Министерство высшего и среднего
специального образования РСФСР

Куйбышевский ордена Трудового Красного Знамени
авиационный институт имени академика С.П.Королева

ВВЕДЕНИЕ В ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ ПЛ/I

Утверждено редакционным
советом института
в качестве методических указаний

Куйбышев 1983

УДК 681.3.06

Описано минимальное, с точки зрения используемых средств полного языка, подмножество ПЛ/I для решения вычислительных задач. Указания предназначены для студентов I-го курса, выполняющих задания по курсу "Введение в программирование" и "Работа в ВЦ", а также для студентов других специальностей, выполняющих курсовые проекты с использованием языка ПЛ/I.

При изучении методических указаний следует помнить, что введенные ограничения относятся только к выбранному подмножеству.

Составитель - В.В.П ш е н и ч н и к о в

Рецензенты: доц. А.Л.Гуревич, доц. В.М. Радомский

I. АЛФАВИТ ЯЗЫКА

Для записи программы на языке ПЛ/I разрешено использовать следующие символы:

заглавные буквы латинского алфавита:
A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, R, S, T
U, V, W, X, Y, Z;
цифры: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9;

специальные символы: пробел (изображается действительным пробелом), = (знак равенства или символ присваивания), + (плюс), - (минус), * (звездочка или символ умножения), / (косая черта или символ деления), ( ) (скобки), , (запятая), ; (точка с запятой), : (двоеточие), . (точка), ' (кавычка), ¬ (символ НЕ), & (символ И), | (символ ИЛИ)*, > (больше), < (меньше), _ (знак подчеркивания).

Для записи комментариев, символьных констант, а также обрабатываемой информации разрешается использовать также русские буквы.

2. СТАНДАРТНЫЕ ОБОЗНАЧЕНИЯ

Для компактного изложения правил синтаксиса языка в тексте методических указаний использованы следующие стандартные обозначения (метаязык):

1. Элементы языка обозначаются либо их названиями, либо ключевыми словами ПЛ/I.

2. Если элемент заключен в квадратные скобки, то это означает, что его наличие в конструкции необязательно.

3. Из столбца элементов, заключенных в фигурные скобки, нужно выбирать только один элемент.

4. Символ "пробел" изображается символом ␣ .

---

* На устройствах подготовки данных вместо знака | имеется ! , его и следует писать на бланке для перфорации

3. ПЕРЕМЕННЫЕ И КОНСТАНТЫ

В выбранном подмножестве будем использовать два типа переменных (целые и действительные) и четыре типа констант.

Для обозначения переменных используются имена, состоящие из букв и цифр, начинающиеся обязательно с буквы, длиной не более 31 символа. Русские буквы в именах не допускаются. Внутри имени допускается символ "_" (подчеркивание), который удобно использовать для имен, состоящих как бы из нескольких слов.

Примеры имен: AB13, TIME, BEC_DETALI, SUMMA_NA_RUKI.

Целые переменные должны быть объявлены в операторе DECLARE с атрибутами FIXED BINARY , а действительные с атрибутами FLOAT DECIMAL .

Для необъявленных переменных система выбирает тип по первой букве имени: если имя начинается с букв из набора I , J , K , L , M , N, то переменная будет целой, в противном случае действительной.

Диапазон представления целых чисел от -2^{31} до $2^{31}-1$, действительных от 10^{-78} до 10^{75}. Целые представляются в виде целых десятичных чисел со знаком или без него, например: -10, 2, 1525. Действительные в виде десятичных чисел в форме с фиксированной или плавающей точкой. С фиксированной точкой: -2.5, 0.25, -4.75. С плавающей: 0.525E + 11, -5.475E -02.

4. МАССИВЫ

Массивом называется n -мерная совокупность элементов данных, имеющих одинаковые атрибуты. Для объявления массивов используется оператор в следующем виде:

DECLARE имя массива (атрибуты размерности)
атрибуты представления;

Атрибут размерности показывает количество измерений массива (не более 3) и верхнюю границу каждого измерения (нижняя всегда равна 1).

