Basics of Programming. Introduction

Course Basics of Programming Semester 1, FIIT

Mayer Svetlana Fyodorovna

Nested loops

- To do: Calculate the value of the function $z(x,y) = x^y$ for every x changing in the interval [2;8], and y changing in the interval [2;5].
- The resulting example:

```
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);
       writeInformat('z(x,y) = {0}^{1} = {2}',x,y,z);
    end;
end.
```

We must create two for loops (nested loop): one loop within the other.
 Variable x has to be modified in the outer loop; variable y has to be modified in the inner loop.

Nested loops

- To do: The program must display the rows with the following sequences as in the example below. Use *nested loops*.
- 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.
```

• The outer *for* loop must iterate over the columns, the inner loop must iterate over the sequences within each row...

- To do:
- Lesson # 9, Tasks, nested loops: 1, 2, 3_0,3_1, 4, 5, 6

Boolean type (the problem of finding a value)

- To do: The values of n and k are entered. n numbers of the sequence are entered. The program should output if there is a number k among them
- Solution. Let's define a boolean variable exists. Initialize it to false value. In the loop, change it to true value if any number is equal to k.

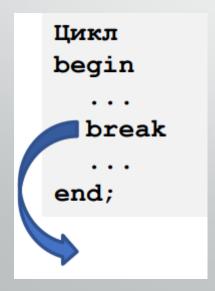
• The resulting example:

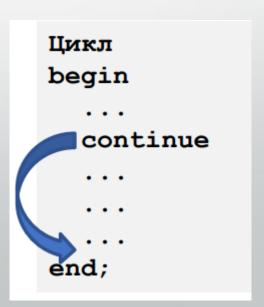
```
how many numbers? >>>5
what number must we find? >>>3
enter numbers, please
1 3 6 4 7
there is 3 number: True
```

```
begin
    var n := ReadInteger('how many numbers?');
    var exists := false;
    var k := ReadInteger('what number must we find?');
    print('enter numbers, please');
     loop n do
    begin
      var x := ReadInteger;
      if x = k then
         exists := true;
     end;
    print ($'there is {k} number: ', exists)
13 end.
```

Break and continue

- When a break statement is encountered inside a loop, the loop is immediately terminated, and the program control resumes at the next statement following the loop.
- When a continue statement is encountered inside a loop, the current iteration of a loop is immediately terminated, and the program control resumes at the next loop iteration.
- break и continue statements can only be used inside loops





The example of using continue statement

- To do: print powers of 2 up to the 10th power, except 2⁶.
- E.g.: 2 4 8 16 32 128 256 512 1024

```
begin
var a:=1;
while a<1000 do
begin
       a:=a*2;
       if a=64 then
         continue;
       print(a);
 end;
 end.
```

The example of using continue statement

• in the example on the right you can see, that the using of continue gives a more readable code:

```
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;
```

The problem of finding a value using break

- To do: The values of n and k are entered. n numbers of the sequence are entered. The program should output if there is a number k among them?
- Solution: If a value is found, we break off the loop

```
pegin
var n := ReadInteger('how many numbers?');
var exists := false;
var k := ReadInteger('what number must we find?');
print('enter numbers, please');
loop n do
begin
var x := ReadInteger;
if x = k then
exists := true;
end;
print ($'there is {k} number: ', exists)
end.
```

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

- To do:
- Lesson # 9, Tasks: 7

To do: 10 integers are generated in the range [-3;10]. The program should output if there is at least one positive number among them?

The resulting example:

5 -2 7 9 1 2 4 6 0 9

There is positive number: true

The Infinite Loop

```
while True do
                             repeat
begin
                             until False;
end;
                             repeat
while x>0 do
                               y += 1;
begin
                             until x <= 0;
  y += 1;
end;
```

The Infinite Loop

• To do: A sequence of integers is given. The last number of the sequence is 0 (if 0 is entered the input of the numbers of the set is finished). The program must output the number of positive elements among the sequence.

Solution # 1

