

**13.05.22г.**

**Класс: 8**

**Предмет: Информатика**

**Тема:** Программирование циклов с условием, с заданным числом повторений  
**Практическая работа №9** «Разработка и выполнение программ в среде программирования для решения прикладных задач»

### **Ход урока**

#### **1. Просмотрите видеоурок**

<https://www.youtube.com/watch?v=dL4AEzoLWqM>

#### **2. Познакомьтесь с теоретическим материалом учебника : параграф 3.5**

#### **Программирование циклических алгоритмов**

#### **Ключевые слова:**

- while (цикл-ПОКА)
- repeat (цикл-ДО)
- for (цикл с параметром)

#### **3.5.1. Программирование циклов с заданным условием продолжения работы**

**Цикл с заданным условием продолжения работы (цикл-ПОКА)** программируется в языке Паскаль с помощью оператора **while**. Общий вид оператора:

**while <условие> do <оператор>**

Здесь:

<условие> — логическое выражение; пока оно истинно, выполняется тело цикла;

<оператор> — простой или составной оператор, с помощью которого записано тело цикла.

Запишем на языке Паскаль рассмотренный в п. 2.4.3 (пример 14) алгоритм получения частного **q** и остатка **r** от деления натурального числа **x** на натуральное число **y** без использования операции деления.

```

program n_14;
var x, y, q, r: integer;
begin
  writeln ('Частное и остаток');
  write ('Введите делимое x>>');
  readln (x);
  write ('Введите делитель y>>');
  read (y);
  r:=x;
  q:=0;
  while r>=x do
  begin
    r:=r-y;
    q:=q+1
  end;
  writeln ('Частное q=', q);
  writeln ('Остаток r=', r)
end.

```

Каким будет результат выполнения программы при  $x = -10$  и  $y = 3$ ? Как вы можете объяснить этот результат?

### 3.5.2. Программирование циклов с заданным условием окончания работы

**Цикл с заданным условием окончания работы (цикл-ДО)** программируется в языке Паскаль с помощью оператора **repeat**. Общий вид оператора:

**repeat <оператор1; оператор2; ...; > until <условие>**

Здесь:

<оператор1>; <оператор2>; ... — операторы, образующие тело цикла;

<условие> — логическое выражение; если оно ложно, то выполняется тело цикла.

Запишем на языке Паскаль рассмотренный в п. 2.4.3 (пример 17) алгоритм решения задачи о графике тренировок спортсмена.

```

program n_15;
var i: integer; x: real;
begin
  writeln ('График тренировок');
  i:=1;
  x:=10;
  repeat
    i:=i+1;
    x:=x+0.1*x;
  until x>=25;
  writeln ('Начиная с ', i, '-го дня спортсмен будет'
           ' пробегать 25 км')
end.

```

### 3.5.3. Программирование циклов с заданным числом повторений

**Цикл с заданным числом повторений (цикл-ДЛЯ)** программируется в языке Паскаль с помощью оператора **for**. Его общий вид:

**for <параметр>:=<начальное\_значение> **to** <конечное\_значение> **do** <оператор>**

Здесь:

<параметр> — переменная целого типа;

<начальное\_значение> и <конечное\_значение> — выражения того же типа, что и параметр, вычисляемые перед началом цикла;

<оператор> — простой или составной оператор — тело цикла.

При выполнении этого оператора после каждого выполнения тела цикла происходит увеличение на единицу параметра цикла; условием выхода из цикла является превышение параметром конечного значения.

Запишем на языке Паскаль рассмотренный в п. 2.4.3 (пример 19) алгоритм вычисления степени с натуральным показателем **n** для любого вещественного числа **a**.

```
program n_16;
  var i, n: integer; a, y: real;
begin
  writeln ('Возведение в степень');
  write ('Введите основание a>>');
  readln (a);
  write ('Введите показатель n>>');
  readln (n);
  y:=1;
  for i:=1 to n do y:=y*a;
  writeln ('y=', y)
end.
```

### 3.5.4. Различные варианты программирования циклического алгоритма

Особенностью программирования является то, что для решения одной и той же задачи могут быть созданы разные программы. Вы могли убедиться в этом, программируя ветвления. Рассмотрим пример, показывающий, что и циклический алгоритм может быть запрограммирован разными способами.

**Пример.** Напишем программу, в которой осуществляется ввод целых чисел (ввод осуществляется до тех пор, пока не будет введён ноль) и подсчёт количества введённых положительных и отрицательных чисел.

Так как здесь в явном виде задано условие окончания работы, то воспользуемся оператором **repeat**.

```

program n_17;
  var n, k1, k2: integer;
begin
  k1:=0;
  k2:=0;
  repeat
    write ('Введите целое число>>');
    readln (n);
    if n>0 then k1:=k1+1;
    if n<0 then k2:=k2+1;
  until n=0;
  writeln ('Введено:');
  writeln ('положительных чисел - ', k1);
  writeln ('отрицательных чисел - ', k2)
end.

```

Имеющееся условие окончания работы можно достаточно просто преобразовать в условие продолжения работы — работа продолжается, пока  $n \neq 0$ . И мы можем воспользоваться оператором **while**:

```

program n_18;
  var n, k1, k2: integer;
begin
  k1:=0;
  k2:=0;
  while n<>0 do
begin
  writeln ('Введите целое число>>');
  read (n);
  if n>0 then k1:=k1+1;
  if n<0 then k2:=k2+1;
end;
  writeln ('Введено:');
  writeln ('положительных - ', k1);
  writeln ('отрицательных - ', k2)
end.

```

В рассмотренном примере число повторений тела цикла заранее не известно, и поэтому оператор **for** здесь применить нельзя. Если число повторений тела цикла известно, то лучше воспользоваться оператором **for**. Вместе с тем любая задача, в которой число повторений тела цикла определено заранее, может быть запрограммирована с помощью любого из трёх рассмотренных выше циклов.

## САМОЕ ГЛАВНОЕ

В языке Паскаль имеются три вида операторов цикла: **while (цикл-ПОКА)**, **repeat (цикл-ДО)**, **for (цикл с параметром)**. Если число повторений тела цикла известно, то лучше воспользоваться оператором **for**; в остальных случаях используются операторы **while** и **repeat**.

## **Выполните практическую работу №9**

Выполнить практическую работу в виде интерактивных упражнений.

Прислать скриншоты выполненных упражнений, каждого в отдельности.

<https://learningapps.org/803807>

<https://learningapps.org/3133782>

<https://learningapps.org/2169382>

<https://learningapps.org/4195453>

**4. Домашнее задание:** повторить теоретический материал параграф 3.5

Практическая работа №9 (скриншоты)

**Не забывайте писать название темы после даты!**

**Выполненные работы** присылайте на адрес электронной почты [isytnikova@mail.ru](mailto:isytnikova@mail.ru)