 A ~ Z 键的小写字母。输入其它键上方的文字记号。

<例>



a



(

② 首字母转换方式的动作

(1) 不按转换键 (SHIFT) 时

输入 A ~ Z 键的小写字母。输入其它键下方的文字。



a



8

(2) 按转换键 (SHIFT) 时

输入 A ~ Z 键的大写字母。输入其它键上方的文字。

<例>



A



(

③ 图形方式的动作

不论按与不按转换键，都可输入预先定好的图形记号。



④ 控制方式的动作

可进行预先定好的特定动作。请参考书后的「键码表」。

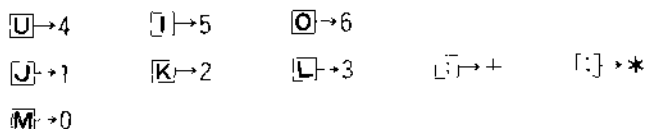
⑤ 转换锁住方式 (SHIFT lock mode) 的动作。

不管按与正按转换键，都将转换方式锁住。

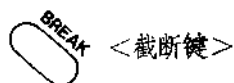


⑥ 数字方式 (NUM mode)

将一部分字母数字键、记号键作为数字键、运算命令键使用，使输入数据更简单。



1 - 3 - 5 其它键



<截断键>

强制中止程序执行。



<停止、继续键>

工作中按下此键，暂时停止执行，再按一下又继续执行。

CONT是CONTINUE的略语。



<回车键>

在向计算机输入指令、程序和数据时，最后要按一下此键。

便将光标所在行（逻辑行）的内容输入到机内。



<光标上移键>

使光标向上一行移动。

在 BASIC 方式中, 按住转换 (SHIFT) 键时按此键就将从本行变成前一行的编辑方式。

在 CETL 方式中, 按住转换键 (SHIFT) 时, 按此键, 就将从现单元变成上单元的编辑方式。



<光标下移键>

使光标向下一行移动。

在 BASIC 方式中, 按住转换键 (SHIFT) 时按此键, 将从现行变到下一行的编辑方式。

在 CETL 方式中, 按住转换键 (SHIFT) 时按此键, 将从现单元变到下一单元的编辑方式。



<光标右移键>

使光标向右移动。

在 BASIC 方式中, 按住转换键 (SHIFT) 时按此键就将从现行变到下一行的编辑方式。

在 CETL 方式中, 按住转换键 (SHIFT) 时按此键就将从现单元变到右一单元的编辑方式。



<插入、清除键>

将在光标位置的文字清除, 并将光标以后到行末尾的文字向在移动一位。按住转换键再按此键时, 使现在的光标位置空一位, 并将此位后面到行末尾的文字全都向右移动一位。



<清除画面、光标返回原点键>

使光标移到左上角。按住转换键再按此键时, 画面全被清掉, 光标移到左上角。

另外, 在编辑方式中, 如果按了光标返回原点键或清除键时, 不能进行行修正。



<光标左移键>

使光标向左移动。

在 BASIC 方式中, 按住转换键 (SHIFT) 时按此键就将从现行变到上一行的编辑方式。

在 CETL 方式中, 按住转换键 (SHIFT) 时按此键, 将从现单元变到左一单元的编辑方式。

1-4 输入方式

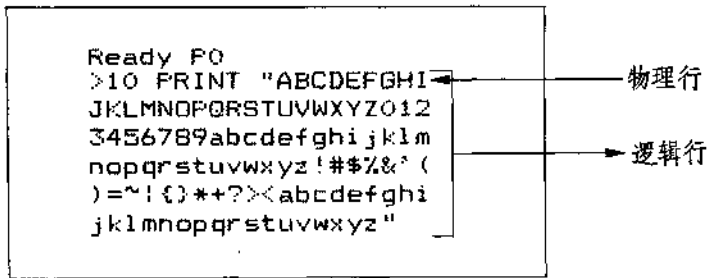
1-4-1 逻辑行与物理行

按一下键,该键的内容便显示在画面上。还可以用光标移动键、INS键、DEL键等,去修正这个内容。

如果按一下RETURN键,就可将数据输入。可是,这里用按RETURN键可以输入的数据的范围并不明确。是整个画面还是有光标的一个字符,没有具体的确定。

我们把输入范围定义为有光标的一行。如果是画面的一行,只能输入20个文字。我们将此画面上的一行叫物理行。

输入时使用的一行有更广的范围,这样的行称为逻辑行。



逻辑行的范围是包含现在光标位置数据继续。此数据内还包含空隔。也就是说,在计算机中即使在什么也没表示的地方,也区别“什么也没有”的状态和“有空隔”的状态。这样,即使按了空隔键,在逻辑行中也将它包括在内。