```
begin
  var count:=0;
  var x:=readinteger;
  while x <> 0 do
  begin
    if x>0 then
      count+=1;
    x:=readinteger;
  end;
  print($'positive = {count}')
end.
```

Solution # 2

```
begin
  var count := 0;
  while true do
  begin
    var x := readinteger;
    if x = 0 then
      break
    else if x > 0 then
      count += 1;
  end;
  print($'positive = {count}')
end.
```

- To do:
- Lesson # 9, Tasks infinite loops: 8, 9

Sum of digits of natural number

- To do: Natural number m is given. Calculate sum of its digits
- Solution: div / mod operations split the number on the series of its digits

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

Number of natural number digits satisfying some condition

• To do: Natural number m is given. How many "1" digits in its decimal representation does it have?

```
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;
```

- To do:
- Lesson # 9, Task 10, 11

Shift of a sequence elements

• To do: A sequence of integers is given. The last number of all the sequences is 0 (if 0 is entered the input of the numbers is finished). The program has to print 0 in the case when the sequence forms a non-increasing sequence of numbers, otherwise the program has to print the number 1.

• The resulting example:

1 5 9 5 0

1 2 5 9 11 0

Solution 1

```
begin
println('please, enter the sequence, print 0 if you want to stop:');
  var al:= 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 \ll 0 do
 begin
    if a2 < a1 then // if non-increasing sequence</pre>
   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.
```

Shift of a sequence elements

To do: A sequence of integers is given. The last number of all the sequences is 0 (if 0 is entered the input of the numbers is finished). The program has to print 0 in the case when the sequence forms a non-increasing sequence of numbers, otherwise the program has to print the number 1.

The resulting example:

1 5 9 5 0

1 2 5 9 11 0

Solution 2

```
begin
 println('please, enter the sequence, print 0 if you want to stop:
  var al := integer.MaxValue; // 1-st element of the seq
 var a2: integer;
  var c := 1; // c = 1 if the sequence is increasing
  while al \ll 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;
    al := a2; // shift of the elements
  end;
 println('result = ', c)
end.
```

- To do:
- Lesson # 9, Task 12, 13 (complex), 14 (complex), 15

Shift of a sequence elements

Fibonacci sequence:

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

- Rule: Each next number is equal to the sum of the two previous numbers
- Solution 1

On some step:

```
b c=a+b
a
      13 21 34 55
```

On next step:

```
13
    21
       34
```

new value of a

```
begin
                         var c := a + b;
                         Print(c);
                         a := b;
             55
                         b := c;
                       end;
new value of b
```

Print(a,b);

loop n-2 do

var (a,b) := (1,1);

a := b;b := c;

reassignment of a & b

PascalABC.NET show

Shift of a sequence elements

Fibonacci sequence:

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

- Rule: Each next number is equal to the sum of the two previous numbers
- Solution 2:

On some step:

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

On next step:

```
5 8 13 21 34 55
a b
```

new value of a

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

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;
```

reassignment of a & b

tuple assignment

- To do:
- Lesson # 9, Task 14

GCD(a,b) – Greater Common Divisor

```
Example.

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

60 = 2*2*3*5
```

• The Euclidean Algorithm (3 century BC):

```
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
```

GCD(a,b) - Greater Common Divisor

```
Example.

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

60 = 2*2*3*5
```

• The Euclidean Algorithm (3 century BC):

```
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);
```

tuple assignment

Prime factorization

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

- Algorithm:
 - i is candidate to divisors
 - First candidate: i = 2
 - If x is divided by i, we print i and divide x by i
 - If not, we increase i by 1.
 - The process stops when x becomes = 1

```
var x := ReadInteger;
Assert (x>=2);
var i := 2;
repeat
  if x \mod i = 0 then
  begin
    Print(i);
    x := x \operatorname{div} i;
  end
  else i += 1;
until x = 1;
144
2 2 2 2 3 3
```

Is the number prime?

- To do: natural n is given. Is it prime?
- Solution. n is a prime number if it can only be divided by 1 and itself. If n is divisible by 2 .. n-1, then this is a composite number

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

Q&A