

Повторение пройденного



Общий вид программы

```
program <имя программы>;  
const <список постоянных значений>;  
var <описание используемых  
переменных>;  
begin  
    <оператор 1>;  
    <оператор 2>;  
    ...  
    <оператор n>;  
end.
```

Операторы - языковые конструкции для записи действия, выполняемого над данными в процессе решения задачи.

Простые типы данных

Название	Обозначение	Допустимые значения	Область памяти
Целочисленный	integer	- 32 768... 32 768	2 байта со знаком
Вещественный	real	$\pm(2.9 * 10^{-39} \dots 1.7 * 10^{+38})$	6 байтов
Символьный	char	Произвольный символ алфавита	1 байт
Строковый	string	Последовательность символов длиной меньше 255	1 байт на символ
Логический	boolean	True и False	1 байт

Вывод данных

Вывод данных из оперативной памяти на экран монитора:

write (<выражение 1> ,< выражение 2> , ...,< выражение **N**>);

Выражения - символьные, числовые, логические, в том числе переменные и константы

Пример:

```
s:=15; a:=0; b:=7;
```

```
write ('s=', s, s+10, a+b, ' ', 'Введите любое число');
```

На экране появится:

```
s=15257 Введите любое число_
```

Вывод с новой строки

writeln (<выражение 1> ,< выражение 2> , ...,< выражение **N**>)

курсор перейдет на новую строчку

Формат вывода

Формат вывода позволяет установить количество позиций на экране, занимаемых выводимой величиной.

write(s:x:y)

x - общее количество позиций, отводимых под число
(под точку тоже отводится место);

y - количество позиций в дробной части числа.

a:=10; b:=3; writeln(a/b);

3,33333333333333333333

Оператор вывода	Результат выполнения оператора
write ('s=', s:2:0);	s=15
write ('s=', s:4:1);	s=15.0
write ('s=', s:5:1);	s= 15.0

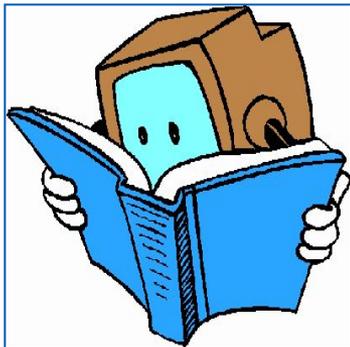
Ввод данных с клавиатуры

Ввод в оперативную память значений переменных :

```
readln (<имя переменной1>, ..., <имя переменной N> );
```

список ввода

ReadLn читается "рид'лайн", переводится "читай строку". Он приказывает компьютеру остановиться и ждать, когда человек введет с клавиатуры определенную информацию, после чего продолжить работу.



```
VAR a,b : Integer;  
BEGIN  
    ReadLn (a,b);  
    WriteLn (a+b);  
END.
```

ReadLn (a,b) будет ждать ввода двух целых чисел.

Стандартные функции языка Паскаль:

Функция	Назначение	Пример	Результат
abs (x)	Модуль x	Abs (-20)	20
sqr (x)	Квадрат x	Sqr (5)	25
sqrt (x)	Квадратный корень из x	Sqrt (25)	5
frac (x)	Дробная часть x	Frac (23.192)	0.192
round (x)	Округление x до ближайшего целого	Round (5.8)	6
int (x)	Целая часть x	Int (3.98), trunc(3.98)	3. 0 3
pi	Число Пи	Pi	3.1415...

