

ПРОГРАММИРОВАНИЕ РАЗВЕТВЛЯЮЩИХСЯ АЛГОРИТМОВ.

Структурные операторы строятся из других операторов по определенным правилам. Операторы, входящие в структурный оператор, выполняются последовательно – в составных операторах и операторах над записями, альтернативно – в условных операторах, многократно – в операторах цикла.

При формировании структурных операторов существуют некоторые ограничения на число входящих в него операторов. В частности, в операторе выбора IF (в школьном алгоритмическом языке команда «если») после служебного слова THEN (аналог – «то») может стоять только один оператор. Поэтому в Паскале возникла необходимость группирования операторов в единое целое – в один составной оператор.

Любая группа операторов, размещенных между словами BEGIN и END (иначе, операторные скобки), рассматривается как один – составной оператор. При выполнении составного оператора все его компоненты (операторы) выполняются в порядке их написания (линейно).

Составные операторы обрабатываются как единое целое, как один оператор, что имеет значение там, где синтаксис языка допускает использование только одного оператора. Это имеет место практически во всех структурных операторах.

Организация ветвлений. Операторы выбора

В языке Паскаль алгоритмическая базовая конструкция выбора может быть реализована с помощью двух структурных операторов – IF и CASE, называемых *операторами выбора*. С их помощью можно выбрать для выполнения один из составных операторов (или ни одного оператора).

Оператор ветвления IF

Оператор IF можно представить в общей форме записи как

IF <Условие> THEN <Оператор 1> ELSE <Оператор 2>,

где конструкция «Условие» есть логическое выражение, которое принимает два значения типа BOOLEAN: TRUE, FALSE (истинно или ложно).

Само логическое выражение складывается из операций сравнения: >, >=, <, <=, =, <>. Результат сравнения может быть TRUE или FALSE.

Логические выражения могут формироваться также и с помощью трех логических операций: NOT, AND, OR. Приоритеты всех используемых в Паскале операций таковы:

Высший: ()
NOT *, /, DIV, MOD
AND
OR +, -

Низший: >, =, <, >=, <>, <=

В качестве условия может быть использована и логическая переменная, например:

I and J or K ---> (I and J) or K;
not X and Y ---> (not X) and Y, где I, J, K, X, Y – переменные типа BOOLEAN;
(A < B) or (B = 0), где A, B – переменные простого типа.

В операторе IF всегда за словами THEN и ELSE должен следовать один оператор. Этим оператором может быть не только оператор присваивания, но и любой другой. Если хотя бы один из них является снова оператором IF, то полученную конструкцию называют **в л о ж е н и е м**.

П р и м е р :

```
IF <условие1> THEN <ветвь 1>  
    ELSE IF <условие2> THEN <ветвь 2>  
        ELSE <ветвь 3>;
```

Такое вложение используется для уменьшения числа необходимых проверок. Этот метод часто обеспечивает большую эффективность, чем составное условие, однако одновременно он уменьшает надежность программы. Не рекомендуется использовать более двух-трех уровней вложения IF. Вложения могут идти и после слова THEN.

Первый способ предпочтительнее, чем второй, так как конструкция THEN-IF менее удобна, чем ELSE-IF. С помощью конструкции ELSE-IF чаще всего осуществляется выбор одного из нескольких альтернативных вариантов. Заметим, однако, что иногда такое вложение можно заменить на последовательность операторов короткой формы IF-THEN. Это видно на следующем примере:

```
program QUARD;  
  var A, B, C : real; D : real;  
  begin  
    read (A, B, C); D := sqr (B) - 4 * A - C;
```

1-й вариант	2-й вариант
if D < 0 then write ('Не имеет корней');	if D < 0 then write ('Нет корней')
if D = 0 then write ('Один корень');	else if D = 0 then write ('Один корень')
if D > 0 then write ('Два корня');	else write ('Два корня');

end.

Однако в данном примере 2-й вариант более эффективен, так как имеет на одно сравнение меньше, и в случае $D < 0$ сразу же дает ответ, не делая последующих проверок.

В Паскале существует специальный оператор, позволяющий сделать выбор из нескольких альтернатив – множественное ветвление, или оператор варианта.

Оператор варианта состоит из выражения и списка операторов, каждому из которых предшествует одна или более констант, называемых *константами выбора*.

<p style="text-align: center;">Общая форма записи CASE <выражение> OF константы: оператор; константы: оператор ELSE <оператор> END;</p>
--

Выражение, стоящее между CASE и OF, называется *селектором*. Константы (значения выражения), предшествующие двоеточию, называются *метками случаев*. Порядок работы оператора: сначала вычисляется значение селектора, затем выполняется оператор, одна из меток которого совпадает со значением селектора. Все остальные операторы не выполняются, и управление передается следующему после END оператору. В случае короткой формы оператора CASE при несовпадении значения селектора (ключа) ни с одной из констант из списка никакой оператор не подлежит исполнению. Если же в операторе есть строка ELSE, то при несовпадении значения селектора ни с одной константой выполняется оператор, следующий за ELSE.

Выражение «селектор» может относиться к любому скалярному типу, кроме REAL. Метки случаев должны принадлежать тому же типу, что и селектор. Нежелательно, чтобы одна и та же метка появлялась более одного раза в операторе CASE. Если же это произойдет (компилятор не проверяет повторяемость меток), то выполнится тот оператор, который соответствует первому вхождению метки в список констант.

Оператор CASE особенно удобно использовать, во-первых, когда характер действий определяется значениями, которые не попадают в последовательно расположенные интервалы, во-вторых, когда нескольким дискретным значениям селектора соответствует одно и то же действие.

Оператор перехода

Оператор перехода имеет вид

GOTO <метка>.

Он позволяет передать управление непосредственно на нужный оператор программы. Перед этим оператором должна располагаться метка отделенная

от него двоеточием. В Турбо Паскале в качестве меток выступают либо целые числа от 0 до 9999, либо идентификаторы. Все метки должны быть описаны в разделе объявления меток следующим образом:

label <список меток через запятую> ;

П р и м е р 1. Печать названия десятичных цифр.

```
program DICITS;
var DIGIT: integer;
begin
writeln ('Введите цифру');
readln (DIGIT);
case DIGIT of
  0: writeln ('нуль');
  1: writeln ('один');
  .....
  9: writeln ('девять');
  else writeln ('это не цифра');
end;
end.
```

П р и м е р 2. Печать номера квартала года.

```
program NUMKVART;
var MESIATZ: 1..12;
begin
write ('Введите номер месяца года – ');
read (MESIATZ);
case MESIATZ of
  1, 2, 3: writeln ('Первый квартал');
  4, 5, 6: writeln ('Второй квартал');
  7, 8, 9: writeln ('Третий квартал');
  10, 11, 12: writeln ('Четвертый квартал');
end;
end.
```

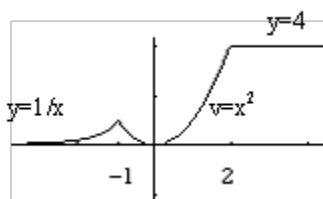
Примечание. В операторе CASE формально нет условий как таковых, однако проверка условий осуществляется в неявном виде на предмет совпадения константы со значением селектора.

Вопросы для самопроверки:

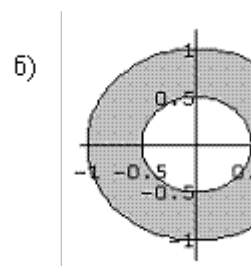
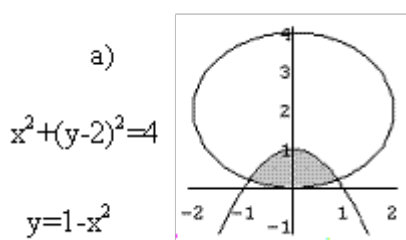
1. Какие операторы используются для программирования развилок?
2. Как выполняются операторы условного перехода?
3. Какую из функций: $\text{Sin}(x)$, $\text{Abs}(x)$, $\text{Trunc}(x)$ можно заменить условным оператором $\text{if } x < 0 \text{ then } x := -x$?
4. Если выбор вариантов осуществляется из конечного числа элементов выбора, то лучше взять для этого оператор if или case ?
5. Как заменить оператор case операторами if ?
6. В чем преимущество оператора case от последовательности «коротких» операторов if ?
7. Какой тип переменной можно использовать в качестве ключа оператора case ?

Задания

1. Даны действительные положительные числа x , y , z . Выяснить, существует ли треугольник с длинами сторон x , y , z .
2. Дано действительное a . Для функции $f(a)$, график которой представлен на рисунке, вычислить $f(a)$.



3. Пусть D - заштрихованная часть плоскости и пусть u определяется по x и y следующим образом (запись $(x, y) \in D$ означает, что точка с координатами x , y принадлежит D):



а)
$$u = \begin{cases} -3, & \text{если } (x, y) \in D \\ y^2 & \text{в противном случае} \end{cases}$$

б)
$$u = \begin{cases} x - y, & \text{если } (x, y) \in D \\ xy + 7 & \text{в противном случае} \end{cases}$$

4. Даны три действительных числа. Выбрать из них те, которые принадлежат интервалу $(1, 3)$.
5. Даны действительные числа x , y . Если x , y отрицательны, то каждое значение заменить его модулем; если отрицательное только одно из них, то

оба значения увеличить на 0.5; если оба значения не отрицательны и ни одно из них не принадлежит отрезку $[0.5, 2.0]$, то оба значения уменьшить в 10 раз; в остальных случаях x , y оставить без изменения.

6. Определить и вывести на печать номер квадранта, в котором расположена точка $M(x,y)$, x и y заданные вещественные числа.

7. Из величин, определяемых выражениями $a=\sin x$, $b=\cos x$, $c=\ln|x|$ при заданном x , определить и вывести на экран дисплея минимальное значение.

8. Определить, какая из двух точек - $M_1(x_1,y_1)$ или $M_2(x_2,y_2)$ - расположена ближе к началу координат. Вывести на экран дисплея координаты этой точки.

9. Определить, какая из двух фигур (круг или квадрат) имеет большую площадь. Известно, что сторона квадрата равна a , радиус круга r . Вывести на экран название и значение площади большей фигуры.

10. Определить, попадает ли точка $M(x,y)$ в круг радиусом r с центром в точке (x_0,y_0) .

11. Перераспределить значения переменных X и Y так, чтобы в X оказалось меньшее из этих значений, а в Y — большее.

12. Значения переменных X , Y , Z поменять местами так, чтобы они оказались упорядоченными по возрастанию.

13. Известны два расстояния: одно в километрах, другое в футах. Какое из расстояний меньше?

14. Ввести два числа. Меньшее заменить полусуммой, а большее — удвоенным произведением.

15. Локатор ориентирован на одну из сторон света («С» — север, «З» — запад, «Ю» — юг, «В» — восток) и может принимать три цифровые команды: 1 — поворот налево, -1 — поворот направо, 2 — поворот на 180 градусов. Дан символ S — исходная ориентация локатора и числа N_1 и N_2 — две посланные ему команды. Вывести ориентацию локатора после выполнения данных команд.

16. Даны два целых числа: D (день) и M (месяц), определяющие правильную дату невисокосного года. Вывести значения D и M для даты, следующей за указанной.

17. Дано целое число в диапазоне 100 – 999. Вывести строку — словесное описание данного числа, например: 256 — «двести пятьдесят шесть», 814 — «восемьсот четырнадцать».

18. В восточном календаре принят 60-летний цикл, состоящий из 12-летних подциклов, обозначаемых названиями цвета: зеленый, красный, желтый, белый и черный. В каждом подцикле годы носят названия животных: крысы, коровы, тигра, зайца, дракона, змеи, лошади, овцы, обезьяны, курицы, собаки и свиньи. По номеру года вывести его название, если 1984 год был началом цикла — годом зеленой крысы.

19. Составьте программу, реализующую эпизод применения компьютера в книжном магазине. Компьютер запрашивает стоимость книг, сумму денег, внесенную покупателем; если сдачи не требуется, печатает на экране «спасибо»; если денег внесено больше, то печатает «возьмите сдачу» и

указывает сумму сдачи; если денег недостаточно, то печатает об этом сообщение, указывающее размер недостающей суммы.

20. Даны три переменные: X , Y , Z . Если их значения упорядочены по убыванию, то удвоить их; в противном случае заменить значение каждой переменной на противоположное.