1-4-2 数据输入

1-4-2-1 可能输入数据的范围

在用键盘输入数据时,既使把光标移到逻辑行前方的位置,也不能在那里将数据输入。另外当逻辑行末尾在画面内时,最下一行的①及右下端的②也不起作用。

1-4-2-2 输入行的换行及上滚

将在端作为 0 位时,当输入 3 第 19 位后便进行改行。另外,当输到最下端第 19 位后画面将向上滚动。

但是当数据输入中,用插入键插入数据,既使一个画面上的数据超过 180 字符,也不上滚只有在右下端的位置上输入了数据,才向上滚一行。

1-4-2-3 数据输入中的光标返回原点键及画面清除键。

数据输入中,如果按了光标返回原点键或画面清除键,将会把到目前为止输入的数据清除、成为无效。

但是,还留在画面上。

1-4-3 直接方式

输入指令语句后再按  键,便直接执行那个指令语句。因此,没有程序也能执行简单的动作。

<例> PRINT "Future Personal computer"
 RETURN

差不多所有的指令都可用直

接方式执行,但也有不能直接执行的指令。

```
>PRINT "Future Personal Computer"
Future Personal Computer
```


2-2 字符数据

可处理 0 ~ 255 字符的可变长字符串。1 个字符以 1 个字节表示，差不多都对应于 ASCII 8 位字码。另外，字符数据可用 CLEAR 语句存储在指定的字符领域中。

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0			(SP)	0	@	P	`	p		┌					二	×
1			!	1	A	Q	a	q		└					卅	円
2			"	2	B	R	b	r		┐					卅	年
3			#	3	C	S	c	s		┐					卅	月
4			\$	4	D	T	d	t							▲	日
5			%	5	E	U	e	u							▲	時
6			&	6	F	V	f	v							▲	分
7			'	7	G	W	g	w							▲	秒
8			(	8	H	X	h	x		┌					♠	〒
9			)	9	I	Y	i	y		└					♥	市
A			*	:	J	Z	j	z		┐					♦	区
B			+	:	K	[	k	l		┐					♣	町
C			,	<	L	\	l								●	村
D			-	=	M	]	m								○	人
E			.	>	N	^	n	~							／	／
F			/	?	O	_	o	?		┐					／	／

3. C₈₅-BASIC的基础

3-1 构成程序的字符

C₈₅-BASIC是由下图各种字符 (JIS号是20~FE) 构成。另外, 粗线框内的字符 (JIS号是3F、5B、5D、5F、60、70B-7E、80~9F、E0-FE) 只用于REM语句、DATA语句、及字符常数表。

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0			(SP) O	@	P	~	p								三	×
1			!	1	A	Q	a	q							上	円
2			"	2	B	R	b	r							丰	年
3			#	3	C	S	c	s							日	月
4			\$	4	D	T	d	t							▲	日
5			%	5	E	U	e	u							▲	時
6			&	6	F	V	f	v							▲	分
7			'	7	G	W	g	w							▲	秒
8			(	8	H	X	h	x							♠	〒
9			)	9	I	Y	i	y							♥	市
A			*	:	J	Z	j	z							♦	区
B			+	:	K	[	k	{							♣	町
C			,	<	L	\	l								●	村
D			-	=	M	]	m								○	人
E			.	>	N	^	n	~							/	人
F			/	?	O		o	?							/	\

