**Простые операторы в Паскале**

Вопрос 1.

Решение самой простой задачи на компьютере не обходится без операций ввода-вывода информации. Ввод данных – это передача информации с внешнего носителя в оперативную память для обработки. Вывод – обратный процесс, когда данные передаются после обработки из оперативной памяти на внешний носитель.

Процедура чтения READ.

Обеспечивает ввод числовых данных, символов, строк и т.п. для последующей обработки программой. Формат:

Read (x1, x2, …, xN);

Readln (x1, x2, …, xN);

Где х1, х2, … - переменные допустимых типов данных. Эти значения вводятся с клавиатуры и отображаются на экране. Значения должны вводится в строгом соответствии с синтаксисом языка Паскаль.

Например.

Var

I : real;

P: integer;

K : char;

Begin

Read (I, P, K) ;

……..

Вариант ввода: 221.23 34 ‘П’

Процедура чтения Readln аналогична процедуре Read, отличие заключается в том, что после считывания в списке одного значения, следующие данные будут считываться с начала новой строки.

Если в примере заменить Read на Readln, то после ввода с клавиатуры первого значения курсор перейдет на новую строку.

Процедура записи WRITE.

Производит вывод числовых данных, символов, строк и т.п. Формат:

Write (y1, y2, …,yN);

Writeln (y1, y2, …, yN);

Где y1, y2, … - выражения типа integer, real, char, byte и т.д.

Форматы вывода числа.

|  |  |  |
| --- | --- | --- |
| Значение Х | Выражение | Результат |
| 134 | Write ( x ) | 134 (с позиции курсора) |
| 134 | Write ( x :10) | \*\*\*\*\*\*\*134 (крайняя правая позиции поля шириной 10) |
| 234,36 | Write ( x ) | 2.3436000000E+02 (с плавающей точкой) |
| 234,36 | Write ( x : 8 : 2) | \*\*234.36 (с фиксированной точкой) |
| 234,36 | Write ( x : 8 : 4) | 234.3600 |
| 234,36 | Write ( x :10 : 1) | \*\*\*\*\*234.4 |

(\* означает пробел)

Форматы вывода чисел.

|  |  |  |
| --- | --- | --- |
| Значение A, B | Выражение | Результат |
| 3 и 45 | Write ( A, B ) | 345 (с позиции курсора) |
| 3 и 45 | Write (‘A=’ , A ,’\*B=’, B) | A=3 B=45 |
| 3 и 45 | Writeln ( A, B) | 3  45 |
| 3 и 45 | Write (A : 4, B : 4) | \*\*\*3\*\*45 (крайняя правая позиция) |
| 3 и 45 | Writeln ( A:4, B:4) | \*\*\*3  \*\*45 |
| 3 и 45 | Writeln (‘A=’ , A:4, ‘B=’ , B:4) | A=\*\*\*3  B=\*\*45 |
| 3 и 45 | Write (A:4);  Write (B:4); | \*\*\*3  \*\*45 |

Вопрос 2.

Оператор безусловного перехода GO TO.

Это оператор означает «перейти к» и применяется в случаях, когда после выполнения некоторого оператора надо выполнить не следующий по порядку, а какой-либо другой оператор, отмеченный меткой.

Формат:

Go to N;

Где N – метка, метка может содержать как цифровые, так и буквенные символы.

Использование безусловной передачи управления в программе считается теоретически избыточной, т.к. способствует созданию малопонятных и запутанных программ, которые вызывают большие сложности при отладке и сопровождении. Поэтому рекомендуется минимальное использование оператора с соблюдением следующих правил:

следует стремиться применять оператор перехода для управления только вниз по тексту программы;

расстояние между меткой и оператором перехода на нее не должно превышать одной страницы текста (или экрана).

Оператор очистки экрана.

Формат: ClrScr;

Структурные операторы.

Такие операторы представляют собой конструкции, построенные из других операторов по строго определенным правилам. Все структурные операторы можно разделить на три группы: составные, условные, повтора.

Составной оператор п\с группу из произвольного числа операторов, отделенных друг от друга точкой с запятой, и ограниченную операторными скобками begin end. Составной оператор воспринимается как единое целое и может находиться в любом месте программы, где синтаксис языка допускает наличие операторов.

