L#5

Основы алгоритмизации и программирования. Введение

Педагогическое образование, 3 семестр

Mayer Svetlana Fyodorovna

Встроенные циклы

- Задача: Посчитать значение функции $z(x,y) = x^y$ для каждого x в интервале [2;8] и y в интервале [2;5].
- Пример выполнения:

z(x,y) = 2^2 = 4

z(x,y) = 2^3 = 8

z(x,y) = 2^4 = 16

z(x,y) = 3^2 = 9

z(x,y) = 3^3 = 27

…

```
begin
for var x:=2 to 8 do
    for var y:= 2 to 4 do
    begin
      var z:=power(x,y);
      writelnformat('z(x,y) = {0}^{1} = {2}',x,y,z);
    end;
end.
```

- Необходимо создать два цикла (встроенных цикла): один цикл в другом. Переменную x следует менять во внешнем цикле; переменную y следует менять во внутреннем цикле.

Встроенные циклы

- Задача: Отобразить строки со следующими последовательностями. Использовать *встроенные циклы.*
- The resulting example:

```
9 9 9 9 9
8 8 8 8 8
7 7 7 7 7
6 6 6 6 6
5 5 5 5 5
```

```
1  begin
2    for var x := 9 downto 5 do
3    begin
4      for var y := 1 to 5 do
5      begin
6        print(x)
7      end;
8      println;
9    end
10 end.
```

- Внешний цикл *for* «проходится» по колонкам, внутренний цикл формирует элементы строки …

Логический тип (задача поиска числа в последовательности)

- Задача: Вводятся значения n и k. Вводятся n чисел последовательности. Программа должна определить есть ли число k среди введенной последовательности.
- Решение. Объявим логическую переменную exists. Инициализируем ее значением false. В цикле поменяем ее значение в true в том случае если число k совпало с числом последовательности.
- Пример выполнения:

Сколько чисел? >>>5

Какое число будем искать? >>>3

Введите последовательность

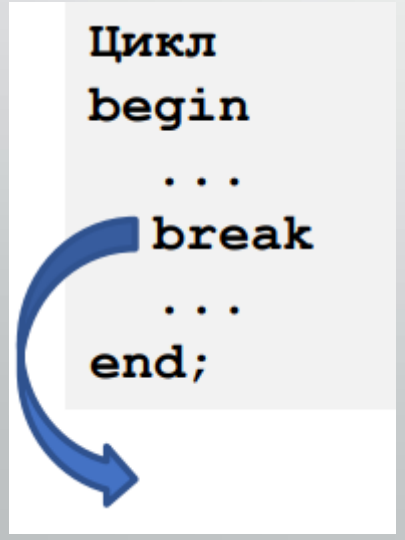
1 3 6 4 7

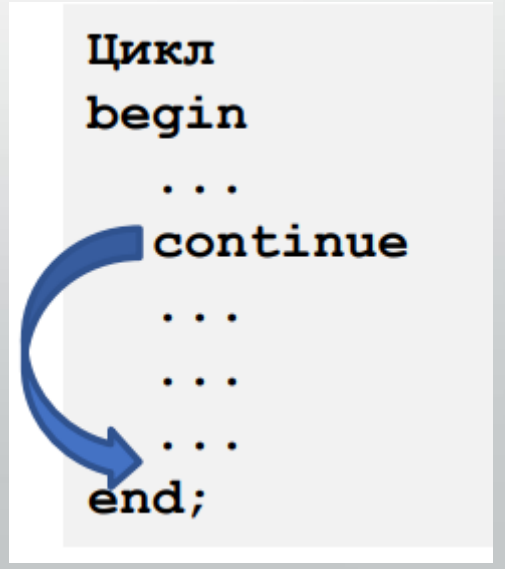
Число 3 найдено: True

```
begin
  var n := ReadInteger('Сколько чисел?');
  var exists := false;
  var k := ReadInteger('Какое число будем искать?');
  print('Введите последовательность');
  loop n do
  begin
    var x := ReadInteger;
    if x = k then
      exists := true;
  end;
  print ($'Число {k} найдено: ', exists)
end.
```

Break и continue

- Если ключевое слово break встречается в цикле, то происходит немедленный выход из цикла, и программа продолжает выполнять операторы, следующие после цикла.
- Если ключевое слово continue встречается в цикле, то происходит выход из текущей итерации цикла, и программа продолжает выполнять операторы следующей итерации цикла.
- break и continue могут использоваться только внутри цикла.