Случайные числа

Для генерации **целого** числа

C := random(50); будет генерировать случайное число в диапазоне **0 <= C < 50**

C := random(15) - 10; будет генерировать случайное число в диапазоне **-10 <= C < 5**

Для генерации **вещественного** случайного числа

C := random; будет генерировать случайное число в диапазоне **0 <= C < 1**

C := random + 5; будет генерировать случайное число в диапазоне **5 <= C < 6**

C := random * 10 - 5; будет генерировать случайные числа в диапазоне **-5 <= C < 5**

Целочисленный тип данных

Операции над целыми числами в языке Паскаль:

Операция	Обозначение	Тип результата
Сложение	+	integer
Вычитание	-	integer
Умножение	*	integer
Получение целого частного	div	integer
Получение целого остатка деления	mod	Integer
Деление	/	Real

Символьный тип данных

Для ввода в программу символьных переменных необходимо указать для них символьный тип данных **char**.

```
VAR a : char;  
BEGIN  
    a := '0';  
    WriteLn(a);  
END.
```

Символы – это все буквы и значки, которые есть на клавиатуре. Количество символов равно 256, каждому из которых соответствует числовой код согласно кодовой таблице.

Строковый тип данных

Значением строковой величины (тип **string**) является произвольная последовательность символов, заключенная в апострофы.



В тексте программы символ мы задаем, заключая его в апострофы, когда вводим с клавиатуры - без

```
VAR a : String;  
BEGIN  
    a:='Привет всем!';  
    WriteLn (a);  
END.
```

```
VAR a : String;  
BEGIN  
    WriteLn ('Введите какое-нибудь слово');  
    ReadLn (a);  
    WriteLn ('Вы ввели слово ',a)  
END.
```

Написать программу:

1. Написать программу нахождения суммы и произведения цифр четырехзначного числа, введенного с клавиатуры и составить новое число, поменяв старший и младший разряд в введенном числе.

Общий вид условного оператора

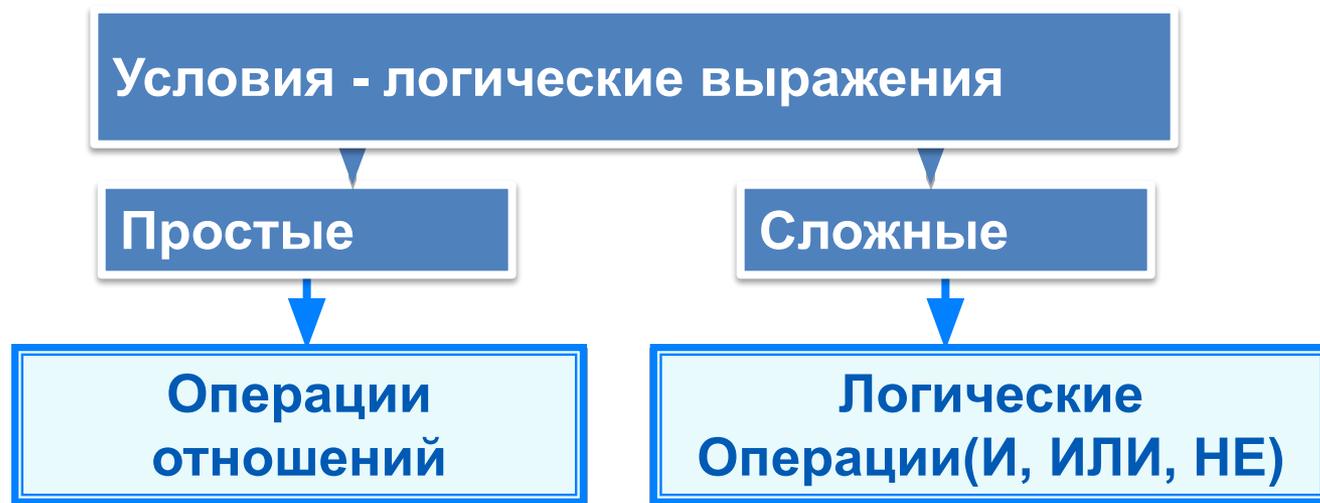
Полная форма условного оператора:

```
if <условие> then <оператор_1>  
    else <оператор_2>;
```

Неполная форма условного оператора:

```
if <условие> then <оператор>;
```

! Перед **else** знак «;» не ставится.



Составной оператор

В условном операторе и после **then**, и после **else** можно использовать **только один оператор**.

