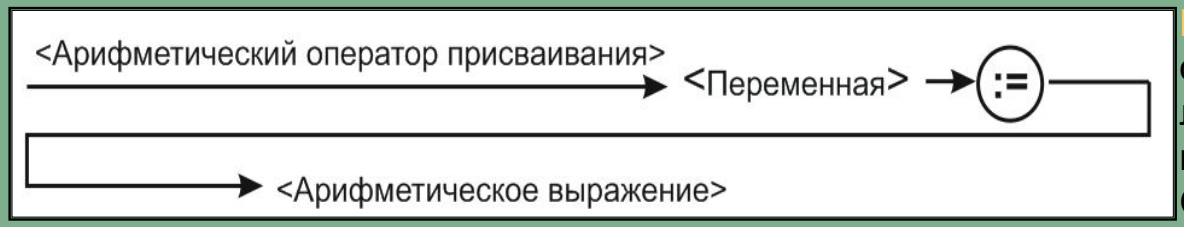
ОПЕРАТОРЫ ЯЗЫКА ПРОГРАММИРОВАНИЯ PASCAL

Операторы языка Паскаль можно разделить на **простые** и **сложные**. **ПРОСТЫЕ ОПЕРАТОРЫ** не содержат внутри себя других операторов. **СЛОЖНЫЕ (СТРУКТУРНЫЕ) ОПЕРАТОРЫ** представляют собой конструкции, содержащие простые операторы.

К простым операторам в языке Паскаль относятся операторы: присваивания, пустой оператор, операторы ввода и вывода, к сложным – составной и условный операторы, операторы цикла, оператор выбора (варианта), оператор присоединения к записям.

АРИФМЕТИЧЕСКИЙ ОПЕРАТОР ПРИСВАИВАНИЯ



ОПЕРАТОР ПРИСВАИВАНИЯ – основной оператор любого языка программирования. Общая форма записи оператора:

V := A

Здесь V– имя переменной; «:=»- знак присваивания; А–выражение. Данный оператор вычисляет значение выражения А, стоящего справа от знака операции присваивания:=, и присваивает

полученное значение переменной V, стоящей слева. Разница между знаком операции присваивания «:=» и обычным знаком равенства «=» не только в форме, но и в содержании.

В частных случаях выражение в правой части оператора присваивания может принимать значение константы, имени переменной или имени функции.
Например:
T := 527.475
M :=ТЕМП;
Y :=SQRT (X);

