

ЦИКЛЫ В ПРОГРАММИРОВАНИИ НА ЯЗЫКЕ ПАСКАЛЬ

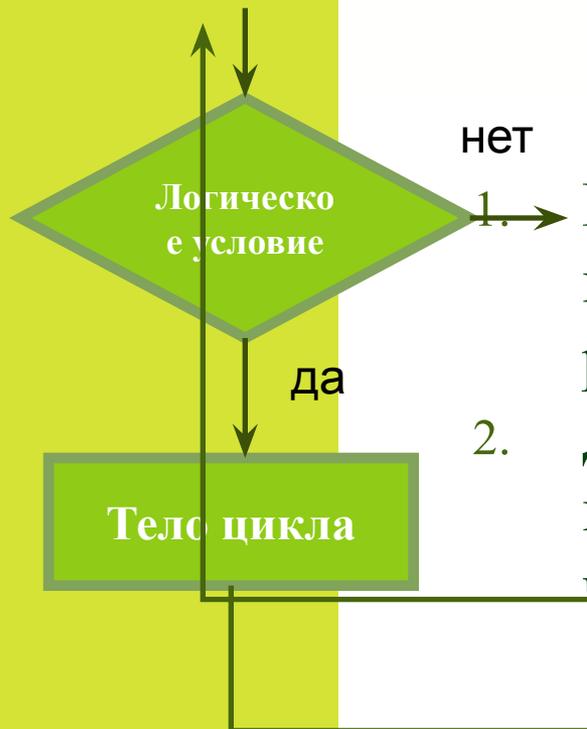
ЦИКЛ С ПРЕДУСЛОВИЕМ

ЦИКЛ С ПРЕДУСЛОВИЕМ

While <логическое условие> **do** <тело цикла>

Пока логическое условие истинно, выполняются операторы, составляющие тело цикла.

Цикл может не выполняться ни разу



Необходимые условия:

1. В цикле обязательно должна использоваться переменная, позволяющая циклу завершить работу.
2. До входа в цикл все переменные, используемые в теле цикла, должны иметь численное значение.

ЦИКЛ С ПРЕДУСЛОВИЕМ

Рассмотрим три задачи:

1. Найти сумму чисел до заданного N .
2. Найти $N!$
3. Найти X^N

ЦИКЛ С ПРЕДУСЛОВИЕМ

При подсчете компьютер выполняет следующие действия:

- 1.Находит результат для двух первых чисел;
- 2.Затем к результату прибавляет следующее число;
- 3.Повторяет эти операции, пока не достигнет последнего (N -ого числа).

Следовательно, компьютер должен выполнить N -шагов для достижения результата, то есть 1, 2, 3, ..., N -ный шаг.

Обозначим через i очередной шаг, который необходимо выполнить компьютеру.

Рассмотрим подробнее задачу на суммирование.

Пусть $N=3$

Сумма чисел до N : $S = 1+2+3$.

То есть 1 шаг $S_1=1$;

2 шаг $S_2=S_1+2$;

3 шаг $S_3=S_2+3$.

Как видно из примера, на каждом шаге к результату предыдущего шага прибавляется следующее значение.

Для подсчета используется несколько раз формула

$$S = S + \langle \text{следующее число ряда} \rangle$$

Надо заметить, значение $\langle \text{следующего числа ряда} \rangle$ в данном случае совпадает с номером шага. Перепишем формулу с учетом этого обстоятельства: $S = S + i$

Задача на суммирование.

Распишем суммирование по полученной формуле:

1 шаг $i=1$ $S=S+1$;

2 шаг $i=2$ $S=S+2$, где **S** – значение, полученное на 1 шаге

3 шаг $i=3$ $S=S+3$, где **S** – значение, полученное на 2 шаге

Остается неопределенным S на первом шаге.

Нам надо подобрать такое число, которое при сложении не изменит результат. Это число – 0.

Из рассмотренного примера видно, что при изменении N будет изменяться число шагов, но решение задачи изменяться не будет.

Следовательно, результат суммирования N чисел для данного примера можно посчитать выполняя N раз формулу $S = S + i$

Рассмотрим подробнее задачу нахождения N!

Пусть N=3

Обозначим через F результат нахождения факториала.

N! при N=3 : $F = 1 * 2 * 3$.

То есть 1 шаг $F_1 = 1$;

2 шаг $F_2 = F_1 * 2$;

3 шаг $F_3 = F_2 * 3$.

Как видно из примера, на каждом шаге результат предыдущего шага умножается на следующее значение.

Для подсчета используется несколько раз формула

$$F = F * \langle \text{следующее число ряда} \rangle$$

Надо заметить, значение $\langle \text{следующего числа ряда} \rangle$ совпадает с номером шага. Перепишем формулу с учетом этого обстоятельства:

$$F = F * i$$

Задача на нахождение N!

