



ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
УПРАВЛЕНИЕ ДИСТАНЦИОННОГО ОБУЧЕНИЯ И ПОВЫШЕНИЯ
КВАЛИФИКАЦИИ

Кафедра «Вычислительные системы и информационная
безопасность»

Учебно-методическое пособие
«Программирование алгоритмов разветв-
ляющихся и циклических структур»
по дисциплине

«Информатика и ИКТ»

Авторы
Полуян А.Ю.,
Петренко С.Б.

Ростов-на-Дону, 2018

Аннотация

Учебно-методическое пособие предназначено для студентов всех форм обучения направлений 10.03.01 «Информационная безопасность», 13.03.02 «Электроэнергетика и электротехника», 10.05.02 «Информационная безопасность телекоммуникационных систем»

Методические указания предназначены для проведения лабораторных работ по дисциплине "Информатика и ИКТ". Содержит общие сведения программировании на языке Pascal ABC, позволяет освоить: основные приемы построения программ для решения различных задач. Лабораторная работа включает набор заданий, методические указания к ним и контрольные вопросы по изучаемой теме. Методические рекомендации могут быть использованы для самостоятельной работы.

Авторы

к.т.н., доцент кафедры «Вычислительные системы и информационная безопасность»

Полюян А.Ю.,

к.т.н., доцент кафедры «Вычислительные системы и информационная безопасность»

Петренко С.Б.



Оглавление

ЦЕЛЬ РАБОТЫ	4
ПОРЯДОК ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ	4
КРАТКИЕ СВЕДЕНИЯ ИЗ ТЕОРИИ	4
Программирование алгоритмов разветвляющейся структуры.....	4
Программирование алгоритмов циклической структуры ..	8
Программирование алгоритмов со структурой вложенных циклов.	12
КОНТРОЛЬНЫЕ ВОПРОСЫ.....	14
ИНДИВИДУАЛЬНЫЕ ЗАДАНИЯ.....	15
Список литературы	18

ЦЕЛЬ РАБОТЫ

Целью работы является ознакомление студентов с организацией программ разветвляющихся и циклических типов, а также закрепление практических навыков работы в программировании.

ПОРЯДОК ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ

Для успешного выполнения данной лабораторной работы студент должен:

- изучить теоретическую часть (раздел 3);
- согласно индивидуальному варианту составить схему алгоритма решения задачи;
- в соответствии со схемой составить программу решения задачи;
- выполнить решение задачи на персональной ЭВМ.

По окончании работы на ЭВМ необходимо должным образом оформить отчет, представить его преподавателю и защитить свою работу.

КРАТКИЕ СВЕДЕНИЯ ИЗ ТЕОРИИ

В большинстве встречающихся задач требуется использование определенного набора типовых приемов алгоритмизации, к которым можно отнести алгоритмы разветвляющейся и циклической структур.

В алгоритмах разветвляющейся структуры предусмотрено разветвление последовательности действия на несколько направлений в зависимости от итога проверки заданного условия.

В алгоритме циклической структуры предусмотрено неоднократное выполнение одной и той же последовательности действий.

Программирование алгоритмов разветвляющейся структуры

В языке ПАСКАЛЬ для программирования разветвляющейся структуры используются операторы безусловного перехода, условного перехода и оператор выбора.

Оператор безусловного перехода имеет вид `GOTO m`; где *m* –метка, представляющая собой целое число без знака. Метка должна быть описана в разделе описания. Этот оператор передает управления к оператору, описанному меткой *m*. Причем `mm` может быть помечен пустой оператор, который не предписывает

никаких действий и используется для выхода.

Условный оператор имеет следующие виды записи:

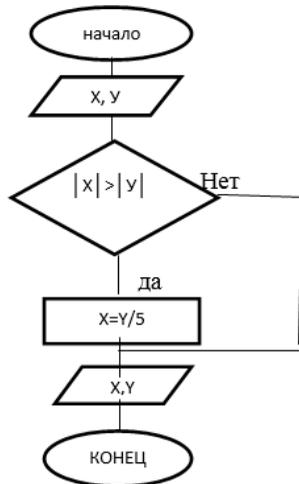
А) IF B THEN A;

Б) IF B THEN A ELSE C; где B - выражение логического типа; A, C- простые или сложные операторы.

