



L#5

Основы алгоритмизации и программирования.

Введение

Педагогическое образование, 3 семестр

Mayer Svetlana Fyodorovna

Сумматоры

- Сумматор в паскале вычисляется по рекуррентному выражению:

$$S = S + Y$$

Где S аккумулируемая сумма
 Y – следующее слагаемое

Сумматоры

- **Задача:** Дано 10 чисел. Необходимо посчитать их сумму.

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

Данные вводятся в цикле

Переменная **sum** инициализируется **0** перед циклом. На каждой итерации цикла **sum** увеличивается на значение следующего введенного числа.

Сумматоры – короткая запись

- **Задача:** Дано 10 чисел. Необходимо посчитать их сумму.

```
var sum := ReadSeqReal(10).Sum;
```

ReadSeqReal(10) – запрашивает 10 целых чисел и возвращает их последовательность

Sum – это метод последовательностей

Сумматоры

- **Задача:** Посчитать сумму всех **2-разрядных нечетных**

$$11 + 13 + 15 + 17 + 19 + \dots + 95 + 97 + 99$$

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;
```

Вычисления
выполняются
внутри цикла

Произведение

- **Задача:** Запрашиваются 10 вещественных чисел. Требуется посчитать их произведение

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

Данные вводятся
внутри цикла

- Переменная **product** инициализируется значением **1** перед циклом. На каждой итерации цикла **product** увеличивается на значение следующего числа.

Произведение — короткая запись

- **Задача** : дано 5 вещественных чисел. Необходимо посчитать их произведение.

```
var p := ReadSeqReal (5) .Product;
```

Product – метод
последовательностей

Произведение

- **Задача** : Дано целое число n ($n \geq 0$). факториал

Посчитать: $n! = n \cdot (n - 1) \cdot (n - 2) \cdot \dots \cdot 2 \cdot 1$

Выполнение: Мы собираемся использовать цикл `for` с изменением счетчика i с **1** до **n** .

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


Счетчики

- **Задача:** вводится $n \geq 0$. Программа должна попросить ввести n целых чисел и найти количество нечетных чисел среди введенных чисел.
- **Выполнение:** Мы собираемся использовать оператор **if** в цикле .

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

Несколько счетчиков

- **Задача:** n – количество оценок за экзамен, mark - оценка (от 2 до 5). Выведите количество каждой оценки
- **Выполнение:** Мы будем использовать несколько счетчиков, - для каждой оценки.

```
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);
```

Minimum и maximum значения n чисел

- **Решение 1.** Присваиваем первое введенное число переменной `min`. Затем в цикле проверяем, меньше ли следующее введенное число чем `min`. Если это так, переприсваиваем это значение для `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.
```

Что не хорошо: Первое число
вводится отдельно

Minimum и maximum значения n чисел

- **Решение 2.** Установим максимальное значение типа *real* (`real.MaxValue`) для переменной `min`. Затем в цикле проверим, меньше ли первое введенное число чем `min`. Если это так, переприсвоим это значение для `min`. Идея состоит в том, что после первой итерации `min` в любом случае будет переназначен, потому что $x < \text{real.MaxValue}$:

```
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.
```

Сумматоры и произведение в циклах **while** и **repeat**

- Посчитать сумму последовательности: $11 + 13 + 15 + \dots + 99$.

```
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;
```



Q & A