Распишем задачу по полученной формуле:

1 шаг $i=1$ $F=F*1$;

2 шаг $i=2$ $F=F*2$, где **F** – значение, полученное на 1 шаге

3 шаг $i=3$ $F=F*3$, где **F** – значение, полученное на 2 шаге

Остается неопределенным **F** на первом шаге.

Нам надо подобрать такое число, которое при умножении не изменит результат. Это число – 1.

Из рассмотренного примера видно, что при изменении N будет изменяться число шагов, но решение задачи изменяться не будет.

**Следовательно, результат поиска N! можно посчитать
выполняя N раз формулу**

$$F = F * i$$

Рассмотрим подробнее задачу нахождения X^N

Пусть $N=3$

Обозначим St результат возведения числа X в степень N .

при $N=3$: $St = X * X * X$.

То есть 1 шаг $St1=X$;

2 шаг $St2=St1 * X$;

3 шаг $St3=St2 * X$.

Как видно из примера, на каждом шаге результат предыдущего шага умножается на X .

Для подсчета используется несколько раз формула

$$St = St * X$$

Задача на возведение числа X в степень N

Распишем задачу по полученной формуле:

1 шаг $i=1$ $St=St*X$;

2 шаг $i=2$ $St=St*X$, где St – значение, полученное на 1 шаге

3 шаг $i=3$ $St=St*X$, где St – значение, полученное на 2 шаге

Остается неопределенным St на первом шаге.

Нам надо подобрать такое число, которое при умножении не изменит результат. Это число – 1.

Из рассмотренного примера видно, что при изменении N будет изменяться число шагов, но решение задачи изменяться не будет.

**Следовательно, результат поиска X^N можно посчитать
выполняя N раз формулу**

$$St = St * X$$

ЦИКЛ С ПРЕДУСЛОВИЕМ

Рассмотрим условия создания цикла:

1. В цикле обязательно должна использоваться переменная, позволяющая циклу завершить работу.

Во всех рассмотренных задачах расчет формулы выполняется пока не будет достигнуто значение N .

Исходя из этого, можно сформулировать логическое условие $i \leq N$, где i – счетчик шагов выполнения цикла.

Переменная i будет изменяться в цикле и позволит выполнить выход из цикла.

ЦИКЛ С ПРЕДУСЛОВИЕМ

Рассмотрим условия создания цикла:

2. До входа в цикл все переменные, используемые в теле цикла, должны иметь численное значение.

Во всех рассмотренных задачах расчет формулы на первом шаге выполняется при заданном значении переменной, не изменяющей выполняемого арифметического действия.

Исходя из этого, можно определить значения переменных до входа в цикл:

$i:=1$; - определяется первый шаг

Суммирование – $S:=0$;

Нахождение факториала и степени числа – $F:=1$; и $St:=1$;

Значение N , а также X (для задачи возведение в степень) задаются вначале программы как данные, вводимые пользователем.

ЦИКЛ С ПРЕДУСЛОВИЕМ НА ЯЗЫКЕ ПАСКАЛЬ ДЛЯ ТРЕХ РАССМОТРЕННЫХ ЗАДАЧ

Подсчет суммы чисел до
некоторого N:

```
S:=0;  
i:=1;  
While i<= N do begin  
    S:=S+i;  
    i:=i+1;  
end;
```

Подсчет произведения чисел до
некоторого N (N!):

```
F:=1;  
i:=1;  
While i<= N do begin  
    F:=F*i;  
    i:=i+1;  
end;
```

Подсчет X^N

```
St:=1;  
i:=1;  
While i<= N do begin  
    St:=St*X;  
    i:=i+1;  
end;
```

```
C:\ Turbo Pascal 7.0
введите количество элементов
n=10

входные данные
s= 0.00   i= 1

работа цикла
сумма 1 элементов равна 1.00
изменение количества шагов i = 2
сумма 2 элементов равна 3.00
изменение количества шагов i = 3
сумма 3 элементов равна 6.00
изменение количества шагов i = 4
сумма 4 элементов равна 10.00
изменение количества шагов i = 5
сумма 5 элементов равна 15.00
изменение количества шагов i = 6
сумма 6 элементов равна 21.00
изменение количества шагов i = 7
сумма 7 элементов равна 28.00
изменение количества шагов i = 8
сумма 8 элементов равна 36.00
изменение количества шагов i = 9
сумма 9 элементов равна 45.00
изменение количества шагов i = 10
сумма 10 элементов равна 55.00
изменение количества шагов i = 11

вывод результата
сумма ряда равна 55.00
```

