

Fun name: enumerable

$Enumerable < T >$

var s: Enumerable < integer >

s := new List < integer >;

s := new HashSet < integer >;

...

Method - expression is not possible.

$x \rightarrow x * x$

i var. var f: function (x: integer): integer := $x \rightarrow x * x$;

ii var. type IntFunc = function (x: integer): integer;
var f: IntFunc := $x \rightarrow x * x$;

iii var. type Func < T1, T2 > = function (x: T1): T2;
var f: Func < integer, integer > := $x \rightarrow x * x$;

iv var. var f: Func < integer, integer > := $x \rightarrow x * x$;
Function - expression is not possible

Func < T1, Res >

Func < T1, T2, Res >

Func < Res >

Action < T >

Action < > = procedure

Action < T1, T2 >

$\text{Predicate} \leftarrow T$ - унарн. ифункционал

$\text{Predicate} \leftarrow T_1, T_2$ - бинарн. ифункционал

$\text{Predicate} \leftarrow 0$

$\text{Int Func} = \text{Func} \leftarrow \text{integer}, \text{integer}$;

Real Func

String Func

бap. $\text{var } f: \text{IntFunc} := x \rightarrow x * x;$

Или $\text{unary} \text{ или } \text{op-um} : T \rightarrow T$
 $T \rightarrow \text{boolean} \sim \text{Predicate} \leftarrow T$

$\text{var } f_2: \text{Func} \leftarrow \text{integer}, \text{integer}, \text{real};$

$f_2 := (x, y) \rightarrow x/y;$

$(x, y: \text{integer}) \rightarrow x/y;$

$(x, y: \text{integer}): \text{real} \rightarrow x/y;$

$(x: \text{integer}; y: \text{integer}) \rightarrow x/y;$

$(x; y) \rightarrow x/y;$

$(x; y) \rightarrow \text{begin var } a := x * y;$
 $\text{result} := a/y; \text{end};$

Задания - упражнения

var p : $\text{Action} \leftarrow \text{integer} := \text{procedure}(x) \rightarrow$ begin
Print(x)
end;

p(1);

Обучение
Pascal

Процедура перебора значений

1, 2 ... 10

Range (1, 10) - перебрать все значения от 1 до 10
Вывод: $\text{integer} := \text{integer}$

Range (1, 10). Print;

Range (1, 10, 2). Println; // 1 3 5 7 9
от 1 до 10 шаг 2

Range (1.0, 2.0, 10). Println; // 1 1.1 1.2 ... 2.0
//

Range ('A', 'Z');

Seq (1, 3, 7) - случайный массив

SeqRandom (20, 1, 5). Println
количество элементов в массиве от 1 до 5

сгенерировать массив из 20 чисел
в диапазоне от 1 до 5
Вывод: массив

SeqRandom Real. Print;

10 эл. в диапазоне от 1.0 до 10.0

`SeqLen (1,  $x \rightarrow x * 2$ , 10)`

начинает
с единицы

кол-во эл. послед

`SeqLen (1, 1, (a, b)  $\rightarrow a + b$ , 20)`. Print

печатают послед. Fibonacci

выпав., пока эл

`SeqWhile (10,  $x \rightarrow x - 1$ ,  $x \rightarrow x > 0$ )`

начинает с 10

интерпретирует, пока выпав. упр-ие

`SeqFill (777, 5)`

возвращает массив из 5 эл., где все эл.
массива ($a[i] = 777$) равны 777

`SeqFill (10,  $i \rightarrow i * i$ )`

заполнение каждого элем. массива функцией.

Методы последовательностей
(массива - запросов).

Примеры кода.

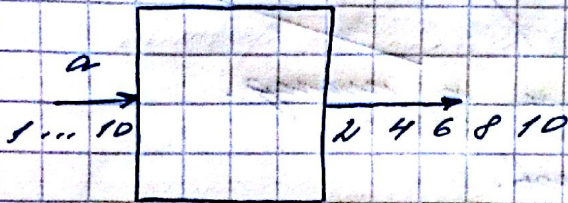
`var a := Range (1, 10);`

`a.Where (x  $\rightarrow$   $x \bmod 2 = 0$ ). Print;`

↑ упр-е. предикат

// 2 4 6 8 10

Where

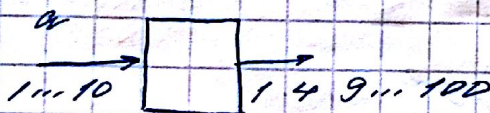


var $q := a$. Where $(x \rightarrow x \bmod 2 = 0)$;
 q . Print;

Abstraction memq.