Условные операторы предназначены для выбора на исполнение одного из возможных действий (операторов) в зависимости от некоторого условия (при этом одно из действий может быть пустым или отсутствовать). В качестве условий выбора используется значение логического выражения. В Паскале имеются два условных оператора.

Оператор условия IF.

Это оператор является одним из самых популярных средств, изменяющих естественный порядок выполнения операторов программы. Синтаксически оператор условия IF выглядит следующим образом:

If <условие> then <оператор1> else <оператор2> ; (оператор полной формы)

If <условие> then <оператор> ; (оператор сокращенной формы)

Оператор выполняется следующим образом. Сначала вычисляется выражение, записанное в условии. В результате его вычисления получается значение, если оно истинно, то выполняется оператор 1, если результат ложный, то выполняется оператор 2 или оператор, следующий за оператором IF. Операторы IF могут быть вложенными.

Пример. Составить алгоритм и программу вычисления частного двух целых чисел.

Ввод a, b

B=0

На ноль делить нельзя

Result:=a/b

Result

Program delenie;

Uses Crt;

Var a, b : integer;

Result : real ;

Begin

ClrScr ; {оператор очистки экрана}

Writeln ( ‘ Введите значение делимого А ‘);

Readln (a);

Writeln ( ‘ Введите значение делителя В ’ );

Readln (b);

If b=0 then writeln (‘на ноль делить нельзя ’)

Else begin {начало составного оператора}

Result := a / b ;

Writeln (‘Частное чисел равно’, Result :8:2);

End; { конец составного оператора}

Readln {задержка результата}

End.

Оператор выбора CASE.

Если один оператор IF может обеспечить выбор из двух альтернатив, то оператор выбора позволяет сделать выбор из произвольного числа имеющихся вариантов. Он состоит из выражения, называемого селектором, и списка параметров, каждому из которых предшествует список констант выбора (список может состоять из одной константы.

Формат оператора:

Case <выражение - селектор> of

<список 1> : <оператор 1> ;

<список 2> : <оператор 2> ;

…………

<список N> : <оператор N> ;

Else <оператор> ;

End;

Оператор Case работает следующим образом. Сначала вычисляется значение выражения-селектора, затем обеспечивается реализация того оператора, константа выбора которого равна текущему значению селектора. Если ни одна из констант не рана текущему значению селектора, то выполняется оператор стоящий за словом Else. Если слово Else отсутствует, то активируется оператор, находящийся за словом End, т.е. первый оператор за границей Case.

Селектор должен относится к одному из целочисленных типов. Список констант выбора состоит из произвольного количества значений, или диапазонов, отделенных друг от друга запятыми. Границы диапазона записываются двумя константами через разграничитель «..». Тип константы должен совпадать с типом селектора.

При использовании оператора выбора должны выполняться следующие правила:

Значения выражения «переключателя» записанного после слова Case, должны принадлежать дискретному типу, для целого типа они должны лежать в диапазоне integer;

Все константы, предшествующие операторам альтернатив, должны иметь тип, совместимый с типом выражения;

Все константы в альтернативах должны быть уникальны в пределах оператора варианта, диапазоны не должны пересекаться и не должны содержать констант, указанных в данной или других альтернативах.

Пример: составить программу вывода названия дня недели по его номеру.

Program DAY;

Var D: byte ;

Begin

Write (‘Введите номер дня недели’);

Readln ( D );

Case D of {вычисление значения селектора и выбор}

1: writeln ( ‘Понедельник’ ) ;

2: writeln ( ‘Вторник’ ) ;

3: writeln ( ‘Среда’ ) ;

4: writeln ( ‘Четверг’ ) ;

5: writeln ( ‘Пятница’ ) ;

6: writeln ( ‘Суббота’ ) ;

Else

Writeln ( ‘Воскресенье’ );

End;

Readln;

End.

Вопрос 3.

Если в программе возникает необходимость неоднократного выполнения некоторых операторов, то используются операторы повтора (или цикла). В языке Паскаль различают три вида операторов цикла: while, repeat, for. Они используются для организации циклов различных типов.

