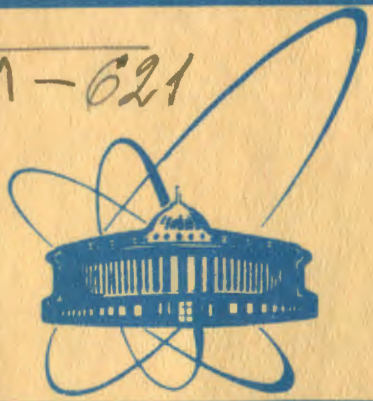


сообщения
объединенного
института
ядерных
исследований
дубна

11-81-673

Ли Рен Хи

ДВУМЕРНОЕ ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ АЛГОЛ-60

1981

Данная работа является результатом развития идеи двумерного программирования, изложенной в [1], для языка CDL , первоисточником которой является работа [2]. В дальнейшем мы будем предполагать, что читатель знаком также с работой [3], которая посвящена двумерному изображению программ, кодируемых с помощью машинно-ориентированных языков. Алгоритмы, записанные на языке Алгол-60 [4], мы будем изображать YES-NO графом. Например, условный оператор

```
if <условие> then <безусловный оператор>
           else <оператор> ;
```

мы изображаем таким образом:

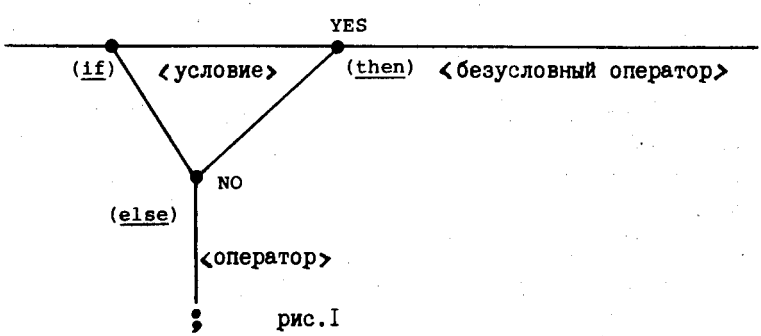


рис.I

Ключевые слова if, then, else изображены точками в углах треугольного узла, в дальнейшем ключевые слова мы будем опускать. Горизонтальная ветвь узла соответствует ветви then , вертикальная - ветви else . Порядок нагружения ребер графа стандартный - сначала нагружается горизонтальное ребро и т.д. (см. [1], [3]).

Классификация операторов и метод обозначения

Классификацию операторов языка Алгол-60 мы проведем с точки зрения наличия в операторах передачи управления.

I) Безусловные операторы.

Это - оператор присваивания, пустой оператор, небулевские процедуры, составной оператор и блок.

Примеры:

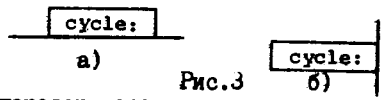
```
x:=x+1;
y:=x+5;
```
а)

```
x:=x+1;
y:=x+5;
```
б)

Рис.2

2) Левая метка.

Примеры наших обозначений:



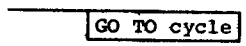
а)

б)

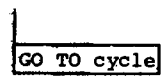
Рис.3

3) Оператор перехода.

Примеры



а)



б)

Рис.4

Отметим, что предлагаемые конструкции обозначений приводятся сразу для двух возможных случаев использования данной конструкции: на горизонтальном ребре (YES-ребро) и на вертикальном ребре (NO - ребро).

4) Условные операторы.

В начале данной работы мы дали полное описание конструкции для условного оператора.

Примеры.

I. Сокращенная форма условного оператора типа

if <условие>then <безусловный оператор> ;

Пусть дан оператор

```
if x = 0 then y:=y+1 ;
```

тогда его можно изобразить так:

Рис.5

2. Случай полной формы условного оператора

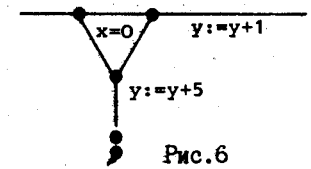


Рис.6

Этой диаграмме соответствует текст

```
if x=0 then y:=y+1 else y:=y+5 ;
```

5) Переключатель.

switch <идентификатор ключа> : = <список ключей> ;

Пусть дано выражение `switch k:=y,s,t,w;` тогда ему соответствует фрагмент

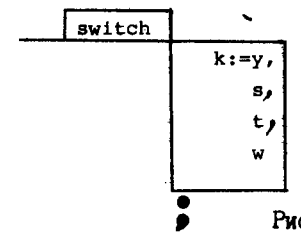


Рис.7

6) Операторы цикла.