При использовании условного оператора вида а) будет организовано вычисление оператора A, если логическое выражение B принимает значение TRUE(истина). Если же оно принимает значение FALSE (ложь), то будет выполняться оператор, который следует за этим условным оператором.

Пример 1. Пусть даны два число X и Y. Если первое больше второго по абсолютной величине, то необходимо уменьшить первое в 5 раз. Иначе оставить числа без изменения.

Блок-схема алгоритма и программа имеют вид:



PROGRAM A;

VAR X, Y: REAL ;

BEGIN

WRITELN (' ВВЕСТИ X,Y');

READ (X,Y);

IF ABS(X)> ABS(Y) THEN X:= X/5;

WRITELN (' X=',X,' ', 'Y='Y);

END.

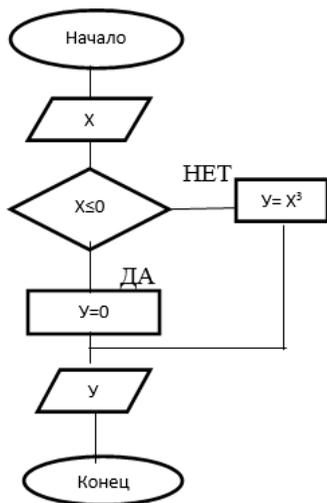
Рис.1

При использовании условного оператора вида б) будет организовано вычисление оператора A,если логическое выражение B принимает значение TRUE. Если логическое выражение принимает значение FALSE, то выполняется оператор C, стоящий после ELSE.

Пример 2. Вычислить Y по значениям X, если

$$y = \begin{cases} 0, & \text{если } X \leq 0; \\ X^3, & \text{если } X > 0; \end{cases}$$

Схема алгоритма и программа имеют вид:



PROGRAM B;

```

VAR X,Y :REAL;
BEGIN
  WRITELN ('ВВЕСТИХ');
  READLN(X);
  IF X:=0 THEN Y:=0 ELSE
  Y:=X*X*X;
  WRITELN('Y=',Y);
END.
    
```

Рис 2.

В случае, если необходимо выполнение операторов при тех или иных условиях, то это осуществляется объединением операторов в блок, формируемый операторами BEGIN и END

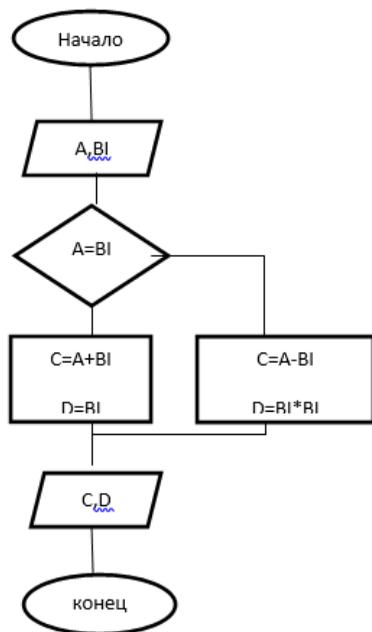
Например, IF B THEN BEGIN A1;A2;A3 END
ELSE BEGIN A4;A5; END;

В этом примере имена с A1 по A5 символически обозначают операторы. Если логическое выражение B принимает значение TRUE, то выполняются операторы A1,A2,A3, в противном случае выполняются операторы A4 и A5. Операторы, заключенные между BEGINи END, образуют так называемый составной оператор, а сами операторы BEGINи ENDчасто именуют операторными скобками.

Пример 3.

Требуется установить значения переменных C и D в зависимости от состояния переменных A и B1, причем, если значение переменной A равно B1, то переменной C присваивается значе-

ние, равное сумме значений переменных A и B1, а переменной D - значение переменной B1. В случае, если значение переменной A не равно значению переменной B1, то переменной C присваивается значение, равное разности значений переменных A и B1, а переменной D - значение переменной B1, возведенное в квадрат. Схема алгоритма и программа имеют следующий вид:



```

PROGRAMC(INPUT, OUTPUT);

LABEL IØ;

VAR A,B1,C,D:REAL;

BEGIN

  WRITELN( 'ВВЕСТИ A,B1' );
  READ( A, B1 );
  IF A=B1 THEN BEGIN
    C=A+B1;
    D=B1;
    GOTO IØ;
  ELSE BEGIN
    C=A-B1;
    D=B1*B1;
  END;

  IØ: WRITE( 'C=',C, ' ', 'D=',D );

END.
    
```

Оператор выбора варианта имеет вид:
 CASE C OF NI : A1;N2:A2;...NN:ANEND;

Где C-выражение скалярного типа (но не вещественного);
 NI,N2,...,NN –метка операторов; A1,A2,...,AN – операторы.

