L#5

Basics of Programming. Introduction

Course Basics of Programming Semester 1, FIIT

Mayer Svetlana Fyodorovna

Sums

- The sum in Pascal is calculated by the recurrent expression:

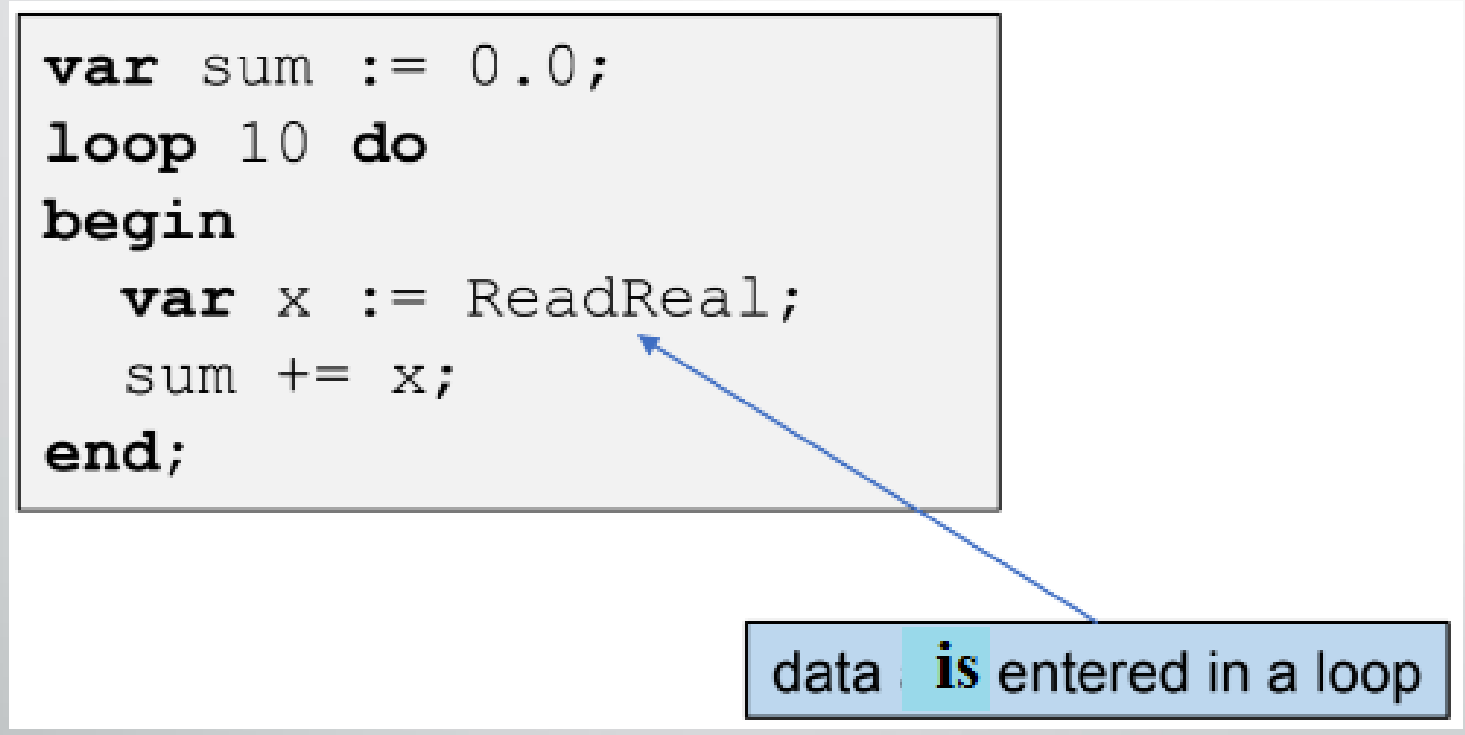
$$S = S + Y$$

where S is the accumulated amount
Y – next summand

Sums

- To do: 10 numbers are given. It's required to calculate their sum.