Общая форма оператора цикла такая:

for <переменная> : =<список цикла> do <оператор> ;

В Алголе-60 имеется три типа элементов списка цикла: арифметическое выражение, элемент типа арифметическая прогрессия и элемент типа while.

Рассмотрим эти три варианта отдельно.

I. Элементом списка цикла является арифметическое выражение, тогда общая форма записи оператора цикла такова:

```
for <переменная>: =<арифметическое выражение I >,
                  <арифметическое выражение 2 >,
                   - - - - - - - - - - - - - - ,
                  < арифметическое выражение n>
              do < оператор> ;
```

Для конкретного примера

```
for x:=1.0, 2.5,3.2 do x:=x+5;
```

двумерное изображение будет таким:

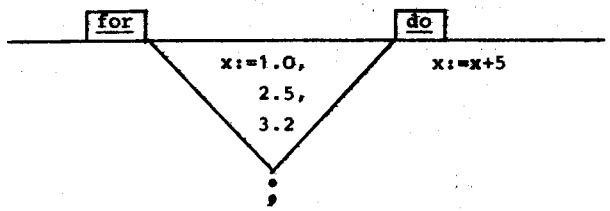


Рис.8

2. Оператор цикла с элементом списка типа арифметическая прогрессия записывается в общей форме следующим образом:

for <переменная>: = <арифметическое выражение 1>
step <арифметическое выражение 2>
until <арифметическое выражение 3>
do <оператор> ;

Пример:

```
for x:=1 step 1  until 10 do x:=x+5 ;
```

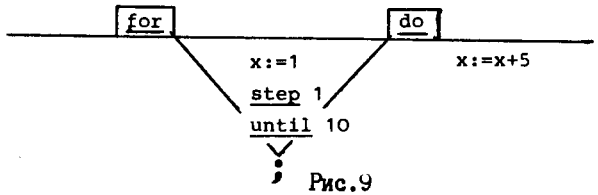


Рис.9

3. Оператор цикла с элементом списка типа while.

for <переменная>: = <арифметическое выражение>
while <логическое выражение >
do <оператор>;

Пример:

```
x:=0;
for  x:=x+1 while  x  50  do  print (x) ;
```

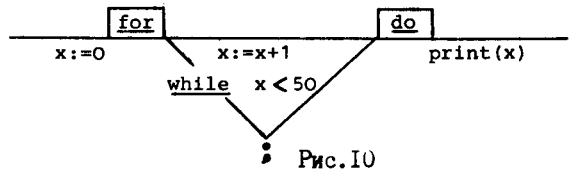


Рис.10

4. Пример программирования сложных циклов:

```
for i:=1 step 1 until 10 do
   for j=1 step 1 until 20 do y:=i+j  ;
```

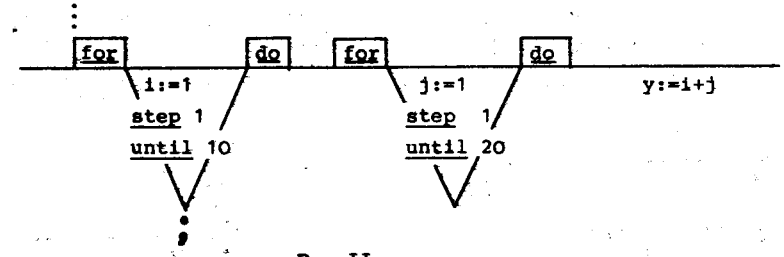


Рис.11

7) Операторные скобки begin, end , их обозначаем так:

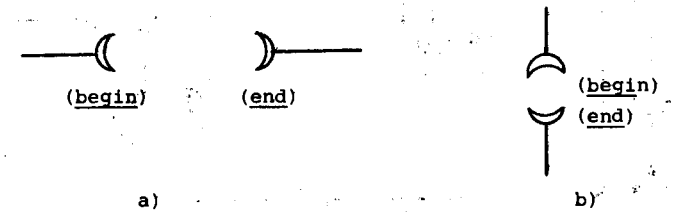


Рис.12

Правила двумерного программирования

Эти правила надо изучать после ознакомления с методами обозначения, указанными выше.

