



Запрещенные методы

Демяненко Я.М.

Научно-методическая конференция
"Использование системы программирования PascalABC.NET
в обучении программированию"
29-30 марта 2023 г.

Что такое методы?

```
var a := ReadArrInteger(N);
```

```
a.Length  
a.Sum  
a.Average  
a.Min  
a.Max
```

Кто их запретил и почему?

Школьники

- Легко и просто запомнить
- Понятно, как использовать
- Коротко записывается

Преподаватели

- А как же алгоритмы?
- Непривычно
- В учебниках нет

Различные языки — различные подходы к реализации методов

Python



Класс языка объектно-ориентированный язык программирования

Появился в 20 февраля 1991^[15]

C#



Класс языка мультипарадигмальный: объектно-ориентированный, обобщённый, процедурный, функциональный, событийный, рефлексивный

Появился в 2001

PascalABC.NET

Семантика императивная

Класс языка мультипарадигменный: структурное, объектно-ориентированное, обобщённое, процедурное программирование, функциональное программирование

Тип исполнения компилятор

Появился в 2007

И ещё много других языков ...

C++



Семантика мультипарадигмальный: объектно-ориентированное, обобщённое, процедурное, метапрограммирование

Класс языка объектно-ориентированный язык программирования, мультипарадигмальный язык программирования, процедурный язык программирования, язык функционального программирования, язык обобщённого программирования^[d], язык программирования, **free-form language**^[d] и компилируемый язык программирования

Тип исполнения компилируемый

Появился в 1983

Познакомимся поближе

Задача 17

Рассматривается множество целых чисел, принадлежащих числовому отрезку $[1016; 7937]$, которые делятся на 3 и не делятся на 7, 17, 19, 27.

Найдите количество таких чисел и максимальное из них. В ответе запишите два целых числа: сначала количество, затем максимальное число

<https://pascalabc.net>

Решение 1. Минимум новых возможностей; длинная запись условия

```
begin
  var count := 0;
  var last := 0;
  for var x := 1016 to 7937 do
    if (x mod 3 = 0) and (x mod 7 <> 0) and (x mod 17 <> 0) and
      (x mod 19 <> 0) and (x mod 27 <> 0) then
      begin
        count += 1;
        last := x;
      end;
  Print(count, last);
end.
```

1568 7935

Решение 2. Использование методов Divs и DivsAny

```
begin  
  var count := 0;  
  var last := 0;  
  for var x := 1016 to 7937 do  
    if x.Divs(3) and not x.DivsAny(7, 17, 19, 27) then  
      begin  
        count += 1;  
        last := x;  
      end;  
  Print(count, last);  
end.
```

1568 7935

Решение 3. Использование методов последовательностей

```
begin  
  // Последовательность целых от 1016 до 7937,  
  // делящихся на 3 и не делящихся ни на одно из 7, 17, 19, 27  
  var seq := (1016..7937).Where(x -> x.Divs(3) and not x.DivsAny(7, 17, 19, 27));  
  // Выведем количество элементов этой последовательности и ее максимальный элемент  
  Print(seq.Count, seq.Last);  
end.
```

1568 7935

Проще или сложнее?

Есть ли алгоритм?

Непривычная реализация?

Последовательности

Последовательности — важнейший тип данных PascalABC.NET.

Они используются даже в простейших программах для начинающих.

Последовательности являются обобщением массивов, списков, множеств.

Последовательности описываются типом `sequence of T`

Универсальность методов для различных видов последовательностей

Для последовательностей существует огромное количество методов.

Все они доступны также и для разновидностей последовательностей:

- массивов,
- списков,
- множеств,
- строк

Метод — Алгоритм

Каждый метод последовательности представляет собой простейший алгоритм.

В качестве параметров таких алгоритмов повсеместно используются лямбды – они позволяют передавать в методы действия (процедуры) и преобразования (функции).

Методы как запросы

Запрос к переменной

вычислить и вернуть единичное значение или целую последовательность

```
var a:=Arr(1,2,3,4,5,6);  
var b:=a.Select(x->x*x);  
b.Println;
```

1 4 9 16 25 36

```
var a:=Arr(1,2,3,4,5,6);  
var b:=a.Sum;  
b.Println;
```

21

Составные запросы

```
var a:=Arr(1,2,3,4,5,6);  
var b:=a.Select(x->x*x);  
b.Println;
```

1 4 9 16 25 36

```
var a:=Arr(1,2,3,4,5,6);  
var b:=a.Select(x->x*x).Println;
```

1 4 9 16 25 36

```
var a:=Arr(1,2,3,4,5,6);  
var b:=a.Select(x->x*x).Sum.Println;
```

91

Важнейшие методы последовательностей

- `Select(x -> f(x))` — **проекция (трансформация)** элементов `x` на элементы `f(x)`
- `Where(x -> filter(x))` — **фильтрация (отбор)** элементов, удовлетворяющих условию `filter`
- `OrderBy(x -> x)` — **сортировка** по возрастанию
- `OrderBy(x -> x.поле)` — сортировка по заданному полю по возрастанию
- `OrderByDescending(x -> x)` — сортировка по убыванию

Преобразование («проекция»)

```
##
```

```
var A := | 1, 2, 3, 4, 5, 6 |;  
A.Select(x->x*x).Println
```

```
1 4 9 16 25 36
```

```
##
```

```
var A := Arr('a'..'z');  
A.Select(x->x.Code).Println; //char в integer
```

```
97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122
```

