Одной из основных алгоритмических структур является ветвление (альтернатива).

Если условие выполняется, то будет выполнена инструкция "1", если нет, то - инструкция "2". Несмотря на то, что в схеме присутствуют два действия, выполнено будет только одно, так как условие либо ложно, либо истинно. Третьего не дано. Такая схема позволяет решать задачи, в которых в зависимости от сложившихся обстоятельств требуется совершить то или иное действие. Нет никакого сомнения, что число задач такого рода огромно. Более того, очень сложно придумать реально значимое задание, алгоритм выполнения которого содержал бы в себе простое прямое следование команд. Даже примитивный пример, взятый из курса математики, как вы увидите, не может быть решен без использования ветвления. Итак, необходимо вычислить значение выражения y=1/x. Вам известно, что данная функция не всегда имеет значение, то есть не для всех значений аргумента существует значение результата. Наша задача так составить алгоритм, чтобы исполнитель ни в коем случае не встал в тупик, даже при получении нуля в качестве аргумента. Сформулировать это на естественном языке не трудно:

1. Получить значение x.

2. Если x=0, то сообщить, что выражение значения не имеет, иначе - вычислить y как 1/x.

Таким образом используется приведенная выше алгоритмическая структура. Она может быть выражена простыми словами:

Если <усл.> {Если выполняется условие}

то <действие 1> {то выполнить действие № 1 }

иначе <действие 2> {иначе - выполнить действие № 2 }

все

Как это записать на Паскале? Да точно так же, только по-английски.

Формат условного оператора на языке Паскаль:

If <условие>

Then <оператор 1>

Else <оператор 2>;

Обратите внимание на то, что в Then- и Else- части стоит только один оператор. Но что делать, чтобы решить задачу, в которой по выполнению или невыполнению условия нужно совершить не одно, а несколько действий? Здесь приходит на помощь уже известный вам составной оператор. В операторные скобки можно заключить любое количество операторов.

Вариант условного оператора в этом случае:

If <условие>

Then Begin <группа операторов 1> end

Else Begin < группа операторов 2> end;

Знак "точка с запятой" не ставится перед служебным словом Else, но операторы в группах, естественно, отделяются друг от друга этим знаком.

Теперь поговорим об условиях. В программах на языке Паскаль условия представляют собой выражения, значением которых является величина логического (Boolean) типа. Это может быть как просто переменная указанного типа, так и сложная последовательность высказываний, связанных логическими операциями.

В простых условиях могут применяться знаки операций сравнения: >(больше), <(меньше), =(равно), <>(не равно), >=(больше или равно), <=(меньше или равно).

Примеры простых условий:

A=5 {Значение переменной А равно 5}

(C+D3)>=(D1\*(45-2)) {Значение выражения в левой части больше либо равно значению выражения из правой части}

S<>'ABC' {Значение переменной S не равно строковой константе 'ABC'}

Приведем пример решения еще одной задачи: "Из двух чисел выбрать наибольшее".

На первый взгляд решение очевидно, но оно не столь тривиально, как кажется.

Program Example;

Var A,B,C: Real; {A,B - для хранения аргументов, C - результат}

Begin

Writeln('Введите два числа');

Readln(A,B); {Вводим аргументы с клавиатуры}

If A>B Then C:=A Else C:=B; {Если A>B, то результат - A, иначе результат - B}

Writeln(C); {Выводим результат на экран}

End.

Еще один классический пример: "По заданным коэффициентам решить квадратное уравнение". Эта задача сложнее, поэтому перед тем как писать программу составим алгоритм, записав его в виде блок-схемы.Сначала вводим коэффициенты, затем вычисляем дискриминант. Теперь возникает две возможности: либо отсутствие действительных корней в случае отрицательного дискриминанта, либо эти корни можно все-таки вычислить и вывести на экран в случае неотрицательного дискриминанта (случай равенства дискриминанта нулю входит сюда же, корней - два, только они одинаковые J).

При записи алгоритма на языке программирования следует учесть, что в ветви "нет" не одно действие, а три, поэтому следует применить составной оператор. Арифметические выражения не забывайте записывать в соответствии с правилами языка Паскаль. В остальном, эта программа не сложнее предыдущей.