Порядок обозначения двумерного алгоритма таков: (см.рис.13).

1) Программа начинается операционной скобкой begin ((( ) .

2) Заполняется горизонтальное ребро.

3) Если требует алгоритм, записываются декларации, метки, оператор перехода, условные операторы, операторы цикла, операторные скобки.

Рассмотрим более подробно узлы, т.е. условные операторы и операторы цикла.

Если встретился узел, то за ним нагружается горизонтальное ребро;если за следующим узлом снова встретился узел,то снова записывается узел и т.д. до тех пор,пока будут встречаться узлы в алгоритме. После того как исчерпаны все узлы и горизонтальные ребра, формируются вертикальные ребра, начиная от точки последнего узла. Если при этом

встретился узел, то после его записи снова формируется горизонтальное ребро и так до тех пор, пока не будут исчерпаны все случаи, где текущий узел дает результат:"Истина", после чего опять формируется вертикальное ребро, начиная с последнего узла. Этот процесс происходит до тех пор, пока встречаются узлы при записи алгоритма сверху вниз по вертикальному ребру. Если на горизонтальном ребре лежат подряд два узла, то между узлами необходимо вставить скобку ((верх).

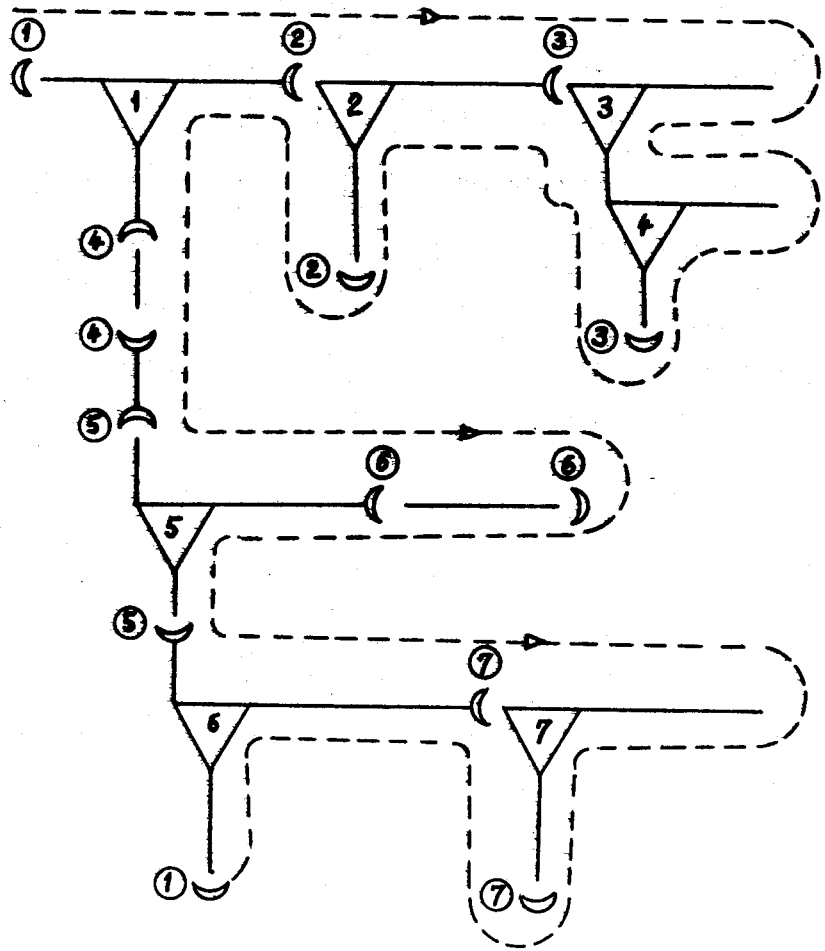


Рис.I3

4) Заканчивается программа скобкой)) (end) , которая может быть изображена знаком ⌣ , если есть в программе хотя бы один узел, и знаком)), если узлов в программе нет.