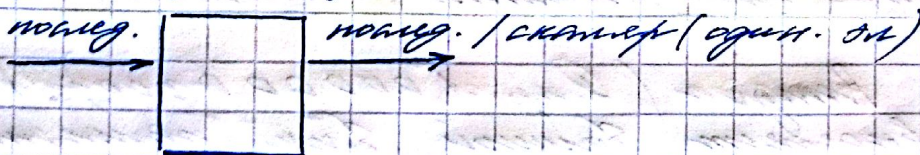
var $a := \text{Range}(1, 10)$;

Select



var $q := a$. Select $(x \rightarrow x * x)$;
 q . Print; // 1 4 9...100

memq memq.



var $a := \text{Range}(1, 10)$;
var $q := a$. Where $(x \rightarrow x \bmod 2 = 0)$. Select $(x \rightarrow x * x)$;
 q . Print;

a . Where $(x \rightarrow x \bmod 2 = 0)$. Print. Select $(x \rightarrow x * x)$. Print;
 ← memq. memq. memq. memq.

a. Where ($x \rightarrow x \bmod 220$). Select ($x \rightarrow x * x$). Print

запросы к подмассиву.

Уточнение запросов.

Range и замена цикла for.

var $s := 0$; var $x := 1$;

for var $i := 1$ to 10 do

begin

$s := s + x * x$;

$x := x + 2$;

end;

// $1^2 + 3^2 + \dots + 19^2$

1 3 ... 19

Write (Range (1, 20, 2). Select ($x \rightarrow x * x$). Sum);

Range и выделение памяти

Range (1, 10000000000)
не выделяет большой блок памяти.

Range (1, 10000000000). Where ($x \rightarrow x < 10$). Print;

Readlines ('a.txt'). Print (NewLine);
 ↳ багганы. $\uparrow$ const #13#10
 Iterable < string >

Дадмал мэдээлэл гаргах.

fregs. txt

Загвар:
 багганы багганы
 (verb).

2.7 word1 verb
 3.4 word2 noun
 ...

LINQ - Language Integrated Query.

query: = Readlines ('fregs.txt'). Select (s → ToWords()). Where (a → AEIJ = 'verb'). Select (a → AEIJ);
 query. Print ();

query - мэдээлэл гаргах

var s: = 'abc 123 de 98 f4g';

Статусын үзүүлэлт

s.Where (c → char. IsDigit(c)). Select (StrToInt). Sum();

x - нэгжээр →

$x \bmod i < 0$ гүйц $i = 2 \dots \sqrt{x}$

• Sum (c → StrToInt (c));

function IsPrime (x: integer): boolean;

begin

Result := Range (2, Round (Sqrt(x))). All (i → x mod i < 0);

end;

begin

Range (2, 1000). Where (IsPrime). Print;

end.

Bei Annahme konstante.

var vowels := 'a e i o u y';

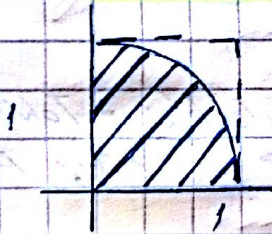
var g := Range('a', 'z'). Except(vowels);

'b' 'c' 'd' ... 'z' → 'bcd...z'

writeln(g. Aggregate("", (s, c) → s+c));

var s := "";
foreach var c in g do
s := s+c;

Berechne mit Pi m-on Monte-Karlo



Seg. exp. = π

$$\frac{m}{n} \rightarrow \frac{\pi}{4}$$

var n := 10000000;
var MyPi := Range(1, n). Select(x → Rec(Random(), Random()));
• Where (p → $\frac{p \cdot \text{Item1}}{\text{Item2}} < 1$)
• Count() / n * 4;

QuickSort

function QuickSort(a: Enumerable<real>):
Enumerable<real>;

begin
if a.Count = 0 then
Result := a

else begin

var f := a.First;

var left := a.Skip(1). Where(x → x ≤ f);
var right := a.Skip(1). Where(x → x > f);

QuickSort(left) & QuickSort(right)

Result := QuickSort(left) + Seq(f) + QuickSort(right);
end;
end;

Параллельность
прямые задачи

Range(1, n). As Parallel. Where (Is Prime): Count();

15.05.2015г.

Написание

ООП (объектн. - ориент. прог.) = инкапсуляция +
+ наследование + полиморфизм.

инкапсуляция = хранение в классе
данных и методов. Защита
данных (private)

Наследование - вид отнош. между
классами, при котором производн. класс
Base - базовый класс / надкласс / супер-класс
↑
Derived - производн. класс / подкласс /
потомок

наслед. все, что есть в базовом
классе (напр., м-ды, св-ва) и
может добав. новые и измен.
стар. м-ды.