Если число повторений оператора заранее неизвестно, а задано лишь условие его повторения (или окончания), используются операторы while, repeat. Оператор for используется, если число повторений известно заранее.

Оператор WHILE.

Его часто называют оператором цикла с предусловием, за то что проверка условия выполнения тела цикла производится в самом начале оператора.

Формат оператора:

While <условие продолжения цикла> do

<тело цикла>;

Условие – булевское выражение, тело цикла – простой или составной оператор. Перед каждым выполнением тела цикла вычисляется значение выражения условия, если результат истина, то тело цикла выполняется. Если результат ложь, то происходит выход из цикла и переход к первому оператору после while оператора.

Оператор повтора REPEAT.

Этот оператор отличается от предыдущего тем, что условие проверяется после очередного выполнения операторов тела цикла и т.о. гарантируется хотя бы однократное выполнение цикла. Второе отличие от первого оператора: прекращением цикла является равенство выражения константе истина. За это оператор repeat часто называют циклом с постусловием.

Формат оператора:

Repeat

<оператор> ;

…………….

<оператор>;

Until <условие окончания цикла>;

Операторы, заключенные между словами repeat / until , являются телом цикла. Сначала выполняется тело цикла, затем проверяется условие выхода из цикла.

Оператор повтора FOR.

В случае когда число повторений известно, для организации циклической обработки информации применяется оператор повтора For. Часто этот оператор называют оператором цикла с параметром, т.к. число повторений задается переменной, которую называют параметром цикла (или управляющей переменной). Этот оператор состоит из заголовка и тела цикла.

Формат оператора:

For <параметр цикла> :=A1 to A2 do <оператор> ; (по возрастанию)

For <параметр цикла> :=A1 downto A2 do <оператор> ; (по убыванию)

Где А1 и А2 – выражения, определяющие соответственно начальное и конечное значение параметра цикла.

Тело цикла может являться простым или составным оператором. Оператор For обеспечивает выполнение тела цикла до тех пор, пока не будут перебраны все значения параметра цикла от начального до конечного.

Заголовок оператора повтора For определяет:

диапазон изменения значений управляющей переменной и одновременное число повторений оператора, содержащегося в теле цикла;

направление изменения параметра цикла (возрастание – to или убывание – downto).

На использование параметра цикла накладываются следующие ограничения:

В качестве параметра должна использоваться простая переменная, описанная в текущем блоке.

Управляющая переменная должна иметь дискретный тип.

Начальное и конечное значение диапазона должны иметь совместимый тип с параметром цикла. При этом допускаются любые скалярные типы, кроме вещественного.

В теле цикла запрещается явное изменение значения управляющей переменной.

Значение управляющей переменной изменяется на +1 или -1, т.е. шаг равен единицы.

Пример. Составить программу для перевода расстояний из дюймов в сантиметры (1дюйм = 2,5 см) для значений длин от 1 до 20 дюймов.

Пусть R1 – расстояние в дюймах, а R2 – это расстояние в см.

1 способ.

Program sb1;

Var

R2 : real ;

R1 : integer ;

Begin

Writeln (‘Таблица перевода расстояний из дюймов в сантиметры’);

For R1:=1 to 20 do

Begin

R2 := 2.5\*R1;

Writeln (R1 :6, ‘ ‘, R2 :8:2 );

End;

Readln;

End.

2 способ.

Program sb2;

Var

R2 : real ;

R1 : integer ;

Begin

Writeln (‘Таблица перевода расстояний из дюймов в сантиметры’);

R1 := 1;

While R1 <= 20 do

Begin

R2 := 2.5\*R1;

Writeln (R1 :6, ‘ ‘, R2 :8:2 );

R1 := R1+ 1;

End;

Readln;

End.

3 способ.

Program sb3;

Var

R2 : real ;

R1 : integer ;

Begin

Writeln (‘Таблица перевода расстояний из дюймов в сантиметры’);

R1 := 1;

Repeat

R2 := 2.5\*R1;

Writeln (R1 :6, ‘ ‘, R2 :8:2 );

R1 := R1+ 1;

Until R1>20;

Readln;

End.