5) Операторные скобки ((,)) и ⌒ , ⌣ , встречающиеся в программе, должны располагаться попарно. При этом необходимо соблюдать следующие правила:
   - если begin изображается скобкой вертикального ребра ⌒, то end должен быть также изображен скобкой вертикального ребра ⌣;
   - если begin является скобкой горизонтального ребра ((и справа от этой скобки есть узел, то end должен быть изображен скобкой на вертикальном ребре самого левого узла ⌣;
   - если begin является скобкой горизонтального ребра ((, а справа от этой скобки нет узла, тогда end должен быть изображен скобкой на горизонтальном ребре)).

6) Большие программы, непомещающиеся на одном листе, соединяются с помощью символов —(i)— , (i)— , где i - есть номер фрагмента программы.

7) Потом, по двумерной программе,можно восстановить ее текст, если "разгрузку" графа проводить, начиная от начальной скобки ((по часовой стрелке, как это показано на рис.I3.

Примеры

I) Составить программу для вычисления значения функции

$$y = \begin{cases} x^2+3-\sqrt[3]{\pi-x}\ ; & \text{если } x<0 \\ (x^2+3)^2+\sqrt{0.5\pi+x}\ ; & \text{если } 0\le x<1 \\ x(x^2+3)+\ln(\pi+x)\ ; & \text{если } x\ge 1 \end{cases}$$

при любых значениях x . Предусмотреть выдачу на печать значения y и аргумента x , при котором функция y определена (см. /4/ с.45).

Решение

В каждой из ветвей расчетной формулы повторяется выражение x^2+3. Его значение необходимо вычислить предварительно, для чего введем дополнительную переменную

$$p = x^2 + 3 .$$

Блок-схема программы решения примера изображена на рис.I4.

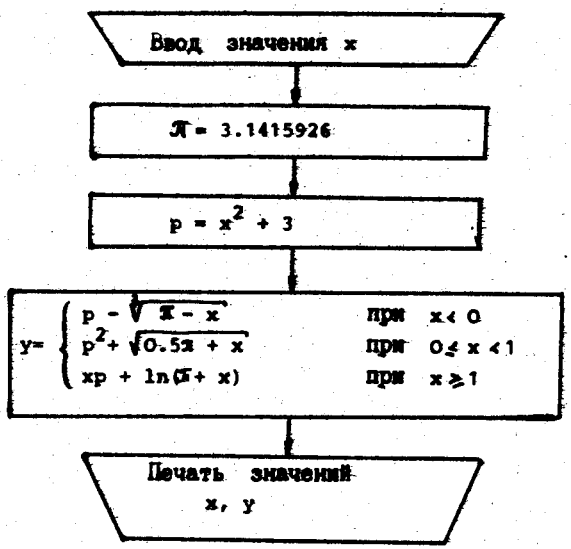


Рис.I4

Пояснительная информация об идентификаторах приведена в табл.I.

Таблица I

Информация об идентификаторах

Обозначение величины в алгоритме	Выбранный идентификатор	Класс величины	тип значений, принимаемых величиной
x	x	Простая переменная	real
y	y	то же	»
p	p	»	»
π	pi	»	»

Двумерное изображение программы решения примера I дано на рис.I5.

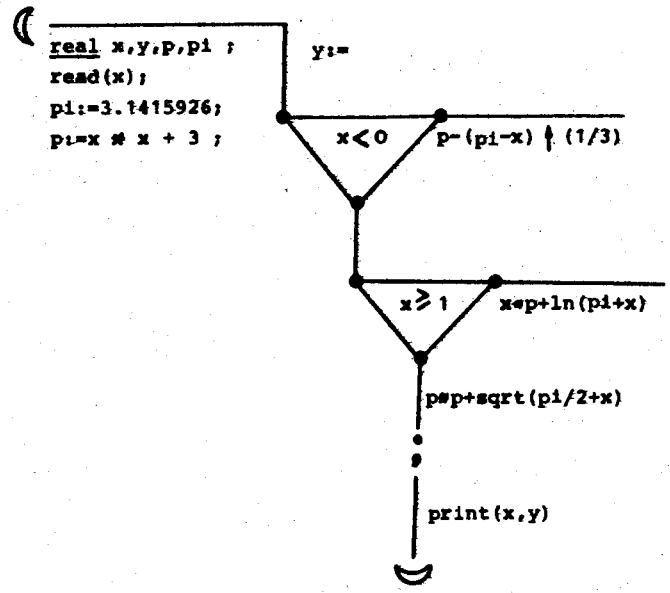


Рис.I5

Программа имеет вид:

```
begin real x,y,p,pi ;
      read (x);
      pi:=3.1415926;
      P :=x*x+3;
      y :=if x<0 then p-(pi-x) ↑ (1/3) else
          if x≥1 then x*p+ln(pi+x)
                 else p*p+sqrt(pi/2+x);
      print(x,y)
end
```