Пример использования continue

- Задача: распечатывать степени 2 до 10-й степени, кроме 2^6.
- Пример: 2 4 8 16 32 128 256 512 1024

```
1  begin
2  var a:=1;
3  while a<1000 do
4  begin
5          a:=a*2;
6          if a=64 then
7            continue;
8          print(a);
9  end;
10 end.
```

Пример использования continue

- В примере справа можно видеть, что использование оператора continue дает более читаемый код:

```
loop …
  begin
    var x := ReadInteger;
    if p(x) then
    begin
      30 statements
      ...
    end; // Oh! We're waiting!
  end;
```

```
loop …
  begin
    var x := ReadInteger;
    if not(p(x)) then
      continue;
    30 statements
    ...
  end;
```

Поиск числа в последовательности с использованием break

- Задача: Вводятся значения n и k. Вводятся n чисел последовательности. Программа должна определить есть ли число k среди введенной последовательности.
- Решение: Если искомое число найдено – выходим из цикла.

```
begin
  var n := ReadInteger('Сколько чисел?');
  var exists := false;
  var k := ReadInteger('Какое число будем искать?');
  print('Введите последовательность');
  loop n do
  begin
    var x := ReadInteger;
    if x = k then
      exists := true;
  end;
  print ($'Число {k} найдено: ', exists)
end.
```

```
var Exists := False;
loop n do
begin
  var x := ReadInteger;
  if x = k then
  begin
    Exists := True;
    break;
  end;
end;
```

Бесконечный цикл

```
while True do
begin
   ...
end;
```

```
repeat
   ...
until False;
```

```
while x>0 do
begin
   y += 1;
end;
```

```
repeat
   y += 1;
until x<=0;
```

Бесконечный цикл

- Задача: Вводится последовательность целых чисел. Признак завершения последовательности - число 0 (если введен 0, ввод последовательности прекращается). Программа должна вывести количество положительных чисел среди введенной последовательности.

Решение # 1

```
1  begin
2    var count:=0;
3    var x:=readinteger;
4    while x<>0 do
5    begin
6      if x>0 then
7        count+=1;
8      x:=readinteger;
9    end;
10   print($'positive = {count}')
11 end.
```

Решение # 2

```
1  begin
2    var count := 0;
3    while true do
4    begin
5      var x := readinteger;
6      if x = 0 then
7        break
8      else if x > 0 then
9        count += 1;
10   end;
11   print($'positive = {count}')
12 end.
```

Сумма цифр натурального числа

- Задача: Дано натуральное число m. Посчитайте сумму его цифр
- Решение: с помощью операций div / mod можно разделить число на цифры

```
var m := ReadInteger;
Assert(m>0);
var s := 0;
while m > 0 do
begin
  s += m mod 10;
  m := m div 10;
end;
```

Количество цифр числа, удовлетворяющих условию

- Задача: Задано натуральное m. Сколько цифр "1" в десятичном представлении числа?

```
var m := ReadInteger;
Assert(m>0);
var count := 0;
while m > 0 do
begin
  if m mod 10 = 1 then
    count += 1;
  m := m div 10;
end;
```

Возрастающая ли последовательность. Сдвиг значений

- **Задача:** Задана последовательность целых чисел. Последнее число последовательности – 0. Программа должна вывести 0 в случае, если числа последовательности образуют Невозрастающую последовательность, в обратном случае программа выводит 1.
- **Пример вывода**:

```
Введите последовательность:
>>>1 >>>5 >>>9 >>>5 >>>0
Результат: 0 (Невозрастающая по
++++++++++++++++++++++++++
Введите последовательность :
>>>1 >>>2 >>>5 >>>9 >>>11 >>>0
Результат: 1 (возрастающая посл
```

1 5 9 5 0 0

1 2 5 9 11 0 1

Решение 1

```
begin
println('please, enter the sequence, print 0 if you want to stop:');
  var a1:= readinteger; // 1-st element of the seq
  var a2:= readinteger; // 2-nd element of the seq
  var c := 1; // c = 1 if the sequence is increasing
  while a2 <> 0 do
  begin
    if a2 < a1 then // if non-increasing sequence
    begin
      writeln('element is less than the previous');
      c := 0 // non-increasing sequence
    end;
    a1 := a2; // shift of the elements
    read(a2); // input of the next element
  end;
  println('result = ', c)
end.
```