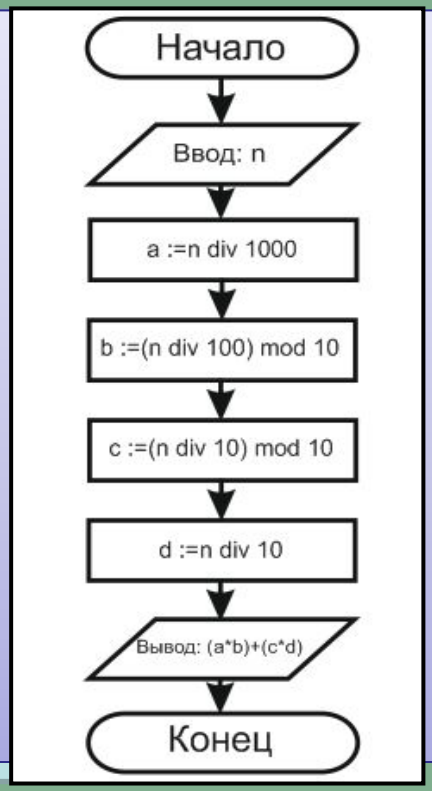
ПРАВИЛО: ТИПЫ ПЕРЕМЕННОЙ И ВЫРАЖЕНИЯ ДОЛЖНЫ БЫТЬ ОДИНАКОВЫМИ

Примеры оператора присваивания:
Y := A+ROUND(B/3)*2;
SUM := SUM+X;
C5 := 2*K-SIN(PI/4-X).
Пример. Вводится четырёхзначное число (abcd). Вывести сумму ab + cd.
Метод решения задачи:
В данном случае проще обойтись без цикла, т.к. известна разрядность числа.
Для извлечения цифры из старшего разряда, надо число нацело поделить на 1000. (Или найти остаток от деления на 10000.)
Для извлечения второго (по старшинству) разряда, сначала делим число на 100, далее избавляемся от первого разряда, найдя остаток от деления на 10.
Третья цифра извлекается также как вторая за исключением того, что делить нацело надо на 10.

Четвертая (последняя) цифра извлекается как остаток от деления исходного числа на 10.

ПРИМЕР

```
PROGRAM A_01;
VAR
   N, A, B, C, D: INTEGER;
BEGIN
   READLN(N); // 8123
   A := N DIV 1000; // 8
   B := (N DIV 100) MOD 10; // 81 MOD 10 = 1
   C := (N DIV 10) MOD 10; // 812 MOD 10 = 2
   D := N MOD 10; // 3;
WRITELN('AB+CD=',A*B,'+',C*D,'=', A*B+C*D);
READLN;
END.
```



СОСТАВНОЙ ОПЕРАТОР

СОСТАВНОЙ ОПЕРАТОР – объединение нескольких операторов в одну группу. Форма записи данного оператора:

```
BEGIN
оператор1;
оператор2;
…
операторn-1;
оператор n
END
```

В этой конструкции служебные слова **BEGIN** (начало) и **END** (конец) называются *операторными скобками*. **Слово BEGIN** выполняет роль открывающей скобки, **слово END** – роль закрывающей скобки.
Составной оператор представляется как единый оператор. Его можно вставлять в любое место программы, где допускается один оператор. Любой из операторов составного оператора, в свою очередь, также может быть составным.

ПУСТОЙ ОПЕРАТОР

ПУСТОЙ ОПЕРАТОР – это оператор, не выполняющий никакого действия.

ВВОД ДАННЫХ – это передача информации от внешних устройств в оперативную память. Вводятся, как правило, исходные данные решаемой задачи. Основным устройством ввода ПК является клавиатура.
Для ввода данных в языке Паскаль предусмотрены стандартные встроенные программы, которые называются процедурами.
Оператор ввода служит для ввода данных в процессе выполнения программы. При этом значения вводимых данных получают переменные. Данные могут быть разбиты на отдельные строки. Признаком конца строки является клавиша «Enter».
Различают три вида оператора ввода:

READ (a1,a2,a3, …,an) – каждое вводимое значение получают последовательно переменные a1, a2, a3, …,an.
READLN (a1,a2,a3, …,an) – каждое вводимое значения получают последовательно переменные a1, a2, a3, …,an, после чего происходит переход на новую строку (следующий оператор ввода будет вводить данные с новой строки).
READLN – переход на новую строку при вводе данных.
Последовательно расположенные операторы вида 1) и 3) эквиваленты одному оператору 2).

ОПЕРАТОР ВВОДА ДАННЫХ

ВВОД ДАННЫХ – это передача информации от внешних устройств в оперативную память. Вводятся, как правило, исходные данные решаемой задачи. Основным устройством ввода ПК является клавиатура.
Для ввода данных в языке Паскаль предусмотрены стандартные встроенные программы, которые называются процедурами.
Оператор ввода служит для ввода данных в процессе выполнения программы. При этом значения вводимых данных получают переменные. Данные могут быть разбиты на отдельные строки. Признаком конца строки является клавиша «Enter».

Различают три вида оператора ввода:

READ (a1,a2,a3, …,an) – каждое вводимое значение получают последовательно переменные a1, a2, a3, …,an.