2) Вычислить последовательно:

I) Сумму кубов вторых 25 элементов одномерного массива

$A=(a_i),\ i=1,2,\ldots,100$;

2) Длину n-мерного вектора w при n=25;

3) Наибольший по модулю элемент массива

$B=(b_j),\ j=1,2,\ldots,25.$

(см. /4/ с.74).

Решение

Запишем расчетные формулы:

$$s=\sum_{i=26}^{50} a_i^3 \quad , \quad p=\sqrt{\sum_{k=1}^{25} w_k^2} \quad , \quad b_i=\max_j |b_j| \quad .$$

В этих формулах переменные k и j обладают следующими общими свойствами: они целые и изменяются по закону арифметической прогрессии от значения, равного единице, до конечного значения, равного 25, с шагом, равным единице.

Блок-схема программы решения примера изображена на рис.I6.

Пояснительная информация об идентификаторах приведена в табл.2.

Таблица 2

Обозначение величины в алгоритме	Выбранный идентификатор	Класс величины	Тип значений, принимаемых величиной
s	s	Простая переменная	real
p	p	то же	»
m	m	»	»
l	l	»	integer
i	i	»	»
a	a	Массив	real
b	b	»	»
w	w	»	»

Двумерное изображение программы решения примера 2 дано на рис.I7.

Программа имеет вид:

```
begin integer l,i;
      real s,p,m;
      array a [1:100] ,b,w, [1:25 ];
      read (a,b,w)
      s:=p:=m:=0;
      for i:=1 step 1 until 25 do
          begin s:=s+a [i+25] ↑ 3;  p:=p+w [i ]↑2 ;
            if abs (b [i] )>  abs(m) then
              begin l:=i; m:=b [i ] end ;
      end;
      p:=sqrt(p);
      print(s,p,m,l)
end
```