Примеры объявлений

```
DECLARE  A(10) FLOAT DECIMAL,
         B(5,5) FIXED BINARY;
```

Для обращения к отдельным элементам массива используются индексированные переменные в виде:

имя массива (индексы).

Индексы задаются целыми десятичными цифрами или выражениями.

Так запись A(2) указывает на 2-й элемент массива A, а запись B(4,2)-на 2-й элемент 4-й строки матрицы B.

Индексированные переменные могут быть использованы в арифметических и логических выражениях.

В памяти машины элементы массива записываются последовательно, причем первым изменяется младший (правый) индекс.

5. ОБЩАЯ СТРУКТУРА ПРОГРАММЫ

Программа на языке ПЛ/I имеет следующий вид:

```
ИМЯ: PROCEDURE OPTIONS (MAIN);
     текст программы
END  ИМЯ;
```

И м я программы должно состоять не более чем из 6 символов (букв и цифр) и начинаться с буквы.

Т е к с т п р о г р а м м ы обычно включает в себя следующую последовательность операторов:

1. Операторы объявления данных;
2. Операторы ввода данных;
3. Операторы, реализующие логику программы (базовые структуры);
4. Операторы печати результатов.

В любое место текста программы могут быть включены комментарии, поясняющие логику программы или объявленные переменные. Комментарием является набор символов вида

/* символы */

Попробуйте до изучения всех операторов языка разобраться в следующей программе:

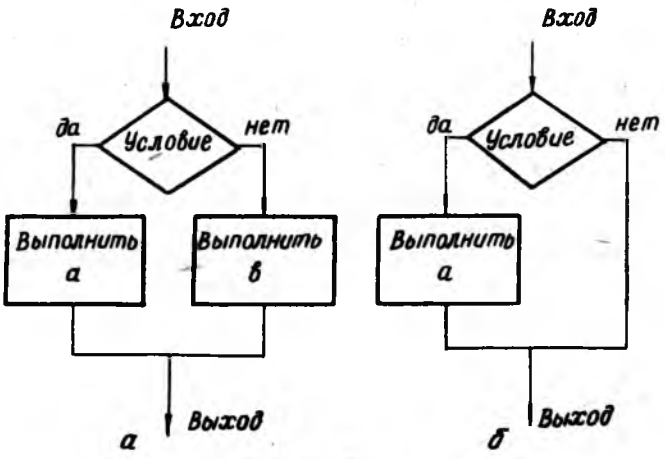
```
PRIM1; PROCEDURE OPTIONS (MAIN);
DECLARE (A,B,C,X1,X2) DECIMAL FLOAT;
GET LIST (A,B,C); /* ВВОД ЗНАЧЕНИЙ */
X1 = (-B + SQRT(B**2 - 4*A*C))/(2*A);
X2 = (-B - SQRT(B**2 - 4*A*C))/(2*A);
PUT EDIT(X1,X2)(SKIP, E(10,2));
END PRIM1;
```

6. ЭЛЕМЕНТЫ СТРУКТУРНОГО ПРОГРАММИРОВАНИЯ

Логическая структура любой программы может быть представлена комбинацией трех базовых структур: следование, развилка (ветвление), цикл (повторение). Каждая структура имеет один вход и один выход.

Структура с л е д о в а н и е означает, что два действия должны быть выполнены друг за другом.

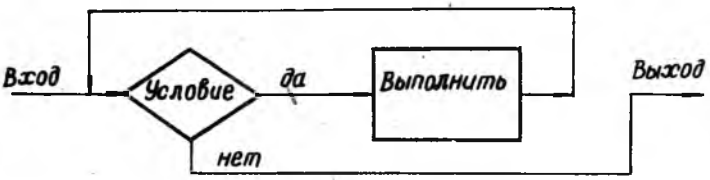
Структура р а з в и л к а обеспечивает выбор одного из двух альтернативных путей. После проверки условия выбирается только один из них (рис. I,а).



Р и с. I

Если для одного из результатов проверки ничего делать не надо, то структура принимает вид, показанный на рис. I,б.