READLN (a1,a2,a3, …,an) – каждое вводимое значения получают последовательно переменные a1, a2, a3, …,an, после чего происходит переход на новую строку (следующий оператор ввода будет вводить данные с новой строки).

READLN – переход на новую строку при вводе данных. Последовательно расположенные операторы вида 1) и 3) эквиваленты одному оператору 2).

ВВОД ЧИСЛОВЫХ ДАННЫХ.

Числовые данные, целые и действительные, должны разделяться пробелом или нажатием клавиши «Enter».

ПРИМЕР ВВОДА:

VAR A,B,C : REAL; K,T: INTEGER; … READ (A,B,C); READLN; READ (K,T);	ВХОДНОЙ ПОТОК ДАННЫХ: 0.5 6.23 -7.1↵ 3 48↵ ПОСЛЕ ВВОДА: A=0.5, B=6.23, C=-7.1, K=3, T=48

ВВОД СИМВОЛЬНЫХ ДАННЫХ.

Ввод символьных данных имеет свои особенности. Поскольку пробел, как и любой символ языка Паскаль, относится к символьным данным, символьные данные вводятся сплошной строкой в соответствии с оператором ввода. Одной переменной можно присвоить значение только одного символа.
Пример ввода:

VAR A,B,C : CHAR; … READ (A,B,C); …	ВХОДНОЙ ПОТОК ДАННЫХ: SNR↵ ПОСЛЕ ВВОДА: A=S, B=N, C=R
VAR A,B,C : CHAR; … READ (A, B, C); …	ВХОДНОЙ ПОТОК ДАННЫХ: SNR↵ ПОСЛЕ ВВОДА: A=S, B=пробел, C=N

Другая особенность ввода символьных данных заключаются в том, что нажатие клавиши «Enter» воспринимается как символ «пробела».
Пример ввода:

VAR A,B : CHAR; C,D : CHAR; … READ (A, B); READ (C, D); …	ВХОДНОЙ ПОТОК ДАННЫХ: 3 4↵ WF↵ ПОСЛЕ ВВОДА: A=3, B=4, C=пробел, D=W

ОПЕРАТОР ВЫВОДА

ВЫВОД ДАННЫХ – это передача данных из оперативной памяти на внешние носители (печать, дисплей, магнитные устройства). Результаты решения любой задачи должны быть выведены. Основным устройством вывода ПК является дисплей (экран монитора).

Оператор вывода данных из памяти ЭВМ на экран дисплея имеет три формы записи:

WRITE (b1,b2,…,bn) – выводит последовательно значения b1, b2, …, bn.

WRITELN (b1,b2,…,bn) – выводит последовательно значения b1, b2, …,bn и осуществляет переход на новую строку (следующий оператор вывода будет выводить данные на новую строку);

WRITELN – осуществляет переход на новую строку при выводе данных.

Последовательно расположенные операторы вида 1) и 3) эквивалентны одному оператору 2). В качестве параметров b1, b2, …, bn могут быть целые, действительные, символьные и логические переменные, а также символьные константы.

Допускается вывод данных с форматами и без них.

ПОРЯДОК ВЫВОДА НА ЭКРАН

1. Очередной символ выводится в текущее положение курсора.
2. После вывода символа курсор перемещается на одну позицию вправо.
3. Новая процедура Write начинает вывод в текущее положение курсора

ФОРМАТНЫЙ ОПЕРАТОР

ФОРМАТНЫЙ ВЫВОД. Оператор вывода позволяет задать ширину поля вывода для каждой переменной.

Для **ЦЕЛЫХ ПЕРЕМЕННЫХ** формат вывода имеет вид **A:M**, где A – переменная целого типа, M – ширина поля вывода (константа целого типа). Если выводимое значение занимает в поле вывода меньше позиций, чем M, то перед этим значением располагаются пробелы.