Преобразование в кортеж

```
##
```

```
var A := Arr('a'..'z');
```

```
A.Select( x->x.Code).Println; //char в integer
```

```
A.Select( x->(x,x.Code)).Println; //char в кортеж или пару
```

```
97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122
```

```
(a,97) (b,98) (c,99) (d,100) (e,101) (f,102) (g,103) (h,104) (i,105) (j,106) (k,107) (l,108) (m,109)  
(n,110) (o,111) (p,112) (q,113) (r,114) (s,115) (t,116) (u,117) (v,118) (w,119) (x,120) (y,121) (z,122)
```

Фильтрация элементов по условию (в новый массив)

```
##  
var a := arrGen(10, i->2*i+1);  
var b := a.Where(x -> x mod 3 = 0);  
b.println;  
println(b.Count);
```

```
3 9 15  
3
```

```
begin  
  var a := arrGen(10, i -> 2 * i + 1);  
  var b := new integer[10];  
  var k := 0;  
  for var i := 0 to 9 do  
    if a[i] mod 3 = 0 then  
      begin  
        b[k] := a[i];  
        k += 1;  
      end;  
  for var i := 0 to k-1 do  
    print(b[i]);  
  println;  
  println(k);  
end.
```

Цепочки методов

Эти методы преобразуют последовательность в другую последовательность и могут вызываться по цепочке:

```
s.Where(x -> x>3).OrderBy(x -> x).Select(x -> x*x).Println  
и т.д.
```

Запросы: All и Any

```
##  
var A := | 11, 21, 32, 43, 54, 61 |;  
A.All( x-> x>20 ).Println;  
A.Any( x-> x>20 ).Println
```

False

True

Задачи на логические выражения

Для какого наибольшего натурального числа A формула
 $\neg \text{ДЕЛ}(x, A) \rightarrow (\text{ДЕЛ}(x, 6) \rightarrow \neg \text{ДЕЛ}(x, 9))$
тождественно истинна (то есть принимает значение 1 при любом натуральном значении переменной x)?

##

```
for var a := 10000 downto 1 do
  if (1..100000).All(x -> not x.Divs(a) <= (x.Divs(6) <= not x.Divs(9))) then
    begin
      Print(a);
      break;
    end;
```

Б.С. Михлин. Напишите программу, которая ищет среди целых чисел, принадлежащих числовому отрезку [194441; 196500] простые числа, оканчивающиеся на 93.

```
## uses school;
```

```
(194441..196500).Where( x-> (x mod 100 = 93) and x.IsPrime ).Println;
```

```
195193 195493 195593 195893 196193
```

```
## uses school;
```

```
(194493..196500).Step(100).Where(x->x.IsPrime).Println;
```

```
195193 195493 195593 195893 196193
```

Получение списка делителей

```
##  
uses school;  
var A := Arr(21..30);  
A.Select( x->(x,x.Divisors) ).PrintLines
```

```
(21, [1, 3, 7, 21])  
(22, [1, 2, 11, 22])  
(23, [1, 23])  
(24, [1, 2, 3, 4, 6, 8, 12, 24])  
(25, [1, 5, 25])  
(26, [1, 2, 13, 26])  
(27, [1, 3, 9, 27])  
(28, [1, 2, 4, 7, 14, 28])  
(29, [1, 29])  
(30, [1, 2, 3, 5, 6, 10, 15, 30])
```

Ну очень длинное решение

(Е. Джобс) Рассматривается множество целых чисел, принадлежащих числовому отрезку $[25552; 58885]$, которые имеют не менее 15 двузначных делителей. Найдите наименьшее и наибольшее из таких чисел, а также их количество.

```
## uses school;

var A := (25552..58885).Select(x -> x.Divisors)
    .Where(divs -> divs.Count(d -> d in 10..99) >= 15)
    .Select(divs -> divs.Last);

Println(A.Min, A.Max, A.Count);

25650 58800 420
```

Ещё одно очень длинное решение

Демо-2021. Напишите программу, которая ищет среди целых чисел, принадлежащих числовому отрезку [174457; 174505], числа, имеющие ровно два различных натуральных делителя, не считая единицы и самого числа. Для каждого найденного числа запишите эти два делителя в таблицу на экране с новой строки в порядке возрастания произведения этих двух делителей. Делители в строке таблицы также должны следовать в порядке возрастания.

```
## uses school;  
  
(174457..174505)  
  .Select( x->x.Divisors )  
  .Where( x->x.Count = 4 )  
  .PrintLines(x->(x[1],x[2]));
```

```
(3,58153)  
(7,24923)  
(59,2957)  
(13,13421)  
(149,1171)  
(5,34897)  
(211,827)  
(2,87251)
```

Чтение чисел из текстового файла в массив

```
##
```

```
var ss:=readlines('17-338.txt');  
var sd:=ss.Select(c->c.tointeger).ToArray;
```

Кто прав: преподаватель или школьник?

Проще или сложнее?

Проще

Есть ли алгоритм?

Алгоритмическое мышление остается и переходит на новый уровень

Новое пугает?

Возможно. Но используется уже везде

Что делать?

Учиться новому

Где пропущена запятая?

Запретить нельзя использовать



demyana@sfedu.ru