Если в условном операторе после **then** или после **else** нужно выполнить **несколько операторов**, то используют **составной оператор** – конструкцию вида:

begin

<последовательность операторов>

end

**Написать программу нахождения
корней квадратного уравнения**

```
program n_11;  
  var a, b, c: real;  
      d: real;  
      x, x1, x2: real;
```

```
begin
```

```
  writeln ('Решение квадратного уравнения');  
  write ('Введите коэффициенты a, b, c >>');  
  readln (a, b, c);  
  d:=b*b-4*a*c;  
  if d<0 then writeln ('Корней нет');  
  if d=0 then
```

```
    begin
```

```
      x:=-b/2/a;  
      writeln ('Корень уравнения x=', x:9:3)
```

```
    end;
```

```
  if d>0 then
```

```
    begin
```

```
      x1:=(-b+sqrt(d))/2/a;  
      x2:=(-b-sqrt(d))/2/a;  
      writeln ('Корни уравнения:');  
      writeln ('x1=', x1:9:3);  
      writeln ('x2=', x2:9:3)
```

```
    end;
```

```
end.
```

Вложенные ветвления

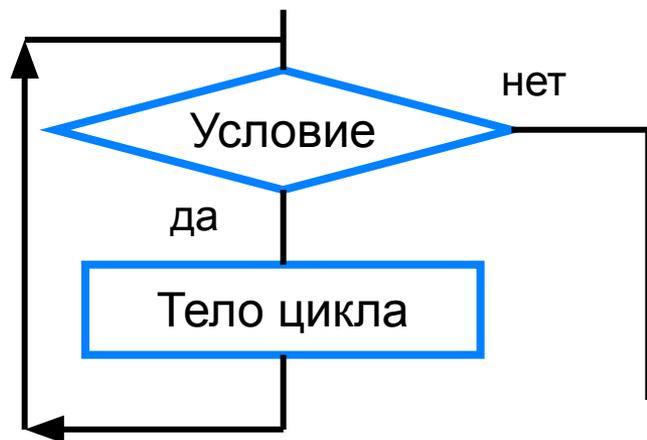
Возможна следующая конструкция:

```
if <условие1> then if <условие2>  
                                then <оператор1>  
                                else <оператор2>  
                                else <оператор3>;
```

! **else** всегда относится к ближайшему оператору **if**

**Написать программу развития событий у камня на дороге,
по которой едет Илья Муромец**

Программирование циклов с заданным условием продолжения работы



Общий вид оператора:

```
while <условие> do  
  <оператор>;
```

Конструкция читается:

Пока условие верно, **делай**
(выполняется тело цикла)

Правила:

- Так как условие проверяется в начале цикла, то цикл может не выполниться ни разу.
- Для того, чтобы цикл не стал бесконечным, в теле цикла должны изменяться величины, входящие в логическое выражение.
- Если в теле цикла несколько операторов, то ставятся `begin ... end`;

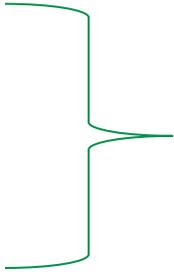
```
while <условие> do  
  begin  
    <оператор 1>;  
    ...  
    <оператор N>;  
  end;
```

Пример

При помощи цикла напечатать на экране:

Начало счета 3 5 7 9 Конец счета

```
VAR f : Integer;  
BEGIN  
  Write('Начало счета ');  
  f:=3;  
  while f<=9 do  
    begin  
      Write(f, ' ');  
    end;  
  write('Конец счета');  
END.
```



Составной оператор
(операторные скобки
begin..end)

Написать программу:

1. Напечатать $1\ 2\ 3\ 4\ \dots\ 99\ 100$
2. Напечатать $1\ 3\ 5\ \dots\ 99\ 100\ 98\ \dots\ 2\ 0$
3. "Таблицы Брадиса"- вычислить и напечатать **с 6 десятичными знаками** квадраты чисел