Для **ДЕЙСТВИТЕЛЬНЫХ ПЕРЕМЕННЫХ** формат имеет в общем случае следующий вид **A:M:N**, где A - переменная или выражение действительного типа, M - ширина поля вывода, N - число цифр дробной части выводимого значения. M и N - константы целого типа. В этом случае действительные значения выводятся в форме десятичного числа с фиксированной точкой.

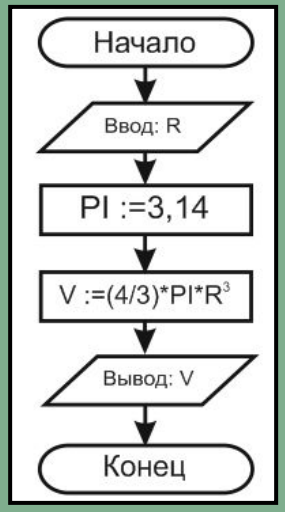
CONST M=6; N=3; VAR SUM, A:REAL; … WRITE (SUM:M:N, A:7:2);	Если число A=21.60, то оно будет выведено в соответствии с форматом в виде: ␣␣21.60 Если число A=-21.60, то оно будет выведено в соответствии с форматом в виде: ␣-21.60

ПРИМЕР. Вычислить объем шара V с радиусом R по формуле $V=(4/3)\pi R^3$.

```
(*-------------------------------------------------------
--------
!       ВЫЧИСЛЕНИЕ ОБЪЕМА ШАРА
!
-------------------------------------------------------
------*)
PROGRAM L_01;
CONST PI=3.14;
  VAR
R: REAL; (*РАДИУСШАРА*)
    V: REAL; (*ОБЪЕМШАРА *)
BEGIN
WRITELN ('ВВЕДИТЕ ЗНАЧЕНИЕ
РАДИУСА R:');
READLN  (R);
    V := 4*PI*R*R*R/3;
    WRITELN;
WRITELN('РЕЗУЛЬТАТ:');
WRITELN('ОБЪЕМ ШАРА=', V:8:3);
READLN;
END.
```

Алгоритм

Результат выполнения:




```
ВВЕДИТЕ ЗНАЧЕНИЕ РАДИУСА R:
0.2

РЕЗУЛЬТАТ:
ОБЪЕМ ШАРА=    0.033
```

```
ВВЕДИТЕ ЗНАЧЕНИЕ РАДИУСА R:
3.5

РЕЗУЛЬТАТ:
ОБЪЕМ ШАРА= 179.503
```

ПРИМЕР. Вывести на экран слово PASCAL шесть раз, используя только оператор вывода.

```
PROGRAM PASCAL;
BEGIN
        WRITELN ('********************');
        WRITELN ('*   P A S C A L    *');
        WRITELN ('*   P A S C A L    *');
        WRITELN ('*   P A S C A L    *');
        WRITELN ('*   P A S C A L    *');
        WRITELN ('*   P A S C A L    *');
        WRITELN ('*   P A S C A L    *');
        WRITELN ('********************');
READLN;
END.
```

Результат выполнения:

УСЛОВНЫЙ ОПЕРАТОР

Условный оператор используется в тех случаях, когда вычисления могут пойти по различным путям, в зависимости от выполнения или невыполнения определенных условий.

Алгоритмическая структура ветвления программируется в Паскале с помощью условного оператора, имеющего вид:

If <условие> **Then** <оператор 1> **Else** <оператор 2>;

Кроме того, возможно использование неполной формы условного оператора:

If <условие> **Then** <оператор>

Здесь IF (если), THEN (тогда), ELSE (иначе) – служебные слова. Условием в условном операторе является логическое выражение, которое вычисляется в первую очередь. Если его значение равно TRUE, то будет выполняться <Оператор 1> (после THEN), если же его значение равно FALSE, будет выполняться <Оператор 2> (после ELSE) для полной формы или сразу оператор, следующий после условного, для неполной формы (без ELSE).

ПРИМЕР. Требуется составить программу вычисления площади треугольника по длинам сторон a, b, c.