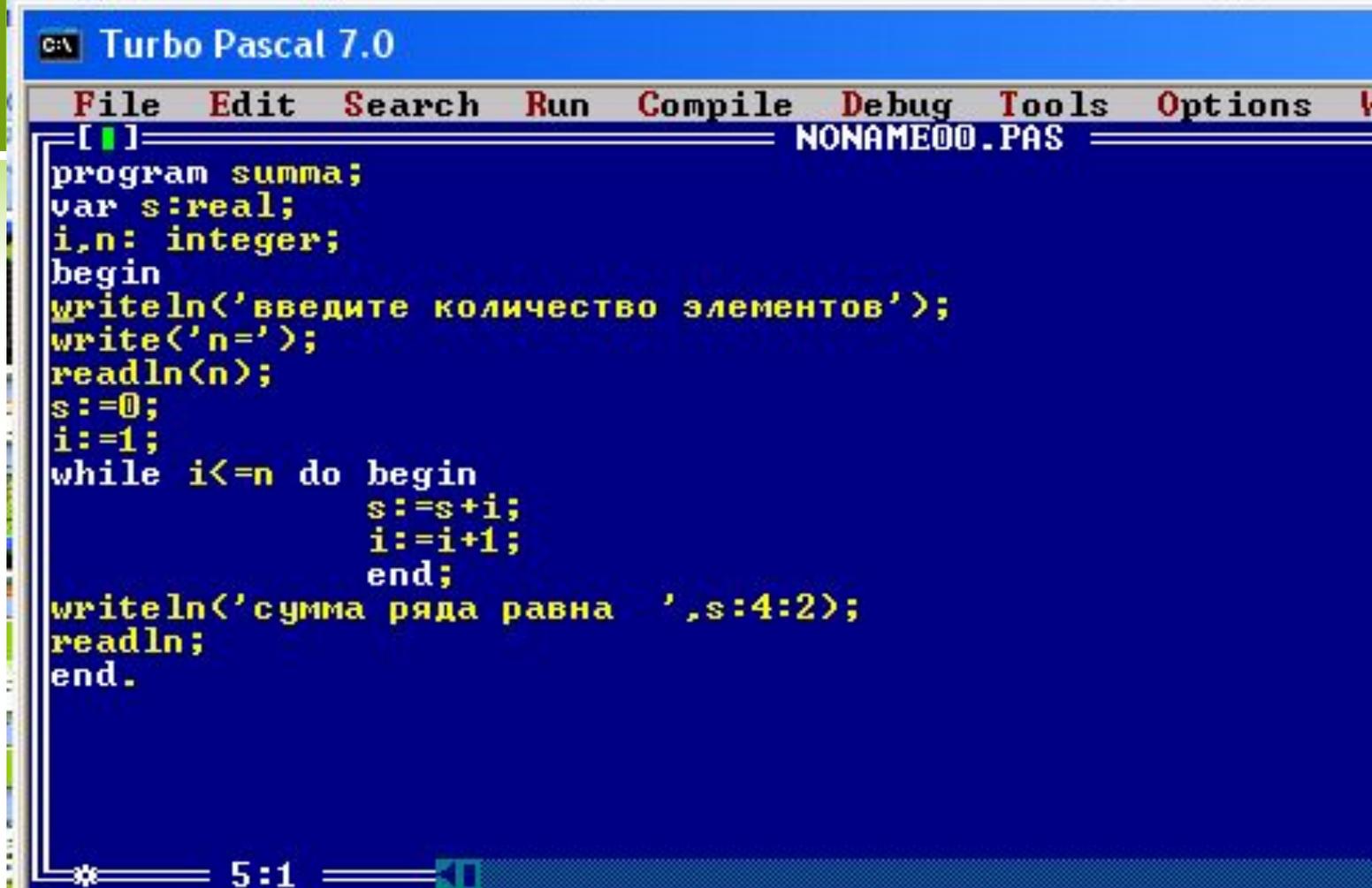
Вводится только количество элементов

Входные данные цикла

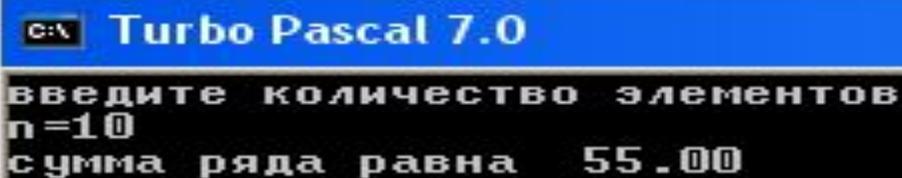
Подсчет промежуточных результатов внутри цикла. На каждом шаге к результату предыдущего шага прибавляется новый элемент

Выходные данные цикла

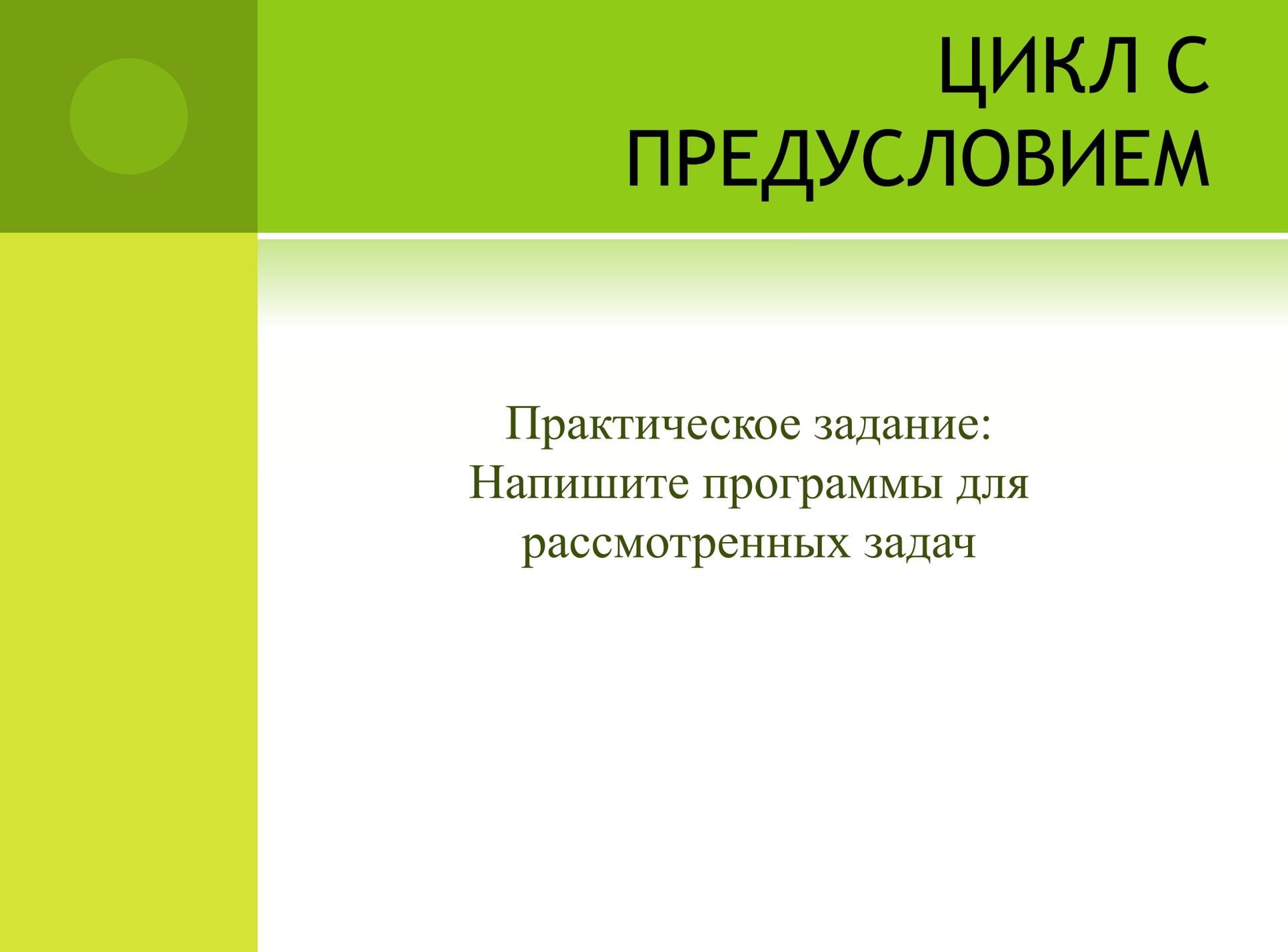
ПРИМЕР ПРОГРАММЫ И ВИД ОКНА ПОЛЬЗОВАТЕЛЯ



```
c:\ Turbo Pascal 7.0
File Edit Search Run Compile Debug Tools Options
[ ] NONAME00.PAS
program summa;
var s:real;
i,n: integer;
begin
writeln('введите количество элементов');
write('n=');
readln(n);
s:=0;
i:=1;
while i<=n do begin
                s:=s+i;
                i:=i+1;
            end;
writeln('сумма ряда равна ',s:4:2);
readln;
end.
* 5:1
```



```
c:\ Turbo Pascal 7.0
введите количество элементов
n=10
сумма ряда равна 55.00
```



ЦИКЛ С ПРЕДУСЛОВИЕМ

Практическое задание:
Напишите программы для
рассмотренных задач

РАССМОТРИМ НЕКОТОРЫЕ ПРИМЕРЫ СОСТАВЛЕНИЯ ФОРМУЛ



Подсчитать сумму N членов ряда:
 $1/2+1/4+1/6+1/8+\dots$

Распишем получение суммы данного ряда по шагам:

$$S_1=0+1/2$$

$$S_2=S_1+1/4$$

$$S_3=S_2+1/6$$

То есть, как рассматривалось выше, сумма ряда считается по формуле: $S=S + \langle \text{число ряда по порядку} \rangle$.

Основой является составление формулы для i -го члена ряда.



Подсчитать сумму N членов ряда:
 $1/2+1/4+1/6+1/8+\dots$

Рассмотрим изменение знаменателя слагаемых: 2, 4, 6, 8, ..., его можно представить как $2*1, 2*2, 2*3, 2*4, \dots$. Следовательно,

1 шаг $S=S+1/2*1$

2 шаг $S=S+1/2*2$

3 шаг $S=S+1/2*3$

Из примера видно, что числитель слагаемых не изменяется, а в знаменателе один из множителей постоянный, а второй совпадает с номером шага.

Следовательно, можно вывести формулу для расчета слагаемых ряда : $1/(2*i)$.



Подсчитать сумму N членов ряда:
 $1/2+1/4+1/6+1/8+\dots$

В рассмотренном выше цикле на суммирование, можно заменить слагаемое i в формуле $S:=S+i$; на $1/(2*i)$ и использовать цикл для решения данной задачи, получаем формулу: $S:=S+ 1/(2*i)$;

Подобный подход можно использовать во всех, задачах на суммирование в цикле.



Подсчитать произведение N членов ряда:
 $2*4*6*8*10*...$

Распишем получение произведения данного ряда по шагам:

$$F1=1*2$$

$$F2=F1*4$$

$$F3=F2*6$$

То есть, как рассматривалось выше, произведение ряда считается по формуле: $F=F * \langle \text{число ряда по порядку} \rangle$.



Подсчитать произведение N членов ряда:
 $2*4*6*8*10*...$

Рассмотрим изменение множителей: 2, 4, 6, 8,..., представим как $2*1, 2*2, 2*3, 2*4, ...$. Следовательно,

1 шаг $F = F*(2*1)$

2 шаг $F = F*(2*2)$

3 шаг $F = F*(2*3)$

Из примера видно, что один из множителей постоянный, а второй совпадает с номером шага.

Следовательно, можно вывести формулу для расчета множителей ряда : $(2*i)$.



Подсчитать произведение N членов ряда:
 $2*4*6*8*10*...$

В рассмотренном выше цикле на нахождение факториала можно заменить множитель i в формуле $F:=F*i$; на $(2*i)$ и использовать цикл для решения данной задачи, получаем формулу: $F:=F*(2*i)$;

Подобный подход можно использовать во всех, задачах на произведение.

ЦИКЛ С ПРЕДУСЛОВИЕМ

Задания для самостоятельного выполнения

1. Напишите программу вычисления суммы N слагаемых ряда: $1 + 1/2 + 1/3 + 1/4 + 1/5 + \dots$
2. Напишите программу вычисления произведения N чисел ряда: $1 * 1/2 * 1/3 * 1/4 * 1/5 * \dots$

Учитывайте: $1 = 1/1$

ЦИКЛ С ПОСТУСЛОВИЕМ

ЦИКЛ С ПОСТУСЛОВИЕМ

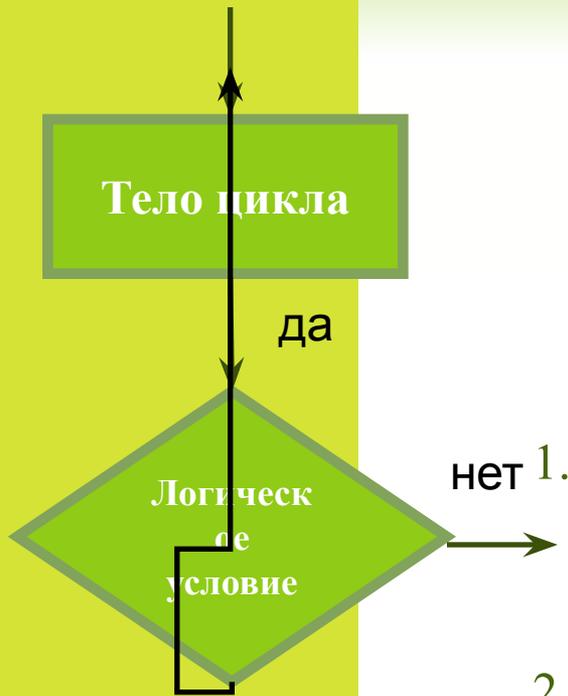
Repeat <тело цикла> Until <условие> ;

Операторы, составляющие тело цикла, выполняются, до тех пор, пока логическое условие не станет истинно.

Цикл выполняется один раз обязательно

Необходимые условия:

1. В цикле обязательно должна использоваться переменная, позволяющая циклу завершить работу.
2. До входа в цикл все переменные, используемые в теле цикла, должны иметь численное значение.



ЦИКЛ С ПОСТУСЛОВИЕМ

Используется,
если число повторений заранее
неизвестно, а определяется по
ходу реализации циклического
процесса.

ЦИКЛ С ПОСТУСЛОВИЕМ

Рассмотрим применение цикла к задачам:

1. Найти сумму чисел до заданного N .
2. Найти $N!$
3. Найти X^N

ЦИКЛ С ПОСТУСЛОВИЕМ НА ЯЗЫКЕ ПАСКАЛЬ ДЛЯ ТРЕХ РАССМОТРЕННЫХ ЗАДАЧ

Подсчет суммы чисел до
некоторого N:

```
S:=0;  
i:=1;  
Repeat S:=S+i;  
       i:=i+1;  
Until i > N ;
```

Подсчет произведения чисел до
некоторого N (N!):

```
F:=1;  
i:=1;  
Repeat F:=F*i;  
       i:=i+1;  
Until i > N ;
```

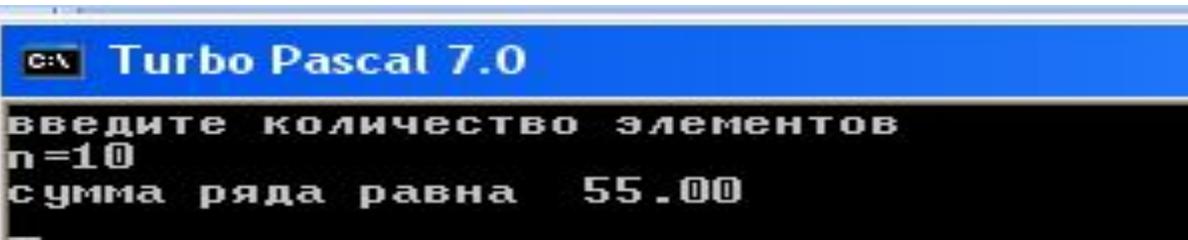
Подсчет X^N

```
St:=1;  
i:=1;  
Repeat St:=St*X;  
       i:=i+1;  
Until i > N ;
```

ПРИМЕР ПРОГРАММЫ И ВИД ОКНА ПОЛЬЗОВАТЕЛЯ



```
c:\ Turbo Pascal 7.0
File Edit Search Run Compile Debug To
[ ] ZIKL.PAS
program summa;
var s:real;
i,n: integer;
begin
writeln('введите количество элементов');
write('n=');
readln(n);
s:=0;
i:=1;
Repeat s:=s+i;
       i:=i+1;
Until i>n;_
writeln('сумма ряда равна ',s:4:2);
readln;
end.
```



```
c:\ Turbo Pascal 7.0
введите количество элементов
n=10
сумма ряда равна 55.00
```

ПРИМЕРЫ ПРОГРАММ

ПРИМЕРЫ ПРОГРАММ

Паскаль-рулетка

Предлагается угадать целое число от 1 до 10

Пусть программа «загадает» число, а пользователь введет предполагаемое значение.

Если число угадано, программа поздравит победителя, а если нет — попросит его повторить попытку еще раз.

Вначале игроку назначается 10 призовых баллов.

Каждая безуспешная попытка снижает призовые баллы.

ПРИМЕРЫ ПРОГРАММ

Паскаль-рулетка.

В этом примере число повторений цикла заранее неизвестно, поэтому лучше использовать одну из разновидностей цикла с проверкой условия.

ПРИМЕРЫ ПРОГРАММ

Описание алгоритма:

- выбрать случайное целое число от 1 до 10;
- вывести приглашение на ввод целого значения;
- если введенное число меньше задуманного, сообщить об этом игроку, иначе сообщить ему о том, что введенное число больше задуманного;
- повторять ввод целого значения до тех пор, пока число не будет угадано;
- вывести поздравление победителю и сообщить ему о набранном числе баллов;
- завершить работу.



ПРИМЕРЫ ПРОГРАММ

Не исключена возможность того, что
число будет угадано сразу.

В этом случае уже не надо выводить подсказку игроку,
поэтому следует использовать цикл с предусловием

ПРОГРАММА

```
[ ] ZIKL.PAS 2
program roulette;
var number, n, bonus: byte;
begin
bonus:=10;
randomize;
number:=random(11);
writeln('Задумано целое число от 0 до 10 Угадайте!');
Write('Введите число ');
readln(n);
while n<> number do begin
  dec(bonus);
  writeln('Вы не угадали!');
  if n<number then writeln('Ваше число меньше задуманного')
    else writeln('Ваше число больше задуманного');
  writeln('Попытайтесь еще раз!');
  readln(n);
end;
writeln('Поздравляю Вы угадали и набрали ', bonus, ' очков!');
readln;
end.
```

КОММЕНТАРИЙ

В этой программе используются новые операторы :

Randomize — начальная установка специальной процедуры — «генератора» случайных чисел **Random(n)** выдающей случайные целые числа от 0 до $n - 1$

Dec(bonus) — вызов процедуры, уменьшающей на единицу значение переменной **bonus**.

Новый тип данных:

Byte — целочисленный, диапазон $0...255$, размер 1 байт.

ОКНО ПОЛЬЗОВАТЕЛЯ

Задумано целое число от 0 до 10 Угадайте!
Введите число 5

Задумано целое число от 0 до 10 Угадайте!
Введите число 5

Вы не угадали!
Ваше число меньше задуманного
Попытайтесь еще раз!

6
Вы не угадали!
Ваше число меньше задуманного
Попытайтесь еще раз!

8
Вы не угадали!
Ваше число меньше задуманного
Попытайтесь еще раз!

9
Вы не угадали!
Ваше число меньше задуманного
Попытайтесь еще раз!

10
Поздравляю Вы угадали и набрали 6 очков!

ОКНО ПОЛЬЗОВАТЕЛЯ

```
Задумано целое число от 0 до 10 Угадайте!  
Введите число 9  
Вы не угадали!  
Ваше число больше задуманного  
Попытайтесь еще раз!  
7  
Вы не угадали!  
Ваше число больше задуманного  
Попытайтесь еще раз!  
5  
Вы не угадали!  
Ваше число больше задуманного  
Попытайтесь еще раз!  
3  
Поздравляю Вы угадали и набрали 7 очков!
```

ПРИМЕРЫ ПРОГРАММ

Пробуем разбогатеть

Предлагается открыть счет в банке.

Банк ежегодно начисляет определенный процент от вклада – учетную ставку процента.

Будем считать, что учетная ставка процента не зависит от времени и величины вклада.

Программа рассчитывает величину вклада и выводит эту величину для каждого года до тех пор, пока величина вклада не удвоится.

ПРИМЕРЫ ПРОГРАММ

Описание алгоритма:

- ввести первоначальную величину вклада, учетную ставку процента и год помещения денег в банк;
- рассчитать новую величину вклада;
- вывести год и величину вклада в данном году;
- повторять шаги 2 и 3 до тех пор, пока величина вклада не удвоится;
- завершить работу.

ПРИМЕРЫ ПРОГРАММ

Деньги в банке должны находиться хотя бы один год, чтобы на них были начислены проценты.

В этом случае цикл должен быть выполнен по крайней мере один раз, поэтому следует использовать цикл с постусловием

ПРОГРАММА

```
program Roskefeler;  
var vklad, balance, rate, interest : real;  
    year: word;  
begin  
write('Введите год помещения денег в банк: ');  
readln(year);  
write('Введите величину вклада: ');  
readln(vklad);  
write('Введите ставку процента (0.0 - 1.0): ');  
readln(rate);  
balance:=vklad;  
writeln('Год      Вклад');  
writeln('=====');  
repeat  
    interest:= rate*vklad;  
    vklad:=vklad+interest;  
    inc(year);  
    writeln(year:4,'      ', vklad:6:2);  
until vklad > 2*balance;  
readln;  
end.
```

КОММЕНТАРИЙ

В этой программе используются новые операторы :

Inc(year) - вызов процедуры, увеличивающей на единицу значение переменной year.

Новый тип данных:

Word – целочисленный, диапазон 0...65535, размер 2 байта.

ОКНО ПОЛЬЗОВАТЕЛЯ

```
Введите год помещения денег в банк: 2013
Введите величину вклада: 50000
Введите ставку процента <0.0 - 1.0>: 0.18
```

```
Год      Вклад
=====
2014     59000.00
2015     69620.00
2016     82151.60
2017     96938.89
2018    114387.89
```

```
Введите год помещения денег в банк: 2013
Введите величину вклада: 10000
Введите ставку процента <0.0 - 1.0>: 0.14
```

```
Год      Вклад
=====
2014     11400.00
2015     12996.00
2016     14815.44
2017     16889.60
2018     19254.15
2019     21949.73
```

ЦИКЛЫ

С ПРЕДУСЛОВИЕМ ПОСТУСЛОВИЕМ

И

Какую разновидность цикла лучше выбрать в каждом конкретном случае?

- Используйте цикл `repeat`, если необходимо, чтобы тело цикла выполнялось *по крайней мере один раз*.
- Используйте цикл `while`, если хотите, чтобы проверка была произведена *прежде*, чем будет выполняться тело цикла.

ЦИКЛ С ПАРАМЕТРОМ

Цикл с параметром имеет два варианта записи:

Шаг +1

For $i := n$ to k do оператор;

i – параметр цикла (счетчик),
 n – начальное значение параметра,
 k – конечное значение параметра

Переменная i увеличивается на единицу от начального значения n до конечного значения k при каждом новом значении i исполняется оператор внутри цикла.



**Для правильного выполнения этого вида
цикла должно выполняться условие $n < k$**

Если $n > k$, то цикл не выполняется ни разу.