3 - 2 符号名

下面的文字称为定义字，具有表示命令，函数等特殊意义。另外，定义字没有大小写的区别，全都变成大写来使用。

<例> PRINT和PRINT相同，都是定义字。

ABS	ACS	ALL	AND	ANGLE
AREA	AS	ASC	ASN	ATN
BASE	CALL	CDBL	CHRS	CLEAR
CLOSE	CLS	CNT	COS	CSNG
CVD	CVS	DATA	DTAE\$	DEF
DEFDBL	DEFSNG	DEFSTR	DIM	DRAW
EDIT	ELSE	END	EOF	EXP
FIELD	FILE	FILES	FIX	FL
FN	FOR	FORMAT	FRAC	FRE
GET	GOSUB	GOTO	IF	INIT
INKEY\$	INPUT	INT	IT	KEY
KILL	LEFT\$	LEN	LET	LGT
LIST	LLIST	LOAD	LOC	LOCATE
LOF	LOG	LPRINT	LRA	LRB
LSET	MEANX	MEANY	MIDS	MKDS
MKS\$	MOD	MOUNT	NEW	NEXT
NOT	ON	OPEN	OPTION	OR
OUTPUT	PASS	PEEK	POINT	POKE
PRINT	PROG	PUT	QUAD	RANDOMIZE
RC	READ	REM	RENUM	RESET
RESTORE	RETURN	RIGHT\$	RND	ROUND
RSET	RUN	SAVE	SDX	SDXN
SDY	SDYN	SGN	SIN	SQR
STAT	STEP	STOP	STR\$	SUMIT
SUMRC	SUMX	SUMX2	SUMXY	SUMY
SUMY2	SYSTEM	TAB	TAN	THEN
TIMES	TO	TROFF	TRON	USING
VAL	VERIFY	XOR		

3-3 程序

3-3-1 程序区

分成从PROG0到PROG9 10个程序区。可以分别输入独立的程序。并且可以根据程序的长度自动地改变程序区的大小。

各个程序的行号都互相独立，但变量在所有程序区及文件区中是通用的。

用PROG命令可以手动切换程序区。在程序执行中可用GOTO、GOSUB语句转移到其它程序区去执行。

3-3-2 程序的构成

程序执行时原则上按行号的顺序进行。

3-4 行

行是由行号与语句组成的。语句可用“:”符号区分开,而写成几个语句。因此行按下面形式写出。

行号 语句[:语句]*

注[]号可以省略。*表示其前的元素可以任意次数反复。详见3-11。

<例> 10 PRINT:PRINT A

行的长度在显示程序表时,是在255个字符以内。

● 行号

在一行的前面一定要有行号。另外,在GOTO等语句中,也使用行号。

行号范围 1~64999的整数

行号间的间隔可忽略。

<例> 10 PRINT:GOTO 10 和

 10 PRINT:GOTO 10 一样。

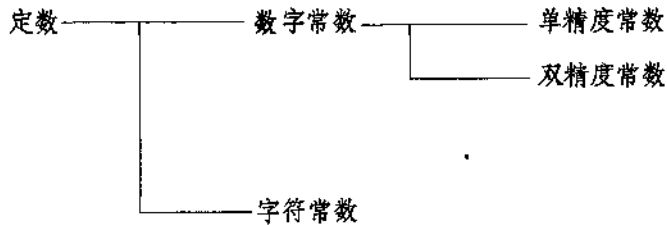
3 - 5 语句

语句的前部表示语句的种类、标明是语句或命令。

以下的文法部分，随语言的种类而不同。C95 - BASIC 中不特别区别命令与语句。主要是在直接方式中使用的命令，而在程序中使用的称语句。

3-6 常数

在程序中，直接代表某一值的称常数



3-6-1 数字常数

数字常量使用以下三种表示法。可以使用“+”或“-”符号，可以用它们将任意个常量连接起来使用。数字前有多个符号时，只数“-”的个数，“-”是奇数个时，变换成一个“-”号。“-”是0或偶数个时，变换成“+”。（与数学规则相同）。

符号与符号间的间隔，符号与数字间的间隔可忽略。

(1) 整数表示

任意个“+”或“-”号后，排列1个以上个数字。

<例> 1 2 5
 - 5 2
 --+- 4 5 (= - 4 5)

(2) 固定小数点表示

在任意个符号之后，按整数、小数点、小数部这样顺序写。整数、小数可有任意个数字。可是整数和小数部分都要写上数字。

<例> +1.2 3
 +- .4 5 (= - 0.4 5)
 2.3 7
 2. (= 2)

(3) 浮动小数点的表示

意思为在固定小数点表示之后加上指数部分。指数部分由E和任意个符