Для решения используется формула Герона , где p = (a+b+c)/2 –полупериметр треугольника.

Исходные данные должны удовлетворять основному соотношению для сторон треугольника: длина каждой стороны должна быть меньше суммы длин двух других сторон. Имея возможность в одном условном операторе записывать достаточно сложные логические выражения, можно сразу «отфильтровать» все варианты неверных исходных данных:

```
PROGRAM Geron;
VAR A,B,C,P,S:Real;
BEGIN
WRITELN('Vveditedlinctorontreugolnica:');
WRITE('a='); Readln (A);
WRITE('b='); Readln (B);
WRITE('c='); Readln (C);
IF(A>0) AND (B>0) AND (C>0) AND (A+B>C)
AND (B+C>A) AND (A+C>B)
THENBEGIN
        P := (A+B+C)/2;
        S := SQRT(P*(P-A)*(P-B)*(P-C));
WRITELN ('Ploshad = ',S:6:2)
END
ELSE WRITELN ('Hevernieischodniedannie');
READLN;
END.
```

Результат выполнения:

```
a=5
b=7
c=8
Ploshad =  17.32
```

ОПЕРАТОР ВЫБОРА

Оператор выбора (варианта) используется в тех случаях, когда в зависимости от значения какого-либо выражения необходимо выполнить один из нескольких последовательных операторов. Оператор выбора относится к сложным и имеет следующую форму записи:

CASE выражение **OF**
константа 1: оператор 1;
константа 2: оператор 2;
…
константа n: оператор n
ELSE оператор;
END

Здесь CASE (в случае), OF (из), END (конец), ELSE (иначе) - служебные слова.

Оператор выбора действует следующим образом. Если значение выражения равно одной из констант, то выполняется соответствующий ей оператор. Затем управление передается за пределы оператора выбора. Если значение выражения не совпадает ни с одной константой, то управление передается за пределы группы.

Выражение может быть любым стандартным типом, кроме действительного (REAL). В соответствии с этим и константа не может быть действительного типа. Тип константы должен совпадать с типом выражения. Оператор выбора может использоваться в полной и неполной форме.

ПРИМЕР ЗАПИСИ ОПЕРАТОРА ВЫБОРА:

CASE K+1 OF 5 : Y:=SQR(X); 11 : Y :=SQRT(X); 4 : Z := 4*(A-B); 7 : WRITELN (A,B) END	Если значение K+1 будет равно 5, то выполнится оператор присваивания Y:=SQR(X)и управление будет передано на оператор, расположенный после слова END. Аналогично, если значение K+1 будет равно 11, 4 или 7, то выполнится один соответствующий оператор и управление будет передано за пределы оператора выбора. Переменная K должна быть объявлена как переменная целого типа. Кроме того, K, X, A, B должны получить значения до выполнения оператора CASE.