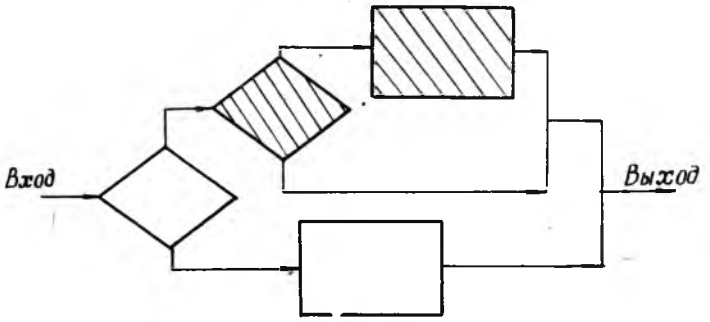
Структура ц и к л определяет циклическое (повторное) выполнение некоторой части программы и является необходимой для большинства программ (рис. 2).



Р и с. 2

Условие проверяется перед первым вхождением в цикл и перед каждым новым повторением. Если перед первым вхождением условие не выполняется (ложь), то цикл не будет вообще выполняться. Цикл может включать на месте "выполнить α " группу операторов. Такая структура называется ЦИКЛ-ПОКА и означает, что цикл повторяется, пока условие остается истинным.

Базовые структуры могут комбинироваться сколь угодно разнообразно , как того требует логика программы. Таким образом, где бы на блок-схеме не появился отдельный прямоугольник, он может быть заменен на любую базовую структуру (рис. 3).



Р и с. 3

Здесь один из блоков в структуре является развилкой.

Использование базовых структур позволяет программировать без оператора *GOTO* (этот оператор не включен в выбранное подмножество), что существенно снижает количество ошибок в программе и обеспечивает выполнение программы только сверху вниз. Конечно, некоторые части программы будут пропускаться и циклически повторяться, но все разветвления будут направлены только вниз (вперед), что существенно упрощает чтение и изменение программы.

7. ОПЕРАТОР ПРИСВАИВАНИЯ

Оператор присваивания состоит из двух частей, разделенных символом присваивания:

имя переменной = выражение;

Левая часть - это обязательно имя переменной, правая часть - арифметическое выражение, значение которого нужно присвоить переменной в левой части. Значение выражения всегда преобразуется к типу переменной, стоящей в левой части оператора присваивания. Так например, если в левой части целая (*FIXED BINARY*) переменная, а значение выражения - действительное число (*FLOAT DECIMAL*), то результат будет преобразован к целому типу отбрасыванием дробной части.

Выражением в выбранном подмножестве может быть только арифметическое выражение. Арифметическое выражение строится из констант, переменных и функций с помощью знаков арифметических операций и круглых скобок.

К арифметическим операциям относятся сложение (символ +), вычитание (символ -), деление (/), умножение (*) и возведение в степень (**).

Выражения в круглых скобках вычисляются в первую очередь. При отсутствии скобок наиболее высоким приоритетом обладают операции возведения в степень, одноместные плюс и минус, далее умножение и деление и, наконец, сложение и вычитание.

Если последовательно записаны операции одного уровня приоритетов, то операции возведения в степень выполняются справа налево, все остальные-слева направо.

Запись арифметических выражений в одну строку требует большего

по сравнению с математической записью количества скобок.

Сравните математическую запись формулы $y = \frac{a+b\cos x}{a-b}$

с записью на языке ПЛ/I `Y=(A+B*COS(X))/(A-B)`

Приведем наиболее употребительные встроенные функции*. Это функции: `SIN, SIND, COS, COSD, TAN, TAND, ATAN, ATAND, SQRT, EXP, LOG, LOG2, LOG10, ABS.`

Аргументами встроенных функций могут быть выражения (кроме функции `ABS`). Встроенные функции `SIN`, `COS`, `TAN` используют аргумент в радианах, а `SIND`, `COSD`, `TAND` – в градусах. Результатом встроенной функции `ATAN` является значение $arctg$ в радианах, а `ATAND` – в градусах.

`SQRT(X)` соответствует $\sqrt{x}$, `EXP(X)` – e^x, `LOG(X)` – $\ln x$, `LOG2(X)` – $\log_2 x$, `LOG10(X)` – $\lg x$.

Примеры.