Сдвиг значений элементов последовательности

- Задача: Задана последовательность целых чисел. Последнее число последовательности – 0. Программа должна вывести 0 в случае, если числа последовательности образуют Невозрастающую последовательность, в обратном случае программа выводит 1.
- Пример вывода:

```
Введите последовательность:
>>>1 >>>5 >>>9 >>>5 >>>0
Результат: 0 (Невозрастак
++++++++++++++++++++++++++
Введите последовательност
>>>1 >>>2 >>>5 >>>9 >>>11
Результат: 1 (возрастающа
```

```
1  5  9  5  0


1  2  5  9  11  0
```

Решение 2

```
begin
  println('please, enter the sequence, print 0 if you want to stop:
  var a1 := integer.MaxValue; // 1-st element of the seq
  var a2: integer;
  var c := 1; // c = 1 if the sequence is increasing
  while a1 <> 0 do
  begin
    read(a2); // input of the next element
    if (a2 < a1) and (a1 <> integer.MaxValue) and (a2 <> 0) then
    begin
      c := 0 // non-increasing sequence
    end;
    a1 := a2; // shift of the elements
  end;
  println('result = ', c)
end.
```

Сдвиг элементов последовательности

Fibonacci sequence:

```
1   1   2   3   5   8   13   21   34   55
a   b   c=a+b
```

- Правило: Каждое последующее число последовательности – это сумма двух предыдущих чисел
- Решение 1:

On some step:

```
a     b     c=a+b
5     8     13     21
```

On next step:

```
5     8     13     21
      a     b
```

новое значение a

новое значение b

```
a := b;
b := c;
```

Переприсваивание a и b

```
var (a,b) := (1,1);
Print(a,b);
loop n-2 do
begin
  var c := a + b;
  Print(c);
  a := b;
  b := c;
end;
```

Сдвиг элементов последовательности

Fibonacci sequence:

1 1 2 3 5 8 13 21 34 55

some step:

b c=a+b

8 13 21 34 55

next step:

8 13 21 34 55

a b

value of a

new value of b

```
var (a,b) := (1,1);
Print(a,b);
loop n-2 do
begin
  (a, b) := (b, a + b);
  Print(b);
end;
```

```
b;
c;
```

новое значение a

новое значение b

reassignment of a & b

Переприсваивание a и b

PascalA

кортежное присваивание

НОД(a,b) – Наибольший общий делитель

Example.
144 = **2*2***2*2*3***3**
60 = **2*2*3***5

GCD(144,60) = 2*2*3 = 12

- Алгоритм Евклида (3 век до Н.Э.):

a	b	c=a mod b		
144	60	24	**12**	0

Solution.

```
var (a, b) := ReadInteger2;
Print(a, b);
var c:integer;
repeat
  c := a mod b;
  Print(c);
  a := b;
  b := c;
until b = 0;
Print(c);
```

```
144 60 24 12 0
```

НОД(a,b) – Наибольший общий делитель

Example.
144 = **2*2***2*2*3***3**
60 = **2*2*3***5

GCD(144,60) = 2*2*3 = 12

- Алгоритм Евклида (3 век до Н.Э.):

a	b	c=a mod b		
144	60	24	**12**	0

Solution 2.

```
var (a,b) := ReadInteger2;
Assert(b<>0);
repeat
  var c := a mod b;
  (a,b) := (b,a mod b);
until b=0;
Print(a);
```

```
12
```

Кортежное присваивание

Простые сомножители

Example.
144 = **2*2***2*2*3*3**

- Алгоритм:
  - i – кандидат в делители
  - Первый кандидат: i = 2
    - Если x делится на i, выводим i и делим x на i
    - Если нет – увеличиваем i на 1.
  - Процесс останавливается когда

  x становится = 1

```
var x := ReadInteger;
Assert(x>=2);
var i := 2;
repeat
  if x mod i = 0 then
  begin
    Print(i);
    x := x div i;
  end
  else i += 1;
until x = 1;
```

```
144
2 2 2 2 3 3
```

Является ли число простым?

- Задача: Дано натуральное n. Является ли число простым?
- Решение. n – простое число если оно делится только на 1 и на само себя. Если n делится на 2 .. n-1, тогда число является составным

```
var IsPrime := True;
for var i:=2 to n-1 do
  if n mod i = 0 then
  begin
    IsPrime := False;
    break;
  end;
```

Улучшенный алгоритм

- Если число n составное, тогда оно может быть разделено на число $\leq \sqrt{n}$

Let's divide n by 2 .. $\sqrt{n}$

Code:

```
var IsPrime := True;
for var i:=2 to Round(Sqrt(n)) do
  if n mod i = 0 then
  begin
    IsPrime := False;
    break;
  end;
```

For n = 1000003

this algorithm makes 1000 iterations

previous one – 1000000 iterations

Q & A