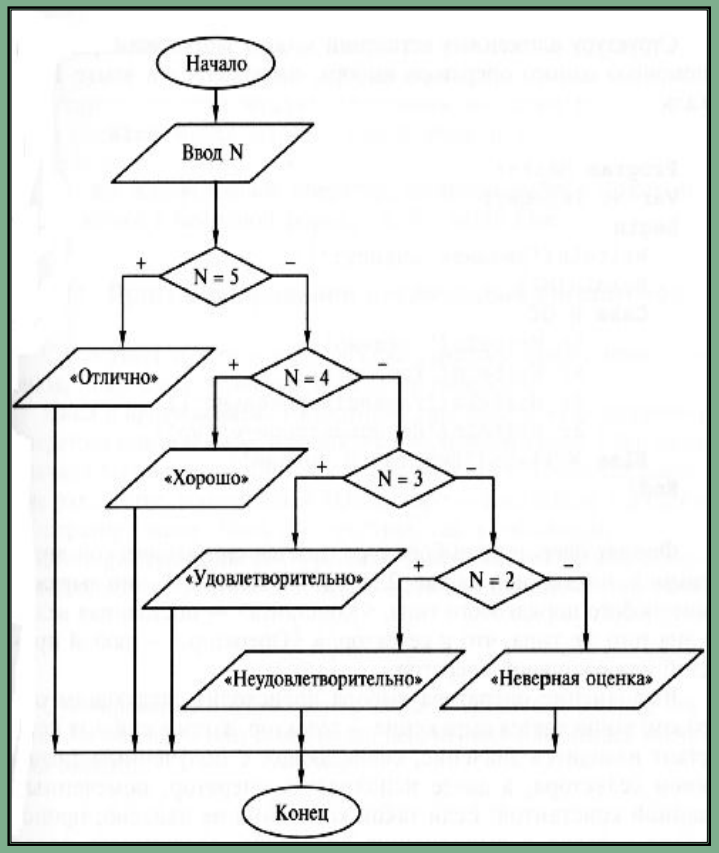
В ОПЕРАТОРЕ ВЫБОРА В КАЧЕСТВЕ КОНСТАНТЫ ДОПУСКАЕТСЯ ИСПОЛЬЗОВАНИЕ:

- символьные константы: ‘+’, ‘B’, ‘␣’, ‘/’ и т. д.;
- диапазон: 2..7;
- последовательность: 2,4,6,8, и т.д.

CASE S OF ’+’,’-’,’*’,’/’: P := 1; ’A’, ’B’ : P := 2; ’.’: P :=3 END	Переменная S должна быть объявлена в разделе описаний как символьная. Если значение S будет один из знаков ’+’,’-’,’*’,’/’ , то переменная P получит значение 1. Если значением S будет символ ‘A’ или ‘B’, то P получит значение 2. Если значением S будет знак точки ‘.’, то переменная P будет присвоено значение 3.

ПРИМЕР. Требуется составить программу перевода пятибалльной оценки в ее наименовании: 5 – отлично, 4 – хорошо, 3 – удовлетворительно, 2 – неудовлетворительно.

Алгоритм имеет структуру вложенных ветвлений.




```
PROGRAM Marks;
VAR N : Integer;
BEGIN
WRITELN('Vvedite ochenky');
READLN (N);
CASE N OF
        5: Writeln ('Otlichno');
        4: Writeln ('Xorocho');
        3: Writeln ('Ydov');
        2: Writeln ('Neydov');
ELSE WRITELN ('Net takoi
ochenki');
END;
READLN;
END.
```

ОПЕРАТОР ЦИКЛА С ПРЕДВАРИТЕЛЬНЫМ УСЛОВИЕМ

WHILE логическое выражение **DO**
BEGIN
операторы циклической части программы
END

Здесь WHILE (пока) и DO (выполнить) – служебные слова.
Оператор цикла действует следующим образом. Предварительно проверяется значение логического выражения. Пока оно истинно, выполняются операторы циклической части. Как только оно становится ложным, происходит выход из цикла. Если с самого начала значение логического выражения ложно, то операторы циклической части не выполнятся ни разу.

Операторы циклической части, заключенные в операторные скобки BEGIN–END, представляют собой составной оператор. Если в циклической части стоит всего один оператор, то операторные скобки BEGIN–END можно не указывать и оператор цикла принимает вид:
WHILE логическое выражение **DO** оператор

ПРИМЕР. Написать программу, подсчитывающую количество четных и нечетных цифр в числе.

Описание переменных: even - количество четных цифр; odd - количество нечетных цифр.