Выполнение оператора начинается с вычисления значения выражения C. Затем выполняется тот оператор, метка которого совпала с полученным значением. Если же вычисленное значение выражение C не совпало ни с одной из меток, то предусмотрено появление сообщения об ошибке.

Пример 4. Пусть требуется напечатать значение введен-

ной с клавиатуры цифры, принимающей значение от единицы до пяти, с контролем введенного значения.

Программа имеет вид:

```
PROGRAM VES;  
VAR I: INTEGER;  
BEGIN  
  WRITELN( 'ВВЕСТИ I');  
  READ(I);  
  WRITELN( 'I=',I);  
  CASE I OF  
    1: WRITELN( 'Вы ВВЕЛИ => 1');  
    2: WRITELN( 'Вы ВВЕЛИ =>2');  
    3: WRITELN( 'ВыВВЕЛИ =>3');  
    4,5: WRITELN( 'Вы ВВЕЛИ=>4 ИЛИ 5');  
  END; END.
```

Программирование алгоритмов циклической структуры

Программы циклической структуры могут быть организованы с помощью следующих операторов IF, FOR, WHILE, REPEAT. Условный оператор IF требует задания начального параметра цикла, проведения в цикле вычисления с этим параметром, изменения его на величину шага и проверки условия окончания цикла. Если цикл не закончен, то вычисления продолжаются.

Пример 5:

Составить схему алгоритма и программу для вычисления выражения. Схема алгоритма и программа имеют вид:

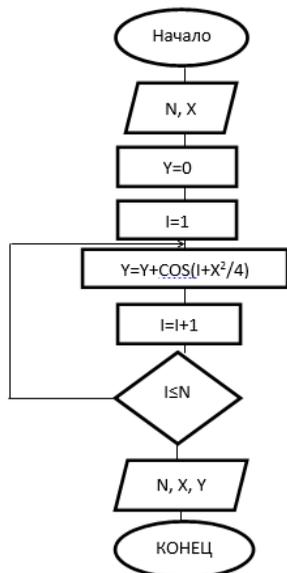


Рис. 4

```

PROGRAM SUMIP;

    LABEL 3;

    VAR X,Y: REAL;

        I,N:INTEGER;

    BEGIN

        WRITELN( 'ВВЕСТИ N,X');

        READ(N, X);

        Y:=0;

        I:=1;

    3: Y:=Y+COS(I+SQR(X)/4);

        I:=I+I;

        IF X<=N THEN GOTO 3;

        WRITELN( 'N=',N);

        WRITELN('X=',X, ' ',Y=',Y);

    END.
    
```

Оператор цикла FOR используется для организации цикла с известным числом повторений. При этом различают две формы записи:

А) с организацией счета при изменении параметра цикла от начального (меньшего) значения до конечного (большого) значения этого параметра цикла. Такой оператор цикла имеет вид:

```
FOR K:=NZ TO KZ DO
```

Б) с организацией счета при изменении параметра цикла от начального(большого) значения до конечного(меньшего) значения этого параметра. Этот оператор имеет вид:

```
FOR K: = NZ DOWNTON KZ DO
```

Где K- параметр цикла (целочисленная переменная), NZ, KZ-выражения, задающие соответственно начальное и конечное значения параметра цикла;
S-простой или составной оператор (тело цикла).