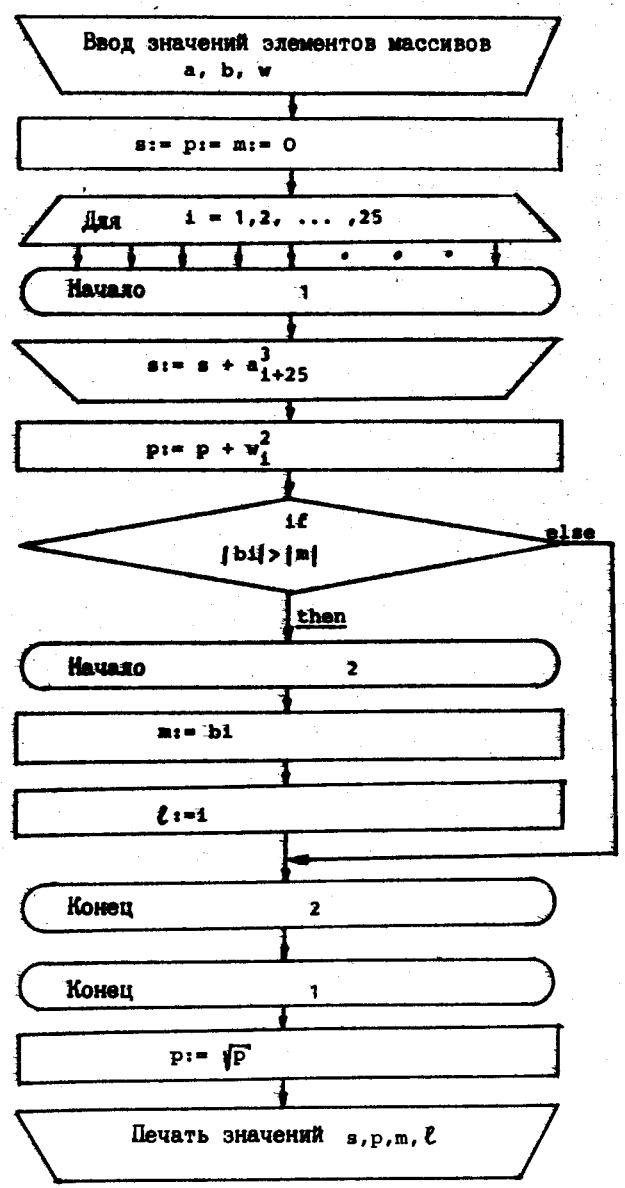



Рис.I6

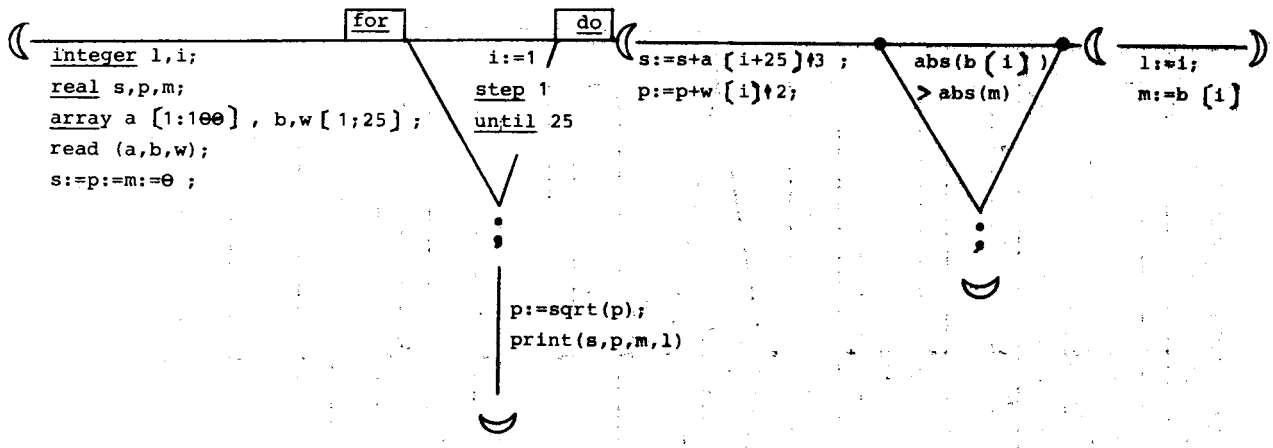

for
do
integer l,i;
real s,p,m;
array a [1:100] , b,w[1;25] ;
read (a,b,w);
s:=p:=m:=0 ;
i:=1
step 1
until 25
s:=s+a [i+25]↑3 ;
p:=p+w [i]↑2;
abs(b [i])
> abs(m)
l:=i;
m:=b [i]
p:=sqrt(p);
print(s,p,m,l)