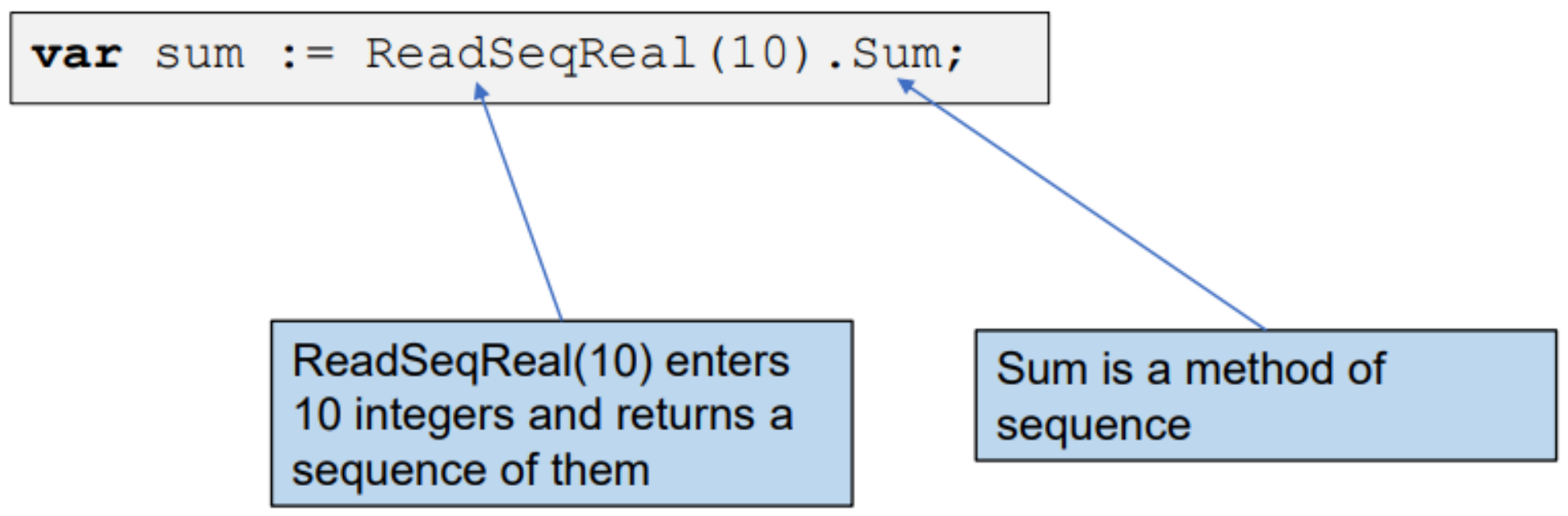
```
var sum := 0.0;
loop 10 do
begin
  var x := ReadReal;
  sum += x;
end;
```



The sum variable is initialized by 0 before loop. Every loop iteration the variable sum is incremented by a value of the next entered number.

Sums – short solution

- To do: 10 integers are given. It's required to calculate their *sum*.



Sums

- To do: calculate a sum of all odd 2-digit integers.

11 + 13 + 15 + 17 + 19 + … + 95 + 97 + 99

Code with loop

```
var sum := 0;
var x := 11;
loop 45 do
begin
  sum += x;
  x += 2;
end;
```