Оператор цикла выполняется следующим образом. Параметру цикла K присваивается начальное значение NZ. Затем уравнение передается в тело цикла и выполняется оператор S,

после выполнения которого параметр цикла меняет свое значение на единицу(шаг изменения параметра цикла). При этом шаг равен +I, если используется оператор цикла с ключевым словом TO(случай а), а если используется оператор цикла с ключевым словом DOWNTO (случай б), то шаг равен -I. Далее измененное значение параметра цикла сравнивается с конечным значением KZ и, если параметр цикла не превышает K(случай а) или превышает KZ (случай б), то управление передается в тело цикла и выполняется оператор S; в противном случае осуществляется выход из цикла.

Рассмотрим фрагменты записи операторов цикла:

```
VARY,I: INTEGER;
```

```
1) Y:=0; FOR I=3 TO 5 DO Y=Y+1;
```

```
2) Y:=0; FOR I=12 DOWNTO 6 DO Y:=Y+1;
```

В результате выполнения первого оператора цикла параметр цикла I будет изменяться от начального значения I=3 до конечного значения I=5 с шагом, равным +I, причем после окончания цикла значения переменной Y=3. При выполнении второго оператора цикла параметр цикла I будет изменяться от начального значения I=12 до конечного значения I=6 с шагом, равным -I. После окончания цикла Y=7.

Пример 6.

Используя условия примера 5 программу с изменением оператора цикла FOR.

```
PROGRAM SUM;
VAR X,Y: REAL;
I,N: INTEGER;
BEGIN
    WRITELN('ВВЕСТИ N,X ');
    READ(N, X);
    Y:=0;
    FOR I:=1 TO N DO
        Y:= Y+ COS (I+SQR(X)/4);
        WRITELN ('N=', N , ' ', 'X=',X);
        WRITELN('Y=',Y);
    END.
```

Оператор цикла WHILE позволяет организовать цикл, количество повторений которого зависит от включенного в него условия, т.е цикл с неизвестным числом повторений. Этот оператор имеет вид

```
WHILE LV DO
```

Где LV- логическое выражение;

S- простой или составной оператор. Выполнение оператора начинается с проверки выражения LV. Если логическое выражение имеет значение TRUE, то выполняется оператор S. Выполнение оператора S продолжается до тех пор, пока значение выражения LV не станет равным FALSE. В этом случае уравнение передается оператору, следующему за оператором S. Если же выражение LV имеет значение FALSE при первоначальной проверке, то оператор S не выполняется ни разу. При этом, чтобы выйти из цикла, внутри его должно быть организовано изменение значений переменных, входящих в логическое выражение. Иначе оператор цикла будет выполняться бесконечное число раз.

Пример 7.

Составим программу, используя условия примера 5. Программа запишется в виде:

```

PROGRAM SUMWHL;
  VAR X,Y : REAL;
      I,N : INTEGER;
  BEGIN
    WRITELN('ВВЕСТИ N,X');
    READ(N,X);
    Y:=0;
    I:=1;
    WHILE I<=N DO
      BEGIN
        Y=Y+ COS( I+SQR(X)/4);
        I:=I+1;
      END;
    WRITELN( 'N=',N,'  ','X=',X);
    WRITELN( 'Y=',Y);
  END.
    
```

Оператор цикла REPEAT также позволяет организовать цикл с неизвестным числом повторений. Такой оператор имеет вид:

```

REPEATS
UNTILLV
    
```

где S – простой или составной оператор;
LV- логическое выражение.

Выполнение оператора REPEAT начинается с вычисления оператора S и продолжается до тех пор, пока не выполняется LV т.е. когда значение этого логического выражения равно FALSE. Это означает, что проверка LV проводится после каждой итера-

ции и в случае принятия LV значения TRUE осуществляется выход из цикла. При использовании составного оператора S операторные скобки (BEGIN и END) не требуются. Кроме того оператор, стоящий перед словом UNTIL, не имеет после себя точку с запятой.

Пример 8.

Задан массив вещественных чисел, состоящий из десяти элементов. Требуется вывести четные элементы этого массива. Программа имеет вид:

```
PROGRAM MAS;
VAR I: INTEGER;
    A: ARRAY[1...10] OF REAL;
BEGIN
    WRITELN(' ВВЕСТИМАССИВ(1..10)');
    FOR I:=1 TO 10 DO;
        READ (A[I]);
        I:=0;
        REPEAT
            I=I+2;
            WRITE (A[I]:5:2)
        UNTIL I>10;
    END.
```

Программирование алгоритмов со структурой вложенных циклов.

Для реализации сложных алгоритмов возникает необходимость организовать в составе одного цикла еще несколько других циклов. При этом цикл, внутри которого находятся другие циклы, называется внешним, а вложенные в него - внутренними. Алгоритмы циклических вычислительных процессов, содержащие внутренние циклы, будем называть алгоритмами со структурой вложенных циклов.

Организовывать вложенные циклы необходимо таким образом, чтобы циклы не пересекались, т.е., чтобы тело внутреннего цикла полностью размещалось в теле внешнего цикла.

Изменение параметров внутреннего и внешнего циклов происходит не одновременно, следующим образом:

- при начальном значении параметра внешнего цикла параметра внутреннего цикла последовательно принимает все возможные значения (от своего начального значения до конечного), причем передача управления из внутреннего во внешний цикл происходит после принятия параметром внутреннего цикла своего конечного значения и выполнения тела этого цикла;

- - происходит изменение параметра внешнего цикла на величину шага, а параметр внутреннего цикла опять последовательно принимает все свои значения;

--выход из внешнего цикла производят тогда, когда параметр этого цикла примет свое конечное значение, а параметр внутреннего цикла последовательно примет все свои значения.

Пример 8.

Составить программу для вычисления

$$P = \sum_{n=1}^m \sqrt{2 * S / n} * \prod_{i=1}^n N / \sqrt{i}$$

```

PROGRAM VLZ;
VAR  N,M, I: INTEGER;
      S,R,P,Y,Z: REAL;
BEGIN
    WRITELN ('ВВЕСТИ M,S');
    READ (M,S);
    R:=0;
    FOR N:=1 TO M DO
    BEGIN
        P:=1;
        FOR I:=1 TO N DO
        BEGIN
            Y:=N/SQRT(I);
            P:=P*Y;
            Z:=SQRT(2*S/N)*P;
            R:=R+Z;
        END;
        WRITELN( 'R=',R);
    END.

```

Z=V25/n *P

R=R+Z

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Опишите оператор безусловного перехода.
2. Опишите конструкцию условного оператора.
3. Для какой цели используются операторные скобки?
4. Опишите оператор выбора варианта.
5. С использованием каких операторов могут быть организованы циклические структуры?
6. В каких случаях используется оператор цикла со служебным словом DOWNTO?
7. Каких этапов требует организация цикла с помощью условного оператора?
8. С помощью каких операторов

можно организовать цикл с неизвестным числом повторений?

9. Что означают служебные слова TRUE и FALSE?

10. Как изменяются параметры внешнего и внутреннего циклов?

ИНДИВИДУАЛЬНЫЕ ЗАДАНИЯ

1. вычислить $y = \sqrt{a^2 + b^2}$ где $a = \begin{cases} \operatorname{tg} x, & \text{если } x > 2 \\ x^2, & \text{если } 0 \leq x \leq 2 \\ 4, & \text{если } -1 \leq x < 0 \\ \sqrt{\sin x}, & \text{в остальных случаях} \end{cases}$

2. вычислить $y = \frac{a\sqrt{b} + b\sqrt{a}}{2}$ где $a = \begin{cases} e^x, & \text{если } b > 0 \\ 0, & \text{если } b = 0 \\ \sin x, & \text{в остальных случаях} \end{cases}$

3. вычислить $y = ax^2 + b$, где $a = \begin{cases} e^x, & \text{если } x < b \\ \sin x, & \text{если } x = b \\ \log_{\sqrt{b}}(\sin x), & \text{если } x > b \end{cases}$

4. вычислить $y = \begin{cases} 1 - e^{ax} \sin(ax + b), & \text{при } x > \pi \\ 1 - e^{-ax}(ax + b), & \text{при } -\pi \leq x \leq \pi \\ 1 - (e^{-ax} + e^{-bx}), & \text{при } x < -\pi \end{cases}$

где $x = ab^2 - \sin b$

5. вычислить $y = \begin{cases} \frac{1}{\sqrt{d}} \operatorname{arctg} \frac{b+c}{\sqrt{d}}, & \text{при } d > 0 \\ -\frac{1}{b+c}, & \text{при } d = 0 \\ \frac{1}{2\sqrt{d}} \ln \sqrt{d}, & \text{при } d > 5 \end{cases}$ где $d = ac - b^2$
6. вычислить $n = \begin{cases} 1, & \text{если } x \geq 0 \text{ и } y \geq 0 \\ 2, & \text{если } x < 0 \text{ и } y \geq 0 \\ 0, & \text{в остальных случаях} \end{cases}$ где $y = \sqrt[3]{\operatorname{tg} x^2 + \sqrt{\sin x + \cos x}}$
7. вычислить $y = \frac{1}{x} + \frac{\sqrt{e^x + e^z}}{x}$ где $z = \begin{cases} x^3, & \text{если } x > b \\ 0, & \text{если } x < b \\ 7x, & \text{если } x = b \end{cases}$
8. вычислить $y = \frac{\sqrt{x + e^{5 \sin x^2}}}{z^x + x^z}$ где $z = \begin{cases} \frac{\sin x}{x}, & \text{если } x \neq 0 \\ 1, & \text{если } x = 0 \end{cases}$
9. вычислить $y = \begin{cases} \frac{\pi}{2}, & \text{если } \alpha = \sin x \text{ и } \beta > 0 \\ 1, & \text{если } \alpha = \cos x \text{ и } \beta > 0 \\ -\pi, & \text{в остальных случаях} \end{cases}$
10. Вычислить: $\sum_{i=1}^{10} \frac{1}{i!}$
11. Вычислить: $\sum_{i=1}^{50} \frac{1}{i^3}$
12. Вычислить: $\sum \frac{1}{(2i)^2}$
13. Дано натуральное n . Вычислить: $\sum_{k=1}^n \frac{1}{K}$
14. Дано натуральное n . Вычислить: $\sum_{k=1}^n \frac{1}{K^5}$

Информатика и ИКТ

15. Дано натуральное n . Вычислить:
$$\sum_{k=1}^n \frac{1}{(2k+1)^2}$$

16. Дано натуральное n . Вычислить:
$$\sum_{k=1}^n \frac{(-1)^k}{(2k+1)k}$$

17. Дано натуральное n . Вычислить:
$$\sum_{k=1}^n \frac{(-1)^k (k+1)}{k!}$$

18. Дано натуральное n . Вычислить:
$$\sum_{k=0}^n \frac{(-1)^k (k+1)}{k!}$$

19. Задан одномерный массив из n элементов. Найти наименьший элемент в массиве.

20. Задан одномерный массив из n элементов. Найти количество положительных элементов массива.

21. Задан одномерный массив из n элементов. Найти количество отрицательных элементов массива.

22. Задан одномерный массив из n элементов. Определить, сколько раз встречается число 7 среди элементов массива.

23. Задан одномерный массив из n элементов. Определить, сколько элементов массива меньше, чем число 6.

24. Задан одномерный массив из n элементов. Определить, сколько элементов массива больше, чем число 3.

25. Задан одномерный массив из n элементов. Найти сумму всех неотрицательных элементов массива.

26. Дан ряд чисел. Найти наименьшее из чисел $a_1, 2a_2, 3a_3, \dots, 8a_8$.

27. Дан ряд чисел. Найти наибольшее из чисел $2a_1, 3a_2, 4a_3, \dots, 9a_8$.

28. Найти сумму ряда чисел $a_1+2a_2+3a_3+\dots+8a_8$.

29. Задан одномерный массив. Найти наименьший по модулю элемент массива.

30. Задан двухмерный массив. Найти произведение элементов третьего столбца матрицы.

31. Задан двухмерный массив. Найти сумму элементов первой и третьей строк матрицы.

32. Задан двухмерный массив. Расположить все элементы матрицы в строку в порядке убывания.

33. Задан двухмерный массив. Определить, сколько раз

встречается число 1 среди элементов третьей строки матрицы.

34. Задан двумерный массив. Найти количество отрицательных элементов в первой строке матрицы.

СПИСОК ЛИТЕРАТУРЫ

1. А.Ю. Полуян, С.Б. Петренкова и др. Основы информатики, Ростов н/Д. НПИ, 2016г.

2. С.В. Симонович. Информатика. Базовый курс. — Питер, 2010г.