Лабораторная работа 3

ОПЕРАТОР УСЛОВНОГО ПЕРЕХОДА

Цель работы: познакомиться с операторами ввода/вывода, присвоения, безусловного перехода, составными и пустыми операторами языка Pascal. Научиться применять оператор условного перехода.

Основные понятия

Операторы ввода-вывода. Для ввода данных в Паскале используются операторы READ и READLN. При вводе данных с клавиатуры действия, выполняемые этими двумя операторами, будут практически одинаковыми. Разница только в том, что после ввода READLN переводит курсор на новую строку. Синтаксис оператора READLN:

READLN (a1, a2, …),

где a1, a2 … – список переменных, в которые осуществляется ввод данных.

Оператор READLN работает следующим образом: программа останавливается в ожидании ввода, пользователь вводит данные в соответствии со списком переменных, перечисленных в скобках. Данные, вводимые с клавиатуры, друг от друга отделяются пробелами или вводятся через Enter. Ввод заканчивается нажатием клавиши Enter. Далее программа распределяет данные по переменным a1, a2, …

Если в программе встретится оператор READLN без списка переменных, то программа будет ожидать нажатие клавиши Enter (т.е. ввода пустой строки). Обычно такой оператор ставится в конце программы, чтобы можно было сначала оценить результаты, а потом нажатием Enter завершить программу, т.е. для организации паузы.

Вывод данных на экран осуществляется посредством операторов WRITE и WRITELN. Отличие в работе оператора WRITE и WRITELN, заключается в том, что оператор WRITELN после вывода значений всех переменных и констант из списка осуществляет перевод курсора на новую строку экрана. Синтаксис оператора WRITELN:

WRITELN(a1, a2, …),

где a1, a2, … – список вывода, в котором кроме имен переменных можно писать строковые константы (последовательность символов в апострофах) и даже выражения (выводятся их результаты).

Все операторы в Pascal отделяются друг от друга символом «;».

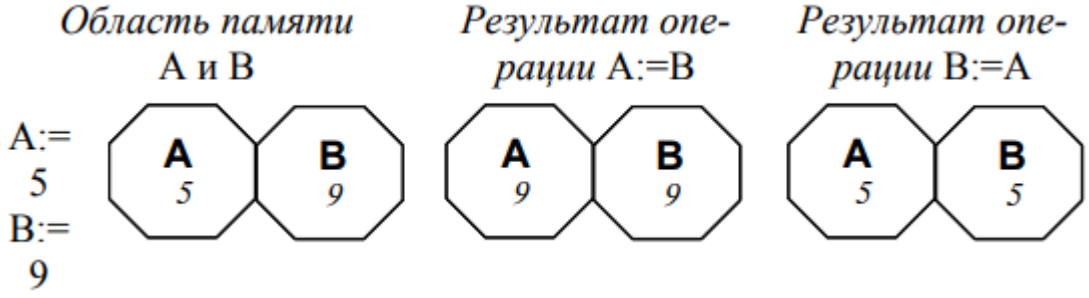
Оператор присваивания. Синтаксис оператора присваивания:

Имя переменной := Выражение.

Переменная (левая часть) и выражение (правая часть) должны быть одного типа.

Данное выражение следует читать: «К присвоить А». Понимается это так, что значение, хранимое в области памяти с именем А, помещается в область памяти с именем К.

Лучше всего продемонстрировать действие операции присвоения на примере. Допустим, значения 5 и 9 присваиваются переменным А и К.

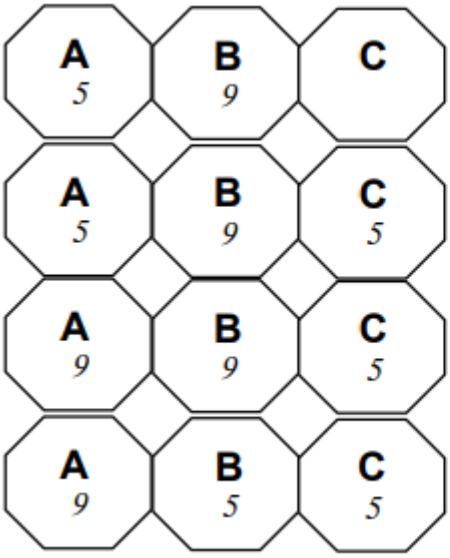


Следует всегда помнить, что в программировании переменные – это не только абстрактные имена, но и конкретные области памяти, которые хранят значения.

Допустим, необходимо сделать так, чтобы A = 9, B = 5. Если на писать A:=B, то получится результат:

A:=B

При этом значение 5 будет потеряно безвозвратно. Поэтому нужна третья переменная C, в которой временно будет храниться значение переменной A.



Дополнительная переменная C

Сохраняем значение A: B:=A

Значение B помещаем в A: A:=B

Первоначальное значение A перемещаем в B: B:=C

Составной оператор и пустой оператор. Составной оператор – это последовательность произвольных операторов программы, заключенная в операторные скобки – зарезервированные слова BEGIN … END. Составные операторы – важный инструмент Pascal, дающий возможность писать программы по современной технологии структурного программирования (без операторов перехода GOTO).

Pascal допускает произвольную глубину вложенности составных операторов:

```
BEGIN
      ……
      BEGIN
```

```
        ……
        BEGIN
                ……
        END;
        ……
    END;
    ……
END.
```

Поскольку BEGIN и END представляют собой структурные скобки, то после BEGIN и перед END ставить знак «;» не обязательно.

В программе может применяться пустой оператор, не выполняющий никакого действия. Например – ; ;.

Оператор условного перехода. Оператор условного перехода (условный оператор) позволяет проверить некоторое условие и в зависимости от результатов проверки выполнить то или иное действие. Таким образом условный оператор – это средство ветвления вычислительного процесса. Структура условного оператора имеет вид:

IF условие THEN оператор1 ELSE оператор2,

где IF, THEN, ELSE – зарезервированные слова («если», «то», «иначе»); оператор1, оператор2 – любые операторы языка Паскаль (в том числе и составные).

Условный оператор работает по следующему алгоритму. Вначале вычисляется условие, если результат True (истина), то выполняется оператор1, а оператор2 пропускается; если результат False (ложь), то, наоборот, оператор1 пропускается, а выполняется оператор2.

Оператор IF может быть неполным, т.е. часть «ELSE оператор2» может быть опущена. Тогда при значении True условного выражения выполняется оператор1, в противном случае он пропускается.

Если оператор1 и оператор2 – составные, то условный оператор будет иметь вид:

```
IF условие THEN
    BEGIN
    ……
    END
ELSE
    BEGIN
    ……
    END.
```

Пример. Напишите программу, определяющую наименьшее значение из двух чисел, введенных с клавиатуры.

```
PROGRAM Minimum;
VAR a, b, min : real;
BEGIN
    WRITELN('Введите два числа');
    READLN(a, b);
    IF a<b Then min:=a ELSE min:=b;
    WRITELN(min);
END.
```

Задания

Используя оператор условного перехода, напишите программу, которая:

1) классифицирует компьютерную сеть. Программа запрашивает у пользователя число компьютеров в сети и в зависимости от введенного

количества выводит класс сети (если число ЭВМ меньше 256 – то это сеть класса С, от 256 до 65535 – сеть класса В, свыше 65535 – сеть класса А);

2) запрашивает у пользователя номер одного из весенних месяцев, и выводит количество дней в этом месяце. Программа должна проверять, является ли введенный месяц весенним;

3) выводит на экран приглашение: «Который час?», вводит с клавиатуры число Х, имеющее смысл времени суток, и печатает слова «Доброе утро», «Добрый вечер», «Добрый день» в зависимости от введенного времени. Программа должна реагировать на ввод неправильного времени: меньше 0 или больше 24;

4) запрашивает с клавиатуры у пользователя размер хищений (р.), определяет и выводит на экран масштаб в соответствии с принятой классификацией (например, если размер хищений меньше 100 р. – «мелкий», от 100 до 1000 – «крупный», свыше 1000 – «особо крупный»);

5) выводит на экран приглашение: «Введите месяц», вводит с клавиатуры число Х, имеющее смысл месяца, и печатает слова «Зима», «Весна», «Лето», «Осень» в зависимости от введенного месяца. Программа должна реагировать на ввод неправильного месяца: меньше 1 или больше 12;

6) запрашивает у пользователя номер одного из летних месяцев, и выводит количество дней в этом месяце. Программа должна проверять, является ли введенный месяц летним;

7) вводит с клавиатуры три числа и выводит на экран максимальное из них;

8) выводит на экран приглашение: «Введите день недели», вводит с клавиатуры число Х, имеющее смысл дня недели, и печатает слова «Рабочий день», «Короткий день», «Выходной» в зависимости от

введенного дня. Программа должна реагировать на ввод неправильного дня недели: меньше 1 или больше 7;

9) запрашивает у пользователя номер одного из осенних месяцев, и выводит количество дней в этом месяце. Программа должна проверять, является ли введенный месяц осенним;

10) запрашивает с клавиатуры два целых числа, их сумму и произведение и выводит на экран сообщение о правильности сделанных пользователем вычислений;

11) вводит с клавиатуры три числа и выводит на экран минимальное из них;

12) запрашивает с клавиатуры два целых числа, их разность и частное (результат деления) и выводит на экран сообщение о правильности сделанных пользователем вычислений;

13) запрашивает у пользователя номер одного из зимних месяцев, и выводит количество дней в этом месяце. Программа должна проверять, является ли введенный месяц зимним;

14) вводит с клавиатуры число X. Если X меньше 10, то вычисляет и выводит на экран квадрат числа X, а если больше или равно, то вводит новое число Y, а затем вычисляет и выводит на экран значение суммы X и Y;

15) запрашивает с клавиатуры три произвольных числа и выводит на экран сообщение о том, сколько из них отрицательных, положительных и нулевых.

Лабораторная работа 4

ОДНОМЕРНЫЕ МАССИВЫ

Цель работы: научиться работать с одномерными массивами и операторами цикла WHILE...DO и REPEAT...UNTIL.

Основные понятия

Массив – это упорядоченный набор переменных, которым присвоено одно имя. К необходимости применения массивов мы приходим каждый раз, когда требуется связать и использовать целый ряд родственных величин.

В одномерном массиве элементы располагаются в последовательных ячейках памяти, т.е. массив занимает непрерывную область памяти. Каждый элемент массива имеет свой номер, являющийся целым числом и называющийся индексом элемента. Иногда массив так и называют переменные с индексами. Для использования в программе массива требуется предварительно описать его в разделе описания переменных:

Имя : ARRAY [Мин_индекс…Макс_индекс] OF Тип,

где Имя – имя массива, ARRAY; OF – зарезервированные для описания массивов слова; Мин_индекс, Макс_индекс – минимальное и максимальное значения индекса массива; Тип – тип переменных массива. Минимальное и максимальное значения индекса массива определяют его размерность.

Доступ к элементу одномерного массива в программе осуществляется путем указания имени массива и после него в квадратных скобках номера этого элемента в массиве.

Для ввода данных в массив и вывода из массива используются циклы. Цикл – многократно выполняемый участок вычислительного процесса.

Паскаль позволяет использовать три различных оператора для организации циклов:

• оператор цикла с параметром;

• оператор цикла с предварительной проверкой условия;

• оператор цикла с последующей проверкой условия.

В данной работе познакомимся с операторами цикла с предварительной и последующей проверкой условий.

Оператор цикла с предварительной проверкой условия

WHILE условие DO оператор,

где WHILE, DO – зарезервированные слова [пока (выполняется условие), делать]; условие – выражение логического типа; оператор – произвольный оператор языка Паскаль.

Оператор WHILE работает следующим образом: если выражение условие имеет значение True, то выполняется оператор, после чего вычисление выражения условие и его проверка повторяются. Если условие имеет значение False, оператор WHILE прекращает свою работу. Поскольку значение логического выражения проверяется в начале каждой итерации, то тело цикла может не выполниться ни разу. Таким образом, в этом цикле логическое выражение – это условие продолжения работы в цикле.

Оператор цикла с последующей проверкой условия

REPEAT тело_цикла UNTIL условие,

где REPEAT, UNTIL – зарезервированные слова (повторять до тех пор, пока не будет выполнено условие); тело_цикла – произвольная последовательность операторов языка Паскаль; условие – выражение логического типа.

Оператор REPEAT работает следующим образом: выполняются операторы, входящие в тело_цикла, после этого вычисляется значение логического выражения условие: если False, то выполнение операторов тело_цикла повторяется, в противном случае оператор REPEAT завершает свою работу. Тело цикла обязательно выполняется хотя бы один раз. Таким образом, в этом цикле логическое выражение – это условие выхода из цикла.

Пример. Программа поиска наибольшего значения в одномерном массиве, размерностью 10.

```
PROGRAM Maximum;
```

```
VAR i, n : integer;
Massiv : ARRAY[1..10] OF Real;
    Begin
    i:=1;
    WHILE i<=10 DO
    BEGIN
    WRITE('Введите ', i, '-й элемент: ');
    READLN(Massiv[i]);
    i:=i+1;
    END;
    n:=1;
    i:=2;
    REPEAT
    IF Massiv[i]>Massiv[n] THEN n:=i;
    i:=i+1;
    UNTIL i>10;
    WRITELN('Максимальный элемент ', n, ' = ', Massiv[n]:7:4);
END.
```

Задания

Напишите программу, которая ввод данных в одномерный целочисленный массив осуществляет посредством цикла WHILE...DO, а вывод – REPEAT...UNTIL (в скобках указана размерность массива):

1) заменить все отрицательные элементы массива нулями (12);

2) увеличить элементы массива с четными индексами на «1» (11);

3) сделать все положительные элементы массива отрицательными (10);

4) заменить каждый отрицательный элемент произведением всех ненулевых элементов массива (8);

5) заменить первый элемент массива максимальным (11);

6) увеличить каждый ненулевой элемент массива на «3» (12);

7) заменить все элементы массива, которые больше 10, на 100 (12);

8) увеличить все положительные элементы массива на единицу (12);

9) заменить все нулевые элементы массива на единицу (11);

10) заменить минимальный элемент массива на единицу (12);

11) увеличить элементы массива с нечетными индексами на «2» (10);

12) увеличить каждый третий элемент массива в 2 раза (12);

13) заменить последний элемент массива минимальным (10);

14) заменить все элементы массива, которые делятся без остатка на «3», на единицы (9);

15) увеличить все ненулевые элементы массива в 3 раза (11);

16) заменить каждый положительный элемент суммой всех элементов массива (8);

17) уменьшить каждый ненулевой элемент массива на 2 (11);

18) заменить каждый второй элемент массива на «1» (15);

19) увеличить все положительные элементы массива, которые меньше 10, на «10» (10);

20) заменить все элементы массива, которые меньше среднего значения, на «3» (12).

Лабораторная работа 5

ОБРАБОТКА ДВУХМЕРНЫХ МАССИВОВ

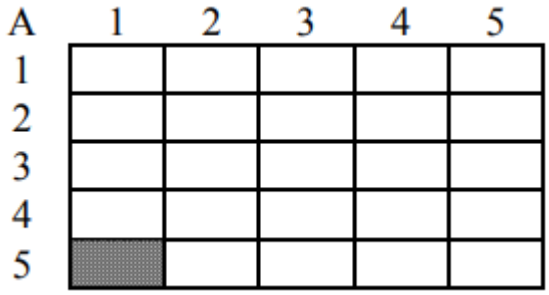
Цель работы: научиться работать с двухмерными массивами.

Основные понятия

Массив – это упорядоченный набор переменных, которым присвоено одно имя. К необходимости применения массивов мы приходим каждый раз, когда требуется связать и использовать целый ряд родственных величин.

В одномерном массиве элемент определяется при помощи одного индекса, поэтому такой массив можно представить в виде строки. Доступ к элементам двухмерного массива осуществляется посредством двух индексов, что позволяет представить двухмерный массив в виде таблицы, в которой первый

индекс определяет номер строки, а второй – номер столбца. На пересечении столбца и строки находится определенный элемент.



Данный двухмерный массив в программе описывается следующим образом:

VAR A : ARRAY [1..5, 1..5] OF REAL.