Code with for

```
var sum := 0;
for var i:=0 to 44 do
  sum += 11 + i * 2;
```

data are calculated during iterations

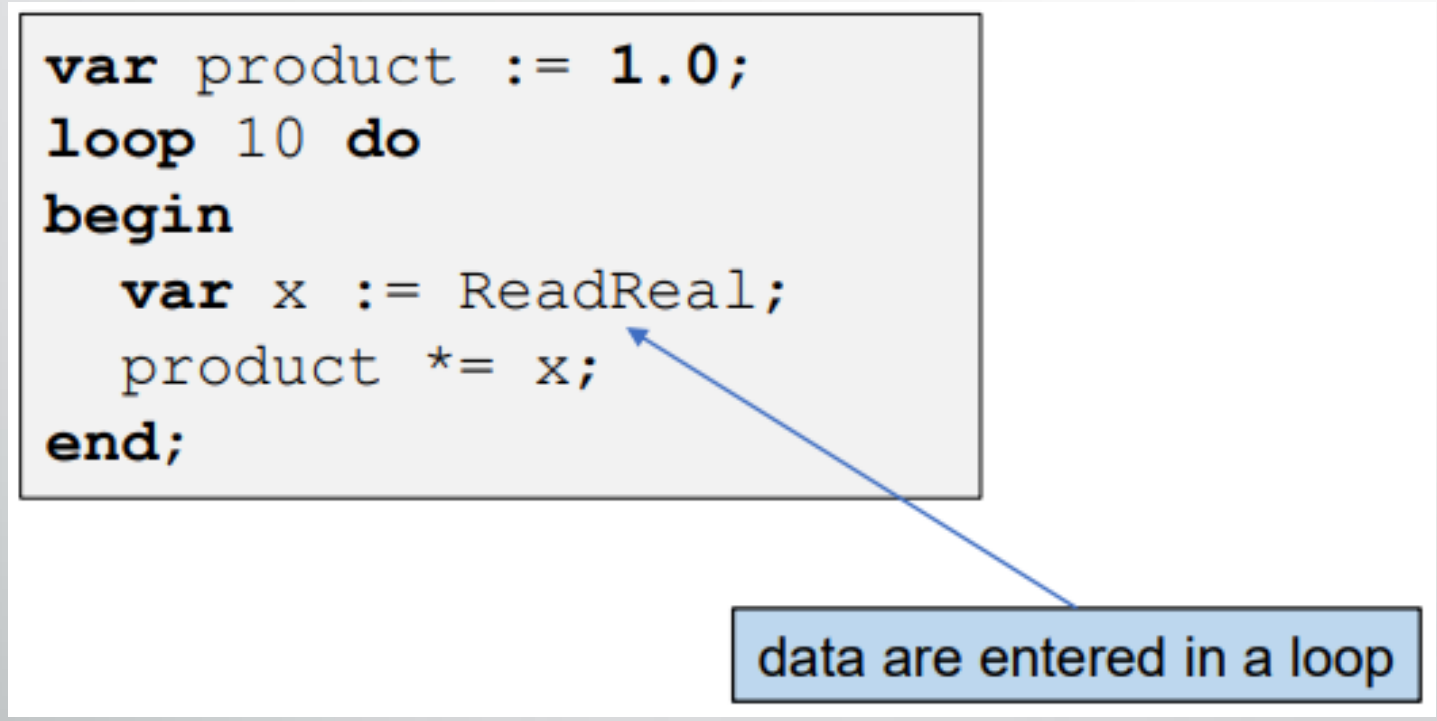
Tasks

- To do:
- Lesson # 8, Tasks, sum: 1, 2, 3, 4, 5, 6, 7

Products (MULTIPLICATION)

- To do: 10 reals are given. It's required to calculate their *product* (multiplication).

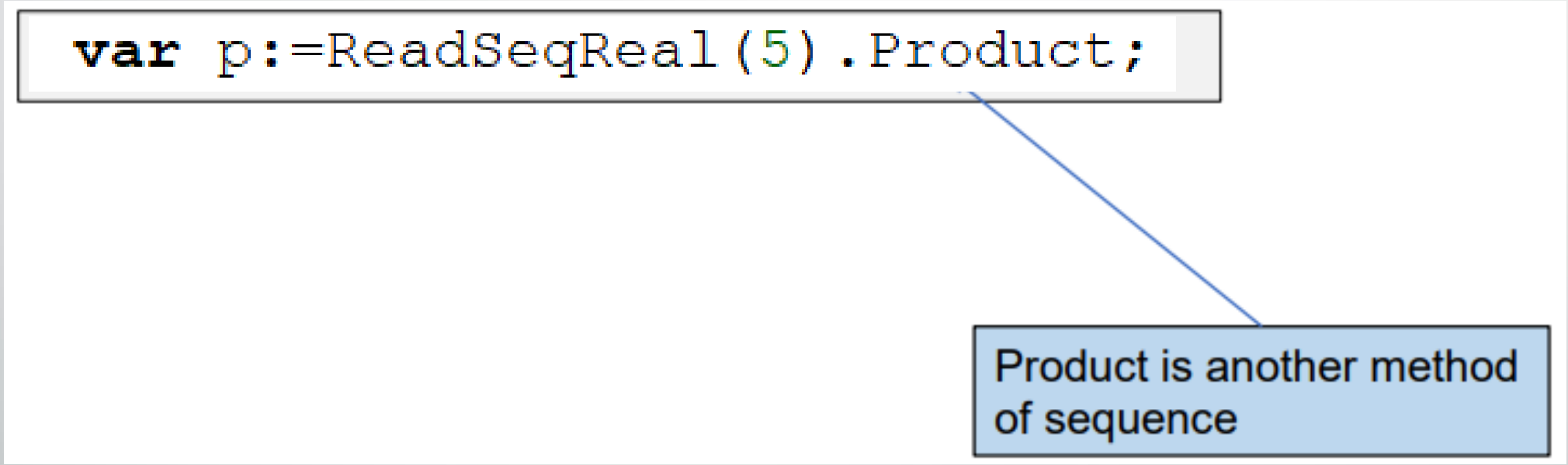
```
var product := 1.0;
loop 10 do
begin
  var x := ReadReal;
  product *= x;
end;
```



- The product variable is initialized by 1 value before loop. Every loop iteration the product is incremented by the value of the next number.

Product (multiplication) — short solution

- To do: 5 real numbers are given. It's required to calculate their *product*.



Products

- To do: Integer n is given (n >= 0).

factorial

Calculate $n! = n \cdot (n-1) \cdot (n-2) \cdot \ \ldots \ \cdot 2 \cdot 1$

Solution: We' re going to use a for loop with counter i changing from 1 to n

```
var n := ReadInteger;
var product := 1;
for var i:=1 to n do
  product *= i;
```

Tasks

- To do:
- Lesson # 8, Tasks, product: 1, 2, 3, 4, 5

Counters

- To do: n >= 0 is entered. The program should ask to input n integer numbers and find a quantity of odd numbers among entered numbers.
- Solution: We're going to use an if statement in a loop.

```
var n := ReadInteger;
var count := 0;
loop n do
begin
  var x := ReadInteger;
  if x mod 2 <> 0 then
    count += 1;
end;
```

Several counters

- To do: n grades (marks from 2 to 5) on an exam are given. Calculate the number of each grade (how many “2”, how many “3” …).
- Solution: We will use several counters, each counter for each grade.

```
var n := ReadInteger;
var (c2,c3,c4,c5) := (0,0,0,0);
loop n do
begin
  var Mark := ReadInteger;
  case Mark of
  2: c2 += 1;
  3: c3 += 1;
  4: c4 += 1;
  5: c5 += 1;
  end;
end;
Print(c2,c3,c4,c5);
```

Tasks

- To do:
- Lesson # 8, Tasks, counters: 1, 2, 3, 4, 5

Minimum and maximum value of n numbers

- Solution 1. Let's assign the first entered number to min variable. Then, in a loop check if the next entered number is less than min. If it is, so reassign this value to min:

```
begin
  var n:=readinteger('enter n');
  var x := ReadReal;
  var min := x;
  loop n - 1 do
  begin
    x := ReadReal;
    if x < min then
      min := x;
  end;
  print($'min = {min}')
end.
```

What is not good: The first value is handled separately.

Minimum and maximum value of n numbers

- Solution 2. Let's set the maximum value of real type (real.MaxValue) to min variable. Then, in a loop check if the first entered number is less than min. If it is, so reassign this value to min. The idea is that after the first iteration min will be reassigned in any case because x < real.MaxValue:

```
begin
  var n:=readinteger('enter n');
  var x:real;
  var min := real.MaxValue;
  loop n do
  begin
    x := ReadReal;
    if x < min then
      min := x;
  end;
  print($'min = {min}')
end.
```

Tasks

- To do:
- Lesson # 8, Tasks, min & max: 1, 2, 3, 4, 5, 6

Sums and products in a while and repeat loops

- Calculate the sum of the sequence: 11 + 13 + 15 + … + 99.

```
begin
  var x:=11;
  var s:=0;
  while x<=99 do
  begin
    s+=x;
    write(x);
    x+=2;
  end;
  Print($'sum = {s}');
end.
```

```
var x := 11;
var s := 0;
loop 45 do
begin
  s += x;
  x += 2
end;
```

Tasks

- To do:
- Lesson # 8, Tasks, sum and while: 1, 2, 3, 4

Q & A