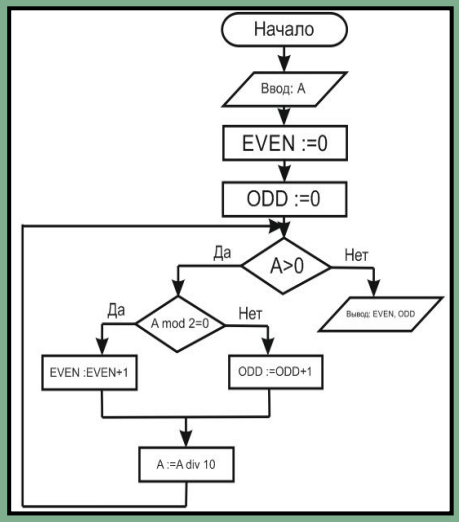
Метод решения задачи:

Если число делится без остатка на 2, значит последняя цифра четная (увеличиваем переменную even). Иначе - нечетная (тогда odd + 1).

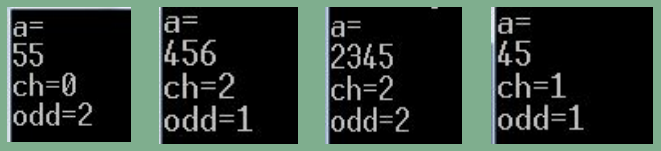
Избавляемся от младшего разряда в числе: операция div на 10.

```
PROGRAM W_01;
VAR
        A: INTEGER;
        EVEN,ODD: BYTE;
BEGIN
WRITELN (’Vveditechiclo’);
READLN(A);
        EVEN := 0;
        ODD := 0;
        WHILE A > 0 DO
BEGIN
IF (A MOD 2) = 0
THEN EVEN := EVEN + 1
        ELSEODD := ODD + 1;
        A := A DIV 10;
        END;
        WRITELN('EVEN: ', EVEN);
        WRITELN('ODD: ', ODD);
READLN;
END.
```

Алгоритм выполнения:



Результат выполнения:



ОПЕРАТОР ЦИКЛА С ПОСЛЕДУЮЩИМ УСЛОВИЕМ

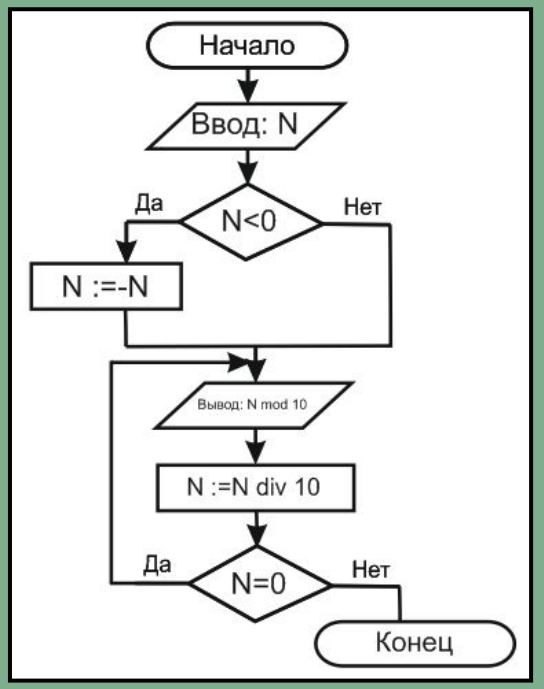
Оператор цикла имеет вид:
REPEAT
операторы циклической части программы
UNTIL логическое выражение
Здесь REPEAT (повторить) и UNTIL (до тех пор) – служебные слова. Оператор цикла с последующим условием действует следующим образом. Операторы циклической части выполняются повторно (по крайней мере, один раз) до тех пор, пока значение логического выражения ложно. Условием прекращения циклических вычислений является истинное значение логического выражения. Итак, сначала выполняется циклическая часть, а затем проверяется условие.

Нижняя граница операторов циклической части четко обозначена, словом UNTIL, поэтому нет необходимости заключать операторы циклической части в скобки BEGIN–END. В то же время и дополнительное наличие операторных скобок не является ошибкой.