Элемент, расположенный в 5-ой строке и первом столбце имеет имя A[5, 1].

Оператор цикла с параметром. Оператор цикла FOR организует выполнение одного оператора заранее определенное число раз. Синтаксис оператора:

FOR пар_цик := нач_знач TO кон_знач DO оператор,

где FOR, TO, DO – зарезервированные слова («для», «до», «выполнить»); пар_цик – параметр цикла – переменная типа Integer; нач_знач – начальное

значение параметра цикла; кон_знач – конечное значение параметра цикла; оператор – произвольный оператор языка Паскаль.

На первом шаге цикла параметр принимает значение нач_знач. В этот же момент происходит вычисление кон_знач – значения параметра на последнем шаге цикла. После каждого выполнения тела цикла, если параметр цикла не равен кон_знач, происходит изменение параметра на следующее большее или меньшее значение в зависимости от формы оператора FOR.

В случае нач_знач > кон_знач в первой форме оператора или нач_знач < кон_знач во второй его форме ошибки не происходит, но цикл не выполняется ни разу. После завершения работы цикла значение параметра остается равным кон_знач.

На месте нач_знач и кон_знач могут находиться выражения целого типа (n+2, 2*k+n и т.д.), а оператор может быть составным оператором.

Для того чтобы установить шаг наращивания параметра цикла –1 вместо служебного слова TO пишется DOWNTO.

Для работы с двухмерным массивом в программе используется алгоритм с вложенными циклами.

Пример. Программа заполнения двухмерного массива 3 × 5 и поиска в нем наименьшего элемента.

```
PROGRAM Minimum;
        VAR i, j : Integer;
        M : ARRAY [1..3, 1..5] OF Real;
         min : Real;
BEGIN
        FOR i:=1 TO 3 DO
        FOR j:=1 TO 5 DO
        BEGIN
```

```
        WRITE('M[',i,',',j,']=');
        READLN(M[i,j]);
        END;
        min:=M[1,1];
        FOR i:=1 TO 3 DO
        FOR j:=1 TO 5 DO
        IF M[i,j]<min THEN min:=M[i,j];
        WRITELN('Минимальный элемент=', min:7:4);
END.
```

Задания

Напишите программу, в которой нужно:

1) вывести столбец и строку двухмерного массива 4 × 5, на пересечении которых расположен максимальный элемент;

2) определить номера строк двухмерного массива 5 × 3, содержащих только положительные элементы;

3) для каждого столбца двухмерного массива 3 × 5 рассчитать произведение ненулевых элементов;

4) подсчитать произведение неотрицательных элементов в двухмерном массиве 6 × 3;

5) в двухмерном массиве 6 × 3 для каждой нечетной строки определить произведение положительных элементов;

6) подсчитать произведение отрицательных элементов в двухмерном массиве 4 × 5;

7) для каждой строки двухмерного массива 5 × 4 рассчитать среднее значение;

8) определить количество элементов в двухмерном массиве 3 × 6, которые больше 2, но меньше 6;

9) в двухмерном массиве 3 × 6 для каждого четного столбца определить сумму элементов;

10) вывести столбец и строку двухмерного массива 5 × 4, на пересечении которых расположен минимальный элемент;

11) определить номера столбцов двухмерного массива 3 × 5, содержащих только отрицательные элементы;

12) для каждой строки двухмерного массива 5 × 4 определить максимальное значение;

13) определить количество положительных элементов в двухмерном массиве 5 × 4;

14) в двухмерном массиве 5 × 4 для каждой нечетной строки определить сумму элементов;

15) подсчитать произведение положительных элементов в двухмерном массиве 3 × 6;

16) определить количество элементов в двухмерном целочисленном массиве 6 × 3, делящихся на

«3» без остатка;

17) для каждой строки двухмерного массива 6 × 4 посчитать количество положительных элементов;

18) определить номера строк двухмерного массива 5 × 4, сумма элементов которых больше 50;

19) определить номера столбца двухмерного массива 4 × 3, произведение элементов которых меньше 15;

20) в двухмерном массиве 5 × 3 определить количество элементов, которые больше 10.