Математическая запись	Запись на языке ПЛ/I
$\frac{ab}{c}$	`A*B/C`
$\sin^2 x$	`SIN(X)**2`
$e^{-\sin x}$	`EXP(-SIN(X))`
$\sqrt{\lvert tg\, x \rvert}$	`SQRT(ABS(TAN(X)))`

8. ВЫРАЖЕНИЯ С МАССИВАМИ

Массивы могут использоваться как непосредственные операнды в арифметических выражениях. Над ними поэлементно выполняются все арифметические операции. Массивы, участвующие в выражениях, должны иметь одинаковые размерности и границы соответствующих измерений. Результатом будет новый массив той же размерности и таких же границ.

---

* Подробно все встроенные функции полного языка ПЛ/I рассмотрены в методических указаниях "Встроенные функции языка ПЛ/I", КуАИ, 1982 г.

Примеры.

I. Объявлены матрицы

```
DECLARE (A(2,2), B(2,2), C(2,2))
        DECIMAL  FLOAT;
```

После выполнения оператора

```
C = A+B;
```

Элементы матрицы получат значения, вычисленные по выражениям:

$$C_{11} = a_{11} + b_{11} \qquad C_{12} = a_{12} + b_{12}$$

$$C_{21} = a_{21} + b_{21} \qquad C_{22} = a_{22} + b_{22}$$

2. для тех же объявлений при выполнении оператора A = 0; все элементы массива A получат нулевые значения, а при выполнении оператора A = B, элементы массива A получат значения соответствующих элементов массива B.

Массивы могут быть аргументами математических встроенных функций SIN, SIND, COS, COSD, TAN, TAND, SQRT, EXP, LOG, LOG2, LOG10, ABS.

В этом случае результатом выполнения функции является массив с размерностью и границами массива аргумента.

Пример.
```
DECLARE (E(2,2), F(2,2)) DECIMAL FLOAT;
        E = SIN(F);
```

Каждый элемент массива E равен синусу соответствующего элемента массива F .

9. УСЛОВНЫЙ ОПЕРАТОР

Условный оператор реализует одну из основных структур – в е т в л е н и е (или развилка) и используется в полной или неполной форме (соответствующие им базовые структуры рассмотрены в разд. 6).

Полная форма оператора:

```
IF    условие  THEN  оператор I ;
               ELSE  оператор 2 .
```

Неполная форма

```
IF    условие  THEN  оператор I ;
```

Здесь если условие – истина, то выполняется оператор I, если ложь, то выполняется оператор 2 для полной формы. После этого управление переходит следующему оператору.

У с л о в и е - это логическое выражение.

Простое логическое выражение строится из арифметических выражений, связанных операциями сравнения. Операции сравнения это: равно (=), неравно (¬=), больше (>), меньше (<), больше или равно (>=), меньше или равно (<=), не больше (¬>), не меньше (¬<). Результат такого выражения истина или ложь.

Простые логические выражения могут объединяться в сложные с помощью знаков логических операций НЕ(¬),И(&) или (|).

Примеры логических выражений:

```
A>B     A+B>5     X<5&X>=7
```

Операторы I и 2 могут быть любыми операторами, в том числе и условными. Следует иметь в виду, что если условный оператор вкладывается в THEN - ветвь, то он должен быть записан в полной форме, в противном случае нарушится соответствие между ELSE и IF .

Примеры записи условных операторов:

```
IF  A>B  THEN  Y=A*X**2;
         ELSE  Y=A*X;

IF  A>5  THEN  Y=A+C;
         ELSE  IF A<3&A>1 THEN Y=B-C;
                          ELSE Y=B/C;
```

Если в THEN -ветви или ELSE -ветви должно стоять несколько операторов, то необходимо объединить их в DO -группу.