Рис.I7

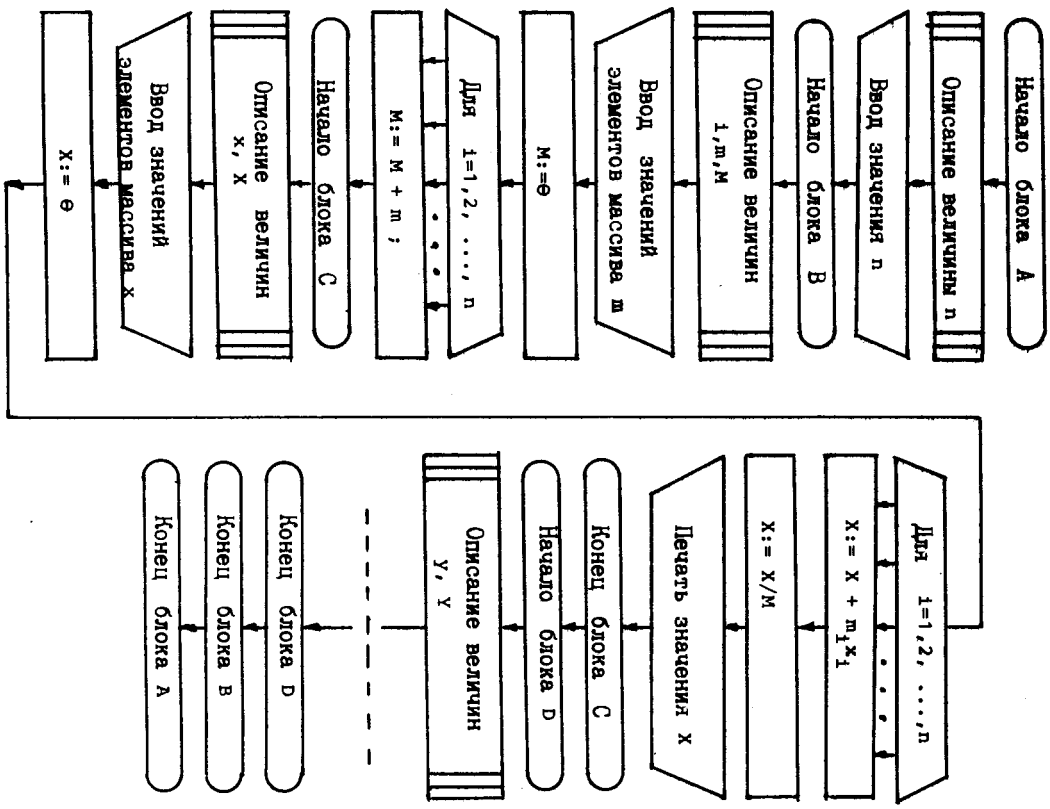

Начало блока A
Описание величины n
Ввод значения n
Начало блока B
Описание величин
i,m,M
Ввод значений
элементов массива m
M:=0
Для i=1,2, ...,n
M:= M + m ;
Начало блока C
Описание величин
x, X
Ввод значений
элементов массива x
X:= 0
Для i=1,2, ...,n
X:= X + m x
i i
X:= X/M
Печать значения X
Конец блока C
Начало блока D
Описание величин
y, Y
Конец блока D
Конец блока B
Конец блока A

Рис.I8

3) Рассмотрим блок-схему программы вычисления координат центра тяжести системы материальных точек, расположенных на плоскости

$$x_c = \sum_{i=1}^{n} m_i x_i \Big/ \sum_{i=1}^{n} m_i$$

$$y_c = \sum_{i=1}^{n} m_i y_i \Big/ \sum_{i=1}^{n} m_i$$

Здесь x_i, y_i - координаты точки;
m_i - масса точки.
(См. /4/ с.86).

Блок-схема программы решения примера 3 изображена на рис.I8.

Пояснительная информация об идентификаторах приведена в табл.3.

Таблица 3

Обозначение величины в алгоритме	Выбранный идентификатор	Класс величины	Тип значений, принимаемых величиной
n	n	Простая переменная	integer
i	i	То же	»
x_i	x	Массив	real
y_i	y	»	»
m_i	m	»	»
x_c	X	Простая переменная	»
y_c	Y	То же	»
$\sum_{i=1}^{n} m_i$	M	»	»

Двумерное изображение программы решения примера 3 показано на рис.I9.

Программа имеет вид:

```
A:begin integer n;
    read(n);
B: begin integer i; real M; array m [1:n];
    read(m);
    M:=0;
   for i:=1 step 1 until n do M:=M+m [i] ;
```