Если $n = k$, то цикл выполняется один раз

ЦИКЛ С ПАРАМЕТРОМ

2. Шаг – 1

For $i := n$ downto k do оператор;

i – параметр цикла (счетчик),
 n – начальное значение параметра,
 k – конечное значение параметра

Переменная i уменьшается на единицу от начального значения n до конечного значения k при каждом новом значении i исполняется оператор внутри цикла.



**Для правильного выполнения этого вида
цикла должно выполняться условие $n > k$**

Если $n < k$, то цикл не выполняется ни разу.

Если $n = k$, то цикл выполняется один раз



Если тело цикла состоит из нескольких операторов, то они заключаются в операторные скобки `begin ... end`;

```
For i := n to k do begin  
    оператор 1;  
    оператор 2;  
    ...  
    оператор N;  
end;
```

ЦИКЛ С ПРЕДУСЛОВИЕМ

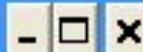
Рассмотрим три задачи:

1. Найти сумму чисел до заданного N .
2. Найти $N!$
3. Найти X^N

ЦИКЛ С ПАРАМЕТРОМ

```
program summa;  
var i,n,s: integer;  
begin  
writeln('введите количество слагаемых');  
readln(n);  
s:=0;  
for i:=1 to n do s:=s+i;  
writeln('сумма равна ',s);  
readln;  
end.
```

Turbo Pascal 7.0



```
Turbo Pascal Version 7.0 Copyright (c) 1983,92 Borland International  
введите количество слагаемых  
25  
сумма равна 325
```

ЦИКЛ С ПАРАМЕТРОМ

```
program factorial;
var i,n,f: integer;
begin
writeln ('введите число для расчета факториала');
readln(n);
f:=1;
for i:=1 to n do f:=f*i;
writeln(n,'! = ',f);
readln;
end.
```

```
введите число для расчета факториала
10
10! = 24320
```

ЦИКЛ С ПАРАМЕТРОМ

```
program stepenx;  
var i,x,n,st: integer;  
begin  
writeln('введите основание степени');  
readln(x);  
writeln('введите степень');  
readln(n);  
st:=1;  
for i:=1 to n do st:=st*x;  
writeln(x,' в степени ', n, ' = ', st);  
readln;  
end.
```

```
введите основание степени  
2  
введите степень  
10  
2 в степени 10 = 1024
```

ЦИКЛ С ПАРАМЕТРОМ

Задача

212° по Фаренгейту

У писателя-фантаста Рэя Бредбери есть роман «451° по Фаренгейту». В подзаголовке к роману сказано, что «451° по Фаренгейту — температура, при которой горит бумага».

СПРАВКА

Температурная шкала Фаренгейта была предложена немецким физиком Габриэлем Фаренгейтом и используется в настоящее время в ряде англоязычных стран.

В этой шкале температура замерзания воды при стандартном атмосферном давлении равна 32°F , а температура кипения составляет 212°F .

В более привычной для нас шкале Цельсия аналогичными «опорными» точками являются соответственно 0°C и 100°C .

ЦИКЛ С ПАРАМЕТРОМ

Формула для пересчета температуры имеет вид:

$$t_f = 9 / 5 * t_c + 32,$$

где

t_f — температура по Фаренгейту,

t_c — температура по Цельсию.



ЦИКЛ С ПАРАМЕТРОМ

Следующая программа предназначена для вывода таблицы соответствия между температурными шкалами Цельсия и Фаренгейта в интервале температур от точки замерзания воды до точки ее кипения.

ПРОГРАММА

C:\ Turbo Pascal 7.0

File Edit Search Run Compile Debug Tools Options Window

[] ZIKL.PAS

```
program Celsius_to_Fahrenheit;
var i, n, Celsius, Fahrenheit : integer;
begin
writeln('Таблица соответствия между температурными шкалами');
writeln('          Цельсия и Фаренгейта');
writeln;
For i:=0 to 20 do begin
  Celsius:=5*i;
  Fahrenheit:=32+Celsius*9 div 5 ;
  writeln('C=', Celsius:4, ' F=', Fahrenheit:4 );
  writeln;
end;
readln;
end.
```

ОКНО ПОЛЬЗОВАТЕЛЯ

Таблица соответствия между температурными шкалами
Цельсия и Фаренгейта

C=	0	F=	32
C=	5	F=	41
C=	10	F=	50
C=	15	F=	59
C=	20	F=	68
C=	25	F=	77
C=	30	F=	86
C=	35	F=	95
C=	40	F=	104
C=	45	F=	113
C=	50	F=	122
C=	55	F=	131
C=	60	F=	140
C=	65	F=	149
C=	70	F=	158
C=	75	F=	167
C=	80	F=	176
C=	85	F=	185
C=	90	F=	194
C=	95	F=	203
C=	100	F=	212