Program Sq1;

Var A, B, C, D, X1, X2: Real;

Begin

Writeln ('Введите коэффициенты квадратного уравнения');

Readln (A,B,C);

D:=B\*B-4\*A\*C;

If D<0 Then Writeln ('Корней нет! ')

Else

Begin

X1:=(-B+SQRT(D))/2/A;

X2:=(-B-SQRT(D))/2/A;

Writeln ('X1=', X1:8:3, ' X2=',X2:8:3)

End

End.

Интересно, что в качестве оператора, который выполняется по выполнению или невыполнению условия, может выступать условный же оператор. В этом случае говорят о вложенности условных операторов. Я настоятельно рекомендую при решении такого рода задач составлять блок-схему алгоритма в тетради. Только потом, при составлении программы, вам остается лишь аккуратно прописывать сначала всю Then- часть, а затем переходить к Else- части. Обычно при записи условных операторов на языке Паскаль (особенно при множественных ветвлениях) команды записывают уступом вправо и вниз. Это повышает наглядность, и, поверьте, снижает потери времени на отладку.

Для иллюстрации решим еще одну задачу: "решить уравнение вида A\*x^2 + B\*x + C = 0". Прошу не путать с квадратным уравнением, для которого нам было известно, что коэффициент А не равен нулю. Здесь же коэффициенты могут быть любыми числами. Исходя из элементарных математических рассуждений, получаем следующий алгоритм:

Program Sq2;

Var A, B, C, D, X, X1, X2: Real;

Begin

Writeln ('Введите коэффициенты уравнения (A, B, C) ');

If A=0 Then

If B=0 Then

If C=0 Then Writeln('X - любое число')

Else Writeln('Корней нет! ')

Else Begin X:=-C/B; Writeln('X=',X:8:3) End

Else

Begin

D:=B\*B-4\*A\*C;

If D<0 Then Writeln ('Корней нет! ')

Else

Begin

X1:=(-B+SQRT(D))/2/A;

X2:=(-B-SQRT(D))/2/A;

Writeln ('X1=', X1:8:3, ' X2=',X2:8:3)

End

End

End.

Выбор вариантов

Существуют условия, которые имеют много вариантов, в зависимости от которых необходимо выполнить то или иное действие, например: при вводе месяца необходимо вывести сезон, которому соответствует данный месяц. В паскале есть специальный оператор выбора case, который оформляется следующим образом:

case mesiac of

'Декабрь','Январь','Февраль': writeln('Зима');

'Март','Апрель','Май': writeln('Весна');

'Июнь','Июль','Август': writeln('Лето');

'Сентябрь','Октябрь','Ноябрь': writeln('Осень');

else wrteln('Такого месяца не существует!');

end;

Где в операторе перечисляются все варианты для которых выполняются те или иные действия, оператор else выполняет действия если операнд имеет значение, которое не перечислено в case.

Сложные условия (связки)

Повествовательное предложение, о котором можно сказать верно оно или нет называют высказыванием. Например:

Сегодня вторник.

Идёт дождь.

Небо может быть голубым или серым.

Число может быть чётным.

Завтра будет не среда.

Высказывания могут объединяться с помощью слов (операций) И, Или, НЕ. Эти операции называют логическими операциями. Объединённые высказывания называют связками. Например:

Сегодня вторник и идёт дождь.

Число может быть чётным или нечётным.

Завтра будет не вторник.

Логическую операцию И(AND) обозначают ^, операцию ИЛИ(OR) V, отрицание НЕ(NOT) ---.

Верность или неверность связки двух высказываний (истинность) определяют по правилам истинности.



Например: Определить истинно или ложно следующее высказывание:

not ((4>8) and (5<7))

|  |  |
| --- | --- |
| 4>8 | ложно |
| 5<7 | истинно |
| (4>8) and (5<7) | ложно |
| not ((4>8) and (5<7)) | истинно |

Итак высказывание истинно.

ОРГАНИЗАЦИЯ ВЕТВЛЕНИЙ В ПРОГРАММАХ НА ПАСКАЛЕ. ОПЕРАТОР IF THEN... ELSE

Формат оператора:

IF условие THEN оператор 1 ELSE оператор 2;

Правило работы:

1. Проверяется условие.

2. Если условие истинно, то выполняется оператор1

3. Если условие ложно, то выполняется оператор2.

Допустим сокращенный формат записи оператора:

IF условие THEN оператор 1;

Правило работы:

1. Проверяется условие.

2. Если условие истинно, то выполняется оператор1.

3. Если условие ложно, то выполняется оператор следующий после IF.

Под условием понимается логическое (булевское) выражение.

ЛОГИЧЕСКОЕ ВЫРАЖЕНИЕ - это выражение, которое принимает значение TRUE(истина) или FALSE(ложь), и включает в себя арифметические выражения, связанные операциями отношения и логическими операциями.

ОПЕРАЦИИ ОТНОШЕНИЯ:

> больше

>= больше или равно

< меньше

<= меньше или равно

= равно

<> неравно

Имеют один и тот же приоритет и соединяют два выражения одного и того же типа. Результатом любой операции сравнения является значение TRUE или FALSE.

ЛОГИЧЕСКИЕ ОПЕРАЦИИ:: NOT - НЕ; AND - И; OR - ИЛИ

ПОРЯДОК ВЫПОЛНЕНИЯ ОПЕРАЦИЙ:

1. - (унарный минус, т.е. отрицание числа) NOT

2. \* / DIV MOD AND

3. + - OR

4. < <= > >= = <>

Оператор ветвления (условный оператор)

Формат оператора:

If условие THEN

оператор1

ELSE оператор2;

где условие - логическое выражение (т.е. в него обязательно входит знак отношения, и можно определить, истинно выражение или ложно)

оператор1 - любой оператор Паскаля; этот оператор исполняется, когда условие истинно;

оператор2 - любой оператор Паскаля, в том числе и условный; этот оператор исполняется, когда условие ложно.

Выполнение оператора:

При выполнении этого оператора сначала вычисляется значение условия.

Если условие истинно (true), то выполняется оператор после слова THEN, а оператор2 пропускается;

если условие ложно (false), то оператор1 пропускается, а выполняется оператор после слова ELSE.

Примеры:

if x > 0 then

y:=sqrt(x

else write ('Решений нет');

Операторы 1 и 2 могут быть любыми, в том числе и условными. В этом случае их называют вложенными.

Примеры:

if x > 0 then

y:=sqrt(x

else

if x <= -3 then

y:=sqr(x)

else write ('Решений нет');

Краткая форма условного оператора

Части ELSE оператор2 может и не быть, тогда формат оператора:

If условие THEN оператор1;

Выполнение оператора:

Если условие истинно, то выполняется оператор после слова THEN, а иначе он пропускается.

Примеры:

if abs(x) >= 5 then

y:=exp(3\*ln(x))

Составной оператор

В условном операторе после слов THEN и ELSE можно помещать по несколько операторов. В этом случае их надо объединить в СОСТАВНОЙ оператор, который начинается словом BEGIN и заканчивается словом END; (на конце не точка, а точка с запятой!)

Примеры

Составить программу, которая перераспределит заданные значения x, y так, что в х окажется большее значение, а в y - меньшее.

program PRMAX;

var x, y, z: real;

begin

write ('Введите два числа: ');

read (x, y);

if x < y then

begin z:=x;

x:=y;

y:=z

end;

writeln ('x =', x, 'y=', y);

end.

Сложное условие

После слова IF допускается:

конъюнкция условий (условия связываются при помощи логической операции and),

дизъюнкция условий (условия связываются при помощи логической операции or);

инверсия условий (отрицание условия not)

Примеры:

Выяснить, принадлежит ли точка с координатами (x, y) кольцу с центром в точке (0, 0), внешним радиусом r1 и внутренним радиусом r2

program PRcir;

var x, y, r1, r2: real;

begin

write ('Введите координаты точки: ');

read (x, y);

writeln ('Введите радиусы r1, r2:: ');

read (r1, r2);

if (sqrt(sqr(x)+sqr(y))<r1) and (sqrt(sqr(x)+sqr(y))>r2) then

writeln ('Точка принадлежит кольцу');

else writeln ('Точка не принадлежит кольцу');

end.