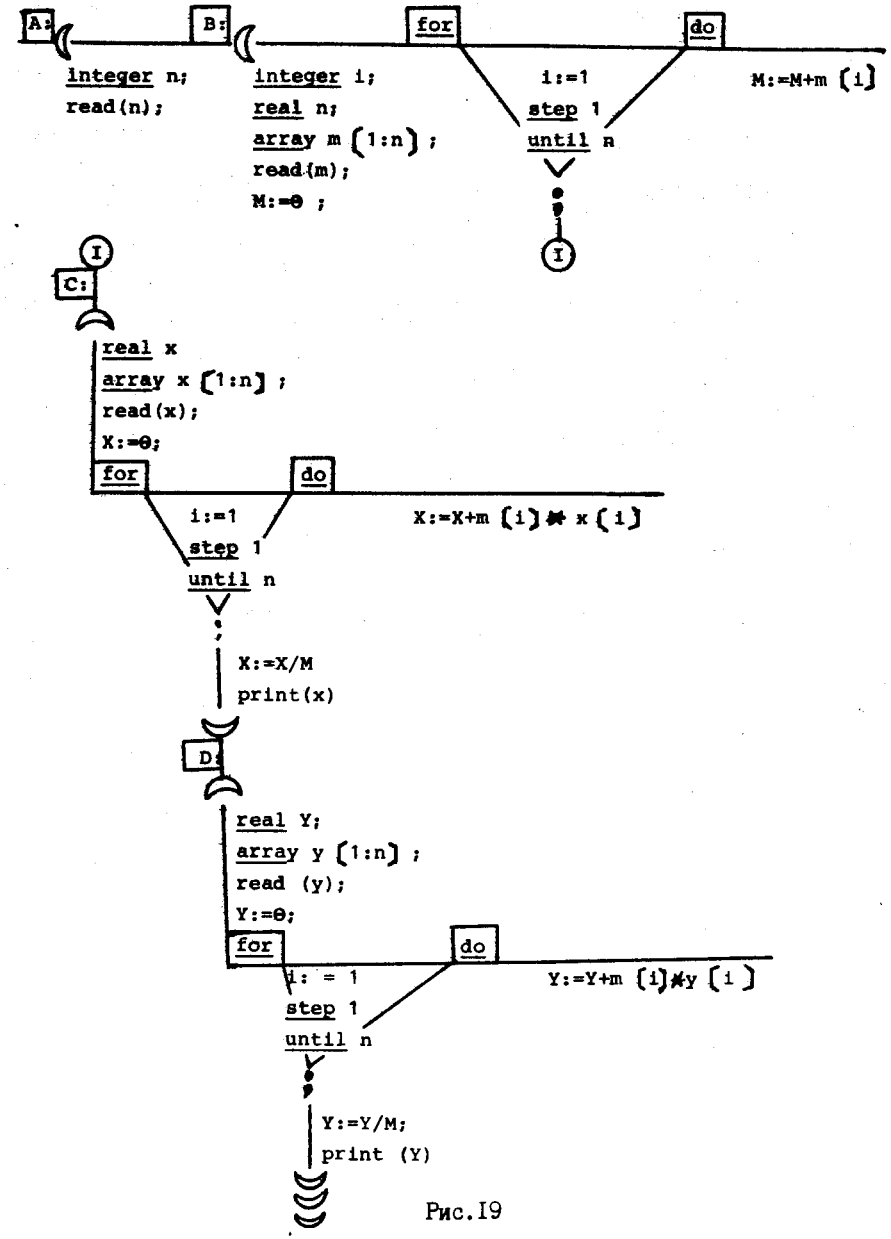



Рис.I9

```
C: begin real X; array x (1:n) ;
          read(x); X:=0;
          for i=1 step 1 until n do X:=X+m (i) * x (i) ;
          X:=X/M;
          print(X)
    end
D:begin real; array y (1:n) ;
        read (y); Y:=0;
        for i:=1 step 1 until n do Y:=Y+m (i) *y (i) ;
        Y:=Y/M;
        print(Y)
  end
 end
end
```

Приведенная программа оформлена в виде блока. В программе блок A является внешним, его подблоком является блок B . В свою очередь, подблоками блока B являются независимые друг от друга блоки C и D .

Заключение

Из приведенных примеров можно видеть, как преобразуется программа из одного вида в другой.

При этом такое преобразование программы может эффективно произвести любой программист, не умеющий писать программы на языке Алгол-60, а также любой лаборант, не умеющий программировать вообще.

Двумерное изображение программы ускоряет такие трудоемкие процедуры, как кодирование и процесс отладки программы.

Кроме того, двумерное изображение может служить средством документирования для завершенных программ.

В заключение автор выражает искреннюю благодарность А.А.Хошенко и Р.И. Гайдамаке за обсуждения и помощь в подготовке рукописи данной работы к печати.

Литература

1. Ли Рен Хи, Хошенко А.А. ОИЯИ,Р11-80-804, Дубна, 1980.

2. Вельбицкий И.В., Ходаковский В.Н., Шолмов Л.И. Технологический комплекс производства программ на машине ЕС ЭВМ БЭСМ-6, М., Статистика, 1980.

3. Ли Рен Хи. Двумерное программирование на языке ассемблер. ОИЯИ, 11-81-671, Дубна, 1981.

4. Савинков В.М., Цальп В.Д. Программирование на Алголе, М., Высшая школа, 1979.

Рукопись поступила в издательский отдел
27 октября 1981 года.