ПРИМЕР. Определить из каких цифр состоит число

```
PROGRAM R_01;
VAR
   N:INTEGER;
BEGIN
   WRITE('TYPE INTEGER: ');
READLN(N);
IF N<0 THEN
       N:=-N; // УНИЧТОЖЕНИЕ
ЗНАКА ЧИСЛА
REPEAT
       WRITELN(N MOD 10); //
ВЫВОД ПОСЛЕДНЕЙ ЦИФРЫ
ЧИСЛА
       N:= N DIV 10; // УДАЛЕНИЕ
ПОСЛЕДНЕЙ ЦИФРЫ ЧИСЛА
UNTIL N=0;
READLN
END.
```

Алгоритм выполнения:



ОПЕРАТОР ЦИКЛА С ПАРАМЕТРОМ

– является неопределенным – это означает, что при последнем выполнении циклической части значение <Параметра цикла> равно <Выражению 2>, а после ухода за Где <параметр цикла> : := <имя простой переменной порядкового типа>. Выполнение оператора FOR (в первом варианте To) происходит по следующей схеме:
Вычисляются значения <Выражение 1> и <Выражение 2>, причем только один раз при входе в цикл.
Параметру цикла присваивается значение <Выражение 1>.
Значение параметра цикла сравнивается с значением <Выражение 2>.
Если параметр цикла меньше или равен этому значению, то выполняется тело цикла, в противном случае выполнение цикла заканчивается.

Значение параметра цикла изменяется на следующее значение в его типе (для целых чисел – увеличивается на единицу) и происходит возврат к п. 3 данной схемы.
Оператор цикла FOR объединяет в себе действия, которые при использовании цикла WHILE выполняют различные операторы: присваивание параметру начального значения, сравнение с конечным значением, изменение значения на следующее.

Работая с оператором FOR, следует учитывать следующие правила:
параметр цикла не может иметь тип REAL;
в теле цикла нельзя изменять переменную – параметр цикла;
при выходе из цикла значение переменной – параметра цикла пределы цикла значение <Параметра цикла> теряется.
Оператор цикла имеет вид:

```
FOR i := m1 TO m2 DO
BEGIN
операторы циклической части программы
END
```

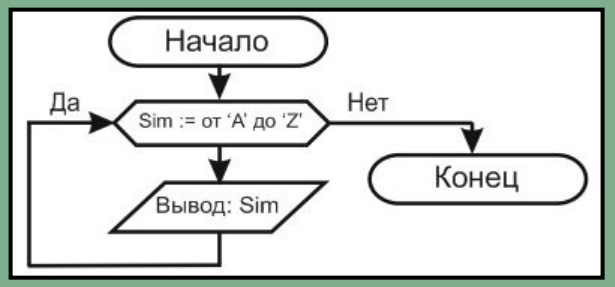
Здесь FOR (для), TO (до), DO (выполнить) – служебные слова; i – параметр цикла; m1, m2 – начальное и конечное значение параметра цикла.
Циклическая часть программы выполняется повторно для каждого значения параметра цикла i от его начального значения m1 до конечного значения m2 включительно.

В качестве параметра цикла может быть только переменная, в качестве m1 и m2 могут быть выражения, за исключением действительного типа (REAL).
Чаще всего параметр цикла i используют как переменную целого типа, а шаг его изменения равен +1 или -1. Если значение параметра цикла возрастает, то шаг его изменения +1. Если значение параметра цикла уменьшается, то шаг изменения -1 и в операторе цикла FOR вместо служебного слова TO записывается служебное слово DOWNTO.

ПРИМЕР. Напечатать все буквы латинского алфавита.

```
PROGRAM ALF;
VAR SIM : CHAR;
BEGIN
WRITELN ('LAT ALFAVIT');
FOR SIM := 'A' TO 'Z' DO
WRITE (' ',SIM);
READLN;
END.
```

Алгоритм выполнения:



Результат выполнения: