

Указания к работе

1. Изучите теоретический материал.
2. Письменно в тетради выполните самостоятельную работу (фотографии прислать 06.05.2020 до 18:00).
3. Выполните домашнее задание.

Теоретический материал

Сегодня мы рассмотрим, как на языке Паскаль программируются **циклы с заданным числом повторений**. Особенность цикла с заданным числом повторений состоит в том, что прежде чем приступить к выполнению цикла в программе уже вычислено, сколько раз выполнится тело цикла. Цикл с заданным числом повторений является частным случаем **цикла с параметром**.

В блоке цикла с параметром задаётся название параметра – **i**, его начальное и конечное значение, а также шаг его изменения – **h** не равный нулю. Цикл с параметром работает так, что перед первым выполнением цикла вычисляются начальное и конечное значение параметра. После этого параметру присваивается его начальное значение и проверяется, в случае **h > 0**, не больше ли значение параметра – **i** заданного конечного значения. В случае **h < 0**, не меньше ли значение параметра – **i** заданного конечного значения. Если условие проверки истинно, то выполняется тело цикла, после чего значение параметра – **i** изменяется на значение шага – **h** и снова проверяется не больше или не меньше ли оно заданного конечного значения. Цикл продолжает выполняться до тех пор, пока значение его параметра – **i** не станет, при **h > 0** – больше, а при **h < 0** – меньше заданного конечного значения.



Важно при этом, что значение параметра нельзя изменять в теле цикла. Т.е. ещё до начала первого выполнения тела цикла можно вычислить количество его повторений. Для этого достаточно вычислить разность конечного и начального значения параметра, после чего разделить эту разность на значение шага и к полученному числу прибавить 1.

Однако в языке Паскаль нет оператора, предназначенного именно для записи описанного нами цикла с параметром. Он реализуется с помощью цикла с предусловием.

```
i:=nz;  
while i<=kz do  
begin  
  ...  
  i:=i+h;  
end;
```

Но есть оператор цикла с параметром, но при этом значение шага изменения параметра всегда равно единице, поэтому его указывать не требуется.

Рассмотрим, как же записывается оператор цикла с параметром на языке Паскаль. Его запись начинается со служебного слова **for**. После него, через пробел, следует название параметра – это переменная целочисленного типа, объявленная в разделе описания переменных. После этого следует знак присваивания – «:=». После него следует выражение, которым задаётся начальное значение параметра. Через пробел после него следует одно из двух служебных слов, **to** или **downto**. Если это слово **to** – значение параметра при каждом выполнении тела цикла будет увеличиваться на 1. Если это слово **downto** – значение параметра при каждом выполнении тела цикла будет уменьшаться на 1. Через пробел после одного из описанных служебных слов следует выражение, задающее конечное значение параметра. После него, через пробел, следует служебное слово **do**. Со следующей строки, с отступом в один пробел от слова **for**, следует тело цикла. Если тело цикла состоит из нескольких операторов, то оно замыкается между служебными словами **begin** и **end**. После слова **end** ставится точка с запятой.

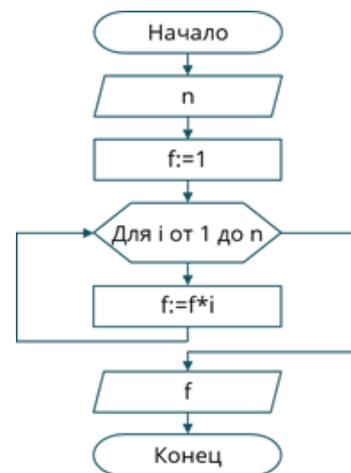
```
for <п-п>:=<N3> to <K3> do  
begin  
  <оператор 1>;  
  <оператор 2>;  
  ...  
  <оператор n>;  
end;
```

Задача. Вычислим факториал натурального числа **n**. **n** вводится с клавиатуры и находится в диапазоне от 1 до 12. Вспомним, что $n! = 1 \times 2 \times \dots \times n$. Таким образом, для того, чтобы вычислить значение **n!** нам достаточно перемножить все целые числа от 1 до **n**.

Словесное описание алгоритма

Сначала пользователь задаёт с клавиатуры значение **n**. Нам понадобится переменная, которая будет хранить значение вычисляемого произведения. Назовём её **f** и присвоим ей значение 1, так как ни одного множителя мы ещё не учли. Далее будет следовать блок цикла с параметром. Назовём параметр **i**, и он будет изменяться от 1 до **n**. Так как шаг изменения параметра равен 1, его указывать не нужно. **f** будет значением очередного множителя нашего произведения **f**. Поэтому тело цикла будет содержать всего один оператор присваивания $f = f \times i$. После цикла нам остаётся лишь вывести значение вычисленного произведения **f**. На этом наш алгоритм завершит свою работу.

Напишем программу `factorial`. Для работы программы нам потребуется три переменные: **n**, **i** и **f**. Так как **n** находится на промежутке от 1 до 12, а **i** изменяется на промежутке от 1 до **n**, объявим их принадлежащими к типу **byte**. **f** – это произведение значений **i** поэтому она будет принадлежать к целочисленному типу **integer**. Запишем логические скобки. Тело программы будет начинаться с оператора `writeln`, который выводит на экран сообщение о том, что это программа, вычисляющая значения **n!**. Далее будет следовать оператор `write`, выводящий запрос на ввод **n** и оператор `readln`, считывающий значение **n** введённое пользователем. Теперь запишем оператор присваивания переменной **f** значения 1. Далее запишем цикл с параметром `for i:=1 to n do`. Так, как тело цикла будет содержать всего один оператор, логические скобки указывать не требуется. Запишем в цикле оператор присваивания $f:=f*i$. После цикла запишем оператор вывода на экран сообщение о том, что $n! = f$.



```
program factorial;
var
  n, i: byte;
  f: integer;
begin
  writeln ('Программа вычисления n!');
  write ('n=');
  readln (n);
  f:=1;
  for i:=1 to n do
    f:=f*i;
  writeln (n, '!=', f);
end.
```

Различные варианты программирования циклического алгоритма

Ссылка на видео: <https://youtu.be/BAS7hnfvTs0>

Самостоятельная работа: решить задачи
Учебник, задание 10, 11, 13 (в) на стр. 143

Домашнее задание: выучить §3.5.3-3.5.4