DO -группа имеет вид:

```
DO; оператор I;
    оператор 2;
    . . . . . .
    оператор N;
END;
```

IO. ОПЕРАТОР ЦИКЛА

IO.I. Общая форма оператора цикла. Оператор цикла существует в двух различных формах:

1. DO WHILE (условие);
   операторы
   END;
2. DO переменная = спецификация [, спецификация] ...;
   операторы
   END;

В форме 2 спецификация имеет вид

выражение 1 [[TO выражение 2 [BY выражение 3]] [BY выражение 3 [TO выражение 2]]] [WHILE (условие)]

10.2. Правила выполнения оператора. Приведем общие правила использования оператора :

условие (см. раздел условный оператор);

режим WHILE указывает, что перед очередным (может быть первым) входом в цикл происходит проверка условия;

переменная в форме 2 должна быть неиндексированной переменной;

выражения в спецификации означают : выражение 1 (начальное значение переменной цикла); выражение 2 (конечное значение), выражение 3 (шаг);

если опущено "BY выр.3" и есть "TO выр.2", шаг выбирается равным единице;

если опущено "TO выр.2", выход из цикла возможен только по WHILE или по оператору внутри цикла;

если опущены "TO выр.2" и "BY выр.3", цикл выполняется только один раз;

если записано несколько спецификаций, каждая новая спецификация означает новый вход в цикл с новыми граничными условиями;

управление нельзя передавать внутрь цикла, минуя заголовок цикла;

внутренними по отношению к циклу могут быть любые операторы, в том числе и операторы цикла (вложенные циклы).

10.3. Примеры использования оператора цикла. Приведем примеры использования оператора цикла :

а) DO X = 1 TO 10 ;
переменная принимает значения 1,2,3,...,10;

б) DO X = 2 TO 10 BY 2 ;
принимает значения 2,4,6,8,10;

в) `DO X = 4 WHILE (B<3);`

цикл выполнится один раз для $X = 4$ (при $B<3$) или не выполнится вообще при $B \geq 3$.

г) `DO X = I, 3 TO 5, 7;`

принимает значения I,3,4,5,7;

д) `DO Y = I0 TO I BY -I;`

принимает значения I0,9,8,7,6,5,4,3,2,I;!

е) `DO WHILE (P=0) ;`

цикл выполняется пока $P = 0$. Для выхода из цикла P должно принять отличное от нуля значение внутри цикла.

I0.4. Примеры циклических программ. Рассмотрим несколько примеров программ.

I. Получить значения функции $y = e^{-A\sin x}$ для X , изменяющегося от 0 до π шагом $\pi/30$.

```
PRIM2: PROCEDURE OPTIONS(MAIN);
DECLARE (X,Y,A,PI,H) DECIMAL FLOAT;
PI = 3.14159; H = PI/30;
DO X = 0 TO PI BY H;
 Y = EXP(-A*SIN(X));
 PUT EDIT(X,Y)(SKIP,P(8,5),X(2),E(12,5));
 END;
END PRIM2;
```

2. Определить номер первого элемента массива A_i , $i = I,2,...,I0$, большего нуля.

```
PRIM3: PROCEDURE OPTIONS(MAIN);
DECLARE A(10) DECIMAL FLOAT,
         (I,P) FIXED BINARY;
```

/* перед входом в цикл $P = 0$, если элемент, больший нуля, найден, делаем $P = I$, если не найден, оставляем $P = 0$ */

```
P=0;
DO I=1 TO 10 WHILE(P=0);
   IF A(I)>0 THEN P=1;
END;
```

```
/* выход из цикла возможен при  P ≠ 0  или при I > I0 */
      I=I-1; /* т.к. перед проверкой условия I увеличи-
                лось на 1 */.
IF P=1 THEN PUT EDIT      ('номер элемента равен', I )
                          (SKIP, A, X(2), P(2));
ELSE PUT EDIT             ('элемент не найден')(SKIP,A);
END PRIM3;
```

3. Программа умножения матриц

```
PRIM4: PROCEDURE OPTIONS(MAIN);
 DECLARE (A(5,5), B(5,5), C(5,5)) DECIMAL FLOAT,
          (I,J,K) BINARY FIXED;
 GET LIST (A,B);
 DO I=1 TO 5;
 DO J=1 TO 5;
  C(I,J)=0;
  DO K=1 TO 5;
  C(I,J)=C(I,J)+ A(I,K)*B(K,J);
  END;
 END;
END;
PUT EDIT(C)(SKIP, 5(X(2), E(10,3)));
END PRIM4;
```

II. ОПЕРАТОРЫ ВВОДА ДАННЫХ С ПЕРФОКАРТ

II.I. Структура операторов *GET LIST* и *GET EDIT*. Данные на перфокартах подготавливаются в кодах перфокарт и могут занимать все 80 позиций. Рассмотрим два оператора ввода *GET LIST* и *GET EDIT*.

Для оператора *GET LIST* (список данных) данные перфорируются в виде десятичных констант в разрешенных формах и отделяются

друг от друга одним или несколькими пробелами.

По операторам GET переменные, указанные в списке данных, последовательно одна за другой получают значения из потока данных на перфокартах, но для GET LIST значения отделены друг от друга пробелами, а для оператора GET EDIT выделение нужной части данных осуществляется по форматам.

Оператор имеет вид GET EDIT (список данных) (список форматов).

Список данных у операторов GET LIST и GET EDIT имеет вид

(элемент [, элемент] [, элемент] ...)

В качестве элементов списка можно использовать простые переменные, массивы и повторяющиеся спецификации.

Повторяющаяся спецификация имеет вид

элемент [, элемент] ... DO переменная = спецификация).

Повторяющаяся спецификация может быть вложенной, т.е. элемент повторяющейся спецификации может быть сам повторяющейся спецификацией.

Повторяющаяся спецификация используется в тех случаях, когда нужно ввести не все элементы массива, или в отличной от предусмотренной в ПЛ/I последовательности, например:

```
GET LIST(((A(I,Y) DO I=1 TO 2)DOY = 3TO4));
```

эквивалентен следующему циклу :

```
DO J=3 TO 4;
 DO I=1 TO 2;
  GET LIST(A(I,Y));
 END;
END;
```

I4.2. Примеры использования оператора GET LIST . Приведем примеры использования оператора GET LIST .

I. Ввести с одной перфокарты значение переменных $a = 2{,}5$ и $b = 3{,}7$.

Используем оператор GET LIST(A,B);

Данные должны быть подготовлены на перфокарте следующим образом

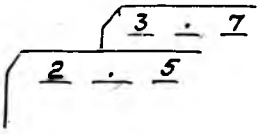
```
2.5 3.7
```

2. Ввести значения переменных a = 2,5 и b = 3,7 с отдельных перфокарт.

Оператор ввода будет иметь вид

```
GET LIST(A,B),
```

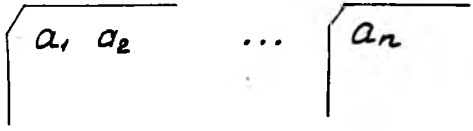
а данные



3. Ввести значения элементов массива

```
DECLARE A(10) DECIMAL FLOAT;
GET LIST(A);
```

Данные подготавливаются на перфокартах в следующей последовательности



II.3. Список форматов оператора `GET EDIT`. Формат `F(l[,d])` для ввода чисел с фиксированной точкой, где l знаков из потока данных интерпретируется как число, причем d цифр числа принадлежит дробной части.

Если десятичная точка содержится на перфокарте, то d - игнорируется.

Число на карте представляется в следующем виде :

[␣ ...] [[±]] [цифра ...] . [цифра ...] [␣...]

Формат `E(l,d)`, где l позиций на перфокарте интерпретируется как число в форме с плавающей точкой, а d позиций принадлежит дробной части мантиссы, требует следующего вида числа на перфокарте :

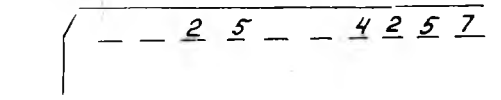
Формат $A(\ell)$ вводит ℓ позиций с перфокарты как строку символов. Формат $X(\ell)$ пропускает ℓ позиций на перфокарте без чтения.

Перед отдельным форматом или группой форматов может быть коэффициент повторения.

II.4. Примеры использования оператора GET EDIT . Приведем примеры использования оператора GET EDIT .