0.000

0.001

0.002

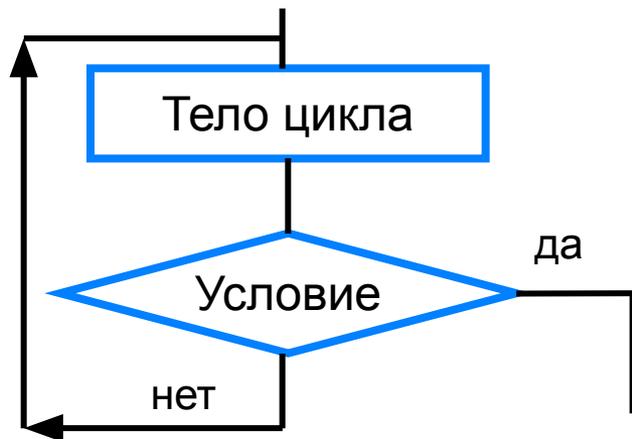
0.003

...

0.999

1.000.

Программирование циклов с заданным условием окончания работы



repeat

<оператор 1>;

<оператор 2>;

...

<оператор N>;

until <условие>;

Конструкция repeat...until (читается – ри’пит...ан’тил...) переводится:

повторяй ... до тех пор, пока условие неверно.

Правила:

- Тело цикла всегда выполняется хотя бы один раз, так как условие стоит в конце цикла.
- В этом операторе цикла можно написать много операторов в теле цикла

Пример:

При помощи цикла напечатать на экране:

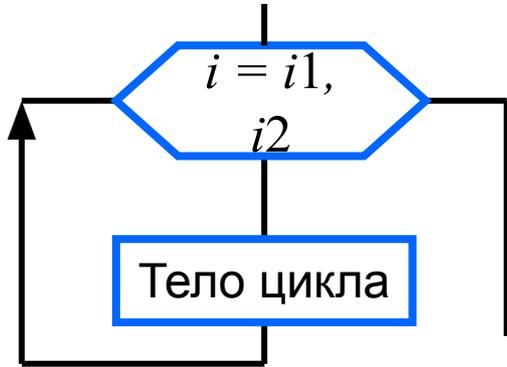
Начало счета 3 5 7 9 Конец счета

```
VAR f : Integer;  
BEGIN  
  Write('Начало счета ');  
  f:=3;  
  repeat  
    Write(f,' ');  
    f:=f+2;  
  until f>9;  
  Write(' Конец счета');  
END.
```

Написать программы:

1. Для $x=2700, 900, 300, 100 \dots$ и т.д. вычислять и печатать $y=x/4 + 20$ и $z=2y+0.23$ до тех пор, пока $y*z$ не станет меньше $1/x$.
2. Усложним уже сделанную задачу. Компьютер предлагает человеку ввести слово, после чего распечатывает это слово, снабдив его восклицательным знаком. Затем снова предлагает ввести слово и так до тех пор, пока человек не введет слово "Хватит". Распечатав его с восклицательным знаком, компьютер отвечает "Хватит так хватит" и заканчивает работу. Пусть компьютер перед распечаткой каждого слова ставит его порядковый номер.
3. Написать программу, которая при введении с клавиатуры чисел, если они не равны 0, суммирует их. Если введен 0, суммирование заканчивается, и на экране отображается результат.

Программирование циклов с заданным числом повторений



Общий вид:

for **i** := **i1** **to** **i2** **do** оператор;
(если $i1 < i2$)

или

for **i** := **i1** **downto** **i2** **do** оператор;
(если $i1 > i2$)

Конструкция **for i:=1 to 200 do** читается так:

Для i , изменяющегося от 1 до 200, делай оператор, стоящий после слова do.

переменная i называется параметром или счетчиком цикла.

Программирование циклов с заданным числом повторений

Правила:

- Параметром цикла может быть только целая переменная.
- Параметр цикла не может явным образом меняться в теле цикла.

```
for i:=1 to 10 do  
    begin  
        a:=a+2;  
        i:=i+1; {неверно}  
    end;
```

- Шаг цикла может быть равен только +1 (цикл с to) или -1 (цикл с downto).
- В соответствии с общим видом тело цикла должно состоять из одного оператора, если операторов несколько, используем составной оператор и операторные скобки begin..end;

Примеры

Пример 1. 200 раз напечатать слово ФУТБОЛ

```
VAR i : Integer;  
BEGIN  
    for i:=1 to 200 do WriteLn('ФУТБОЛ');  
END.
```

Пример 2: Пусть компьютер печатает такую информацию:

10 ФУТБОЛ 11 ФУТБОЛ 12 ФУТБОЛ 200
ФУТБОЛ

```
VAR i : Integer;  
BEGIN  
for i:=10 to 200 do  
    begin  
        Write(i);  
        Write(' ФУТБОЛ ');  
    end;  
END.
```

Написать программу:

- Написать программу с помощью оператора for:

Прямой счет: -5 -4 -3 -2 -1 0 1 2 3 4 5 Обратный счет: 5 4 3 2
1 0 -1 -2 -3 -4 -5

- Во дворце 40 залов. Известны длина, ширина и высота каждого зала. Вычислить площадь пола и объем каждого зала.

Написать программу:

Теперь создадим более дружелюбный интерфейс, для чего, кроме всего прочего, дадим возможность пользователю самому задавать число залов во дворце:

```
VAR i, dlina, shirina, visota, N, S, V : Integer;  
BEGIN  
  WriteLn('Введите число залов');  
  ReadLn (N);           {N - число залов}  
  for i:=1 to N do  
    begin  
      WriteLn('Введите длину, ширину и высоту зала');  
      ReadLn (dlina, shirina, visota);  
      S:=dlina*shirina;  
      V:=S*visota;  
      WriteLn('Площадь пола=',S,' Объем зала=',V)  
    end;  
END.
```

Задачи

1. Определите значение переменной P после выполнения следующего фрагмента программы:

```
P:=1; i:=3;  
while i <= 9 do begin  
  P := P * (i div 3);  
  i := i + 1;  
end;
```

2. Определите, что будет напечатано в результате работы следующего фрагмента программы:

```
var n, s: integer;  
begin  
  n := 0;  
  s := 0;  
  while s <= 256 do begin  
    s := s + 25;  
    n := n + 1  
  end;  
  write(n)  
end.
```

3. Определите, что будет напечатано в результате работы следующего фрагмента программы:

```
var k, s: integer;  
begin  
  s:=0;  
  k:=0;  
  while k < 30 do begin  
    k:=k+3;  
    s:=s+k;  
  end;  
  write(s);  
end.
```

4. Определите, что будет напечатано в результате работы следующего фрагмента программы:

```
var k, s: integer;  
begin  
  s:=1;  
  k:=0;  
  while k < 13 do begin  
    s:=s+2*k;  
    k:=k+4;  
  end;  
  write(s+k);  
end.
```

5. Укажите наибольшее из таких чисел ,
при вводе которых алгоритм печатает сначала 3,
а потом 7.

```
var x, L, M: integer;  
begin  
    readln(x);  
    L:=0; M:=0;  
    while x > 0 do  
        begin  
            L:= L + 1;  
            M:= M + x mod 10;  
            x:= x div 10;  
        end;  
    writeln(L); write(M);  
end.
```

Написать программы:

1. Даны натуральные числа от 35 до 87. Найти и напечатать те из них, которые при делении на 7 дают остаток 1, 2 или 5.
2. Сколько нечетных среди n случайно введенных.
3. Вычислить средний балл учеников вашего класса по физике.
4. Вычислить произведение N произвольных чисел.
5. Напишите программу, которая получает два целых числа A и B ($0 < A < B$) и выводит квадраты всех натуральных чисел в интервале от A до B .
6. Вводится целое число. Если оно является положительным, то прибавить к нему 255; если отрицательным, то вычесть из него 45; если нулевым, то заменить его на 67. Вывести полученное число.