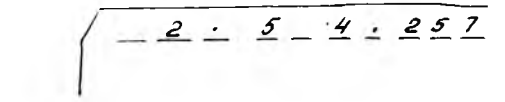
I. При выполнении оператора (A,B)(F(4,1),F(6,3));

для перфокарты вида

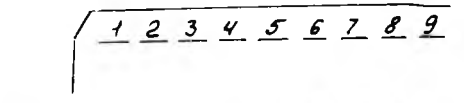


переменная A получит значение 2,5 , а переменная B - 4,257.

2. Те же значения могут быть введены и в следующем виде



3. С перфокарты вида



Оператором вида GET EDIT(A)(F(1)) элементам массива A_i , i = I,2, ... , 9 будут присвоены следующие значения A_1 = I, A_2 = 2, A_3 = 3, ... , A_9 = 9.

I2. ОПЕРАТОР ВЫВОДА ДАННЫХ НА ПЕЧАТЬ

I2.I. Общая форма оператора. Оператор вывода значений на пе-

чать имеет вид PUT EDIT

(список данных) (список форматов).

Элементами описка данных могут быть переменные, массивы, выражения и константы.

Так,оператор

PUT EDIT (A, B+C ,'ТАБЛИЦА', I5, C) (список форматов) ,

если A – массив из четырех чисел,а B и C простые переменные, будет печатать значения элементов массива, значение суммы переменных B и C , слово ТАБЛИЦА, число I5 и значение переменной C .

I2.2. Форматы данных. В списке форматов могут быть использованы следующие форматы данных :

F – для печати целых и действительных чисел в форме с фиксированной точкой;

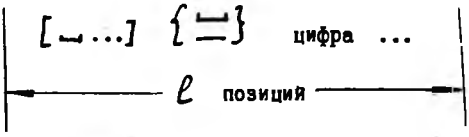
E – для печати чисел в форме с плавающей точкой;

A – для печати символьных данных.

Формат F записывается в форме F(l[,d]), где l – число позиций в строке печати, предназначено для числа, d – число цифр в дробной части числа.

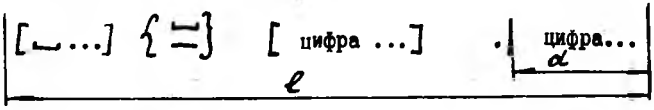
Рассмотрим различные случаи сочетания значений и форматов.

I. Используется формат F(l). Вид строки печати

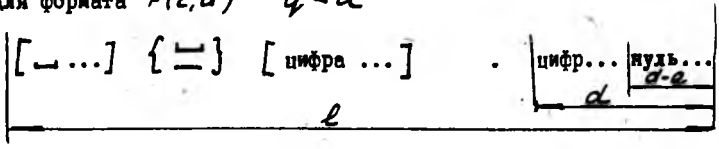


2. Для формата F(l) число цифр в целой части числа больше, чем l . Все l позиций на печати заполняются символами * .

3. Используется формат F(l,d) и число цифр в дробной части q больше или равно d .



4. Для формата F(l,d) q<d



5. Для формата $F(\ell,d)$ число цифр в целой части числа вместе с указанным для вывода числом дробных цифр больше ℓ. Все ℓ позиций на печати заполняются символами $*$.

Формат E записывается в форме $E(\ell,d[,s])$, где ℓ – общее число позиций для печати числа, d – число цифр в дробной части мантиссы, s – число значащих цифр в мантиссе (общее число цифр в целой и дробной части мантиссы);

$[\sqcup \ldots]\{\overset{\sqcup}{-}\}$ мантисса $E\{\overset{+}{-}\}$ порядок

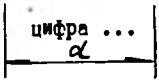
Чтобы при любых значениях чисел печать произошла, необходимо выбирать $\ell - d \geq 7$.

Для правильного выбора значения s необходимо рассмотреть 4 случая печати мантиссы.

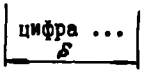
I. $s > d$

2. $s < d$

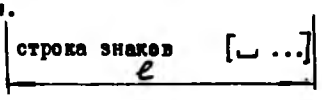
3. s отсутствует



4. d = 0, т.е. мантисса печатается как целое число



Формат A записывается в виде $A(\ell)$, где ℓ – общее число позиций на строке печати для символьной строки. Символьные строки "прижимаются" к левому краю отведенного поля и дополняются, если необходимо, пробелами.



Если ℓ не указано, то информация о длине берется из оператора DECLARE или, при использовании в списке данных символьной константы, ℓ принимается равной длине константы.

I2.3. Управляющие элементы форматов. Управляющими элементами формата для печати являются X , SKIP , LINE , PAGE , COLUMN.

Формат $X(\ell)$ означает, что в строку печати вставляется ℓ пробелов.

Формат $SKIP(n)$, где $n = 0, 1, 2, 3$, производит печать текущей строки, после чего пропускает $n-1$ строк. Печать следующего элемента производится с первой позиции установленной строки.

$SKIP(0)$ производит печать текущей строки, но строку не переводит. Следующая печать будет производиться в той же строке с I-й позиции. SKIP означает то же, что $SKIP(1)$.

Формат PAGE означает переход на новую страницу печати. Стандартный размер страницы - 60 строк.

Формат $LINE(n)$ означает переход на строку n внутри страницы. Если в текущей странице строка n уже напечатана, то происходит переход к строке n на новой странице.

Формат $COLUMN(S)$ означает переход на позицию внутри строки. Если позиция S уже занята, то происходит печать с S -й позиции новой строки.

I2.4. Режимы печати. Возможна следующая форма оператора печати:

$$PUT \left\{ \begin{array}{l} PAGE \\ PAGE\ \ LINE\ (\text{выражение}) \\ SKIP\ (\text{выражение}) \\ LINE\ (\text{выражение}) \end{array} \right\} EDIT(\ )(\ )$$

Управляющие слова сохранят свой смысл, но управление происходит не во время печати, а до нее, и, кроме того, значения выражений можно изменять в процессе работы программы.

I2.5. Примеры использования оператора. Приведем следующие примеры использования оператора.

I. Напечатать строку ТАБЛИЦА в середине страницы

```
PUT EDIT('ТАБЛИЦА')(PAGE, LINE(31), COLUMN(57), A);
```

2. Напечатать значение переменной Y с I5-й позиции 3-й строки.

```
PUT PAGE LINE(3) EDIT(Y)(COLUMN(15),F(4));
```

ПРИЛОЖЕНИЕ
СПИСОК ИСПОЛЬЗУЕМЫХ В ПОДМНОЖЕСТВЕ КЛЮЧЕВЫХ СЛОВ

Ключевыми словами языка ПЛ/I являются английские слова (приведены их переводы на русский язык и фонетическая транскрипция для правильного произношения) :

BINARY [bainarɪ] - двоичный;
BY [bai] - через;
COLUMN ['kɔlam] - колонка (здесь позиция в строке);
DECIMAL ['desiməl] - десятичный;
DECLARE [dɪ'klɛə] - объявить;
DO [du:] - выполнять;
EDIT ['edit] - редактировать;
ELSE ['els] - иначе;
END [end] - конец;
FIXED [fikst] - фиксированная (точка);
FLOAT ['flout] - плавающая (точка);
GET [get] - получать;
GOTO ['gou tu:] - идти к ;
IF [if] - если;
LINE [lain] - строка;
LIST [list] - список;
PAGE [peidʒ] - страница;
PROCEDURE [prəsi:dʒə] - процедура;

PUT [put] - поместить;
SKIP [skip] - скачок (здесь на новую строку);
THEN [ðen] - то (тогда);
TO [tu:] - к ;
WHILE [wail] - пока.

Составитель – Виктор Владимирович П ш е н и ч н и к о в

ВВЕДЕНИЕ В ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ ПЛ/I

Редактор Е.Д.А н т о н о в а
Техн.редактор Н.М.К а л е н ю к
Корректор Н.С.К у п р и я н о в а

Подписано в печать 27.12.83 г. Формат 60x84 1/16.
Бумага оберточная белая. Оперативная печать.
Усл.п.л. 1,39. Уч.-изд.л. 1,25. Т. 500 экз.
Заказ 1895 Бесплатно.

Куйбышевский ордена Трудового Красного Знамени
авиационный институт имени академика С.П.Королева,
г. Куйбышев, ул. Молодогвардейская, 151.

Областная типография им. В.П.Мяги,г. Куйбышев,ул.Венцека,60.