Вся правда о лямбда-выражениях

Демяненко Я.М.

Что думают о лямбда-выражениях

Школьники

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

Взрослые

- Кажутся сложными
- Используют профессионалы
- В учебниках нет

https://en.wikipedia.org/wiki/Anonymous_function

- 4.1APL
- 4.2C (non-standard extension)
- 4.3C++ (since C++11)
- 4.4C#
- 4.5ColdFusion Markup Language (CFML)
- 4.6D
- 4.7Dart
- 4.8Delphi
- 4.9PascalABC.NET
- 4.10Elixir
- 4.11Erlang
- 4.12Go
- 4.13Haskell
- 4.14Haxe
- 4.15Java
- 4.16JavaScript
- 4.17Julia
- 4.18Lisp
- 4.19Lua
- 4.20Wolfram Language, Mathematica
- 4.21MATLAB, Octave
- 4.22Maxima
- 4.23ML
- 4.24Next Generation Shell
- 4.25Nim
- 4.26Perl
- 4.27PHP
- 4.28Prolog's dialects
- 4.29Python
- 4.30R
- 4.31Raku
- 4.32Ruby
- 4.33Rust
- 4.34Scala
- 4.35Smalltalk
- 4.36Swift
- 4.37Tcl
- 4.38Vala
- 4.39Visual Basic .NET

Лямбда выражения. Что это за зверь?

```
f(x) = x * x
```

```
f: x -> x * x
```

```
f(x) = x mod 2 = 0
```

```
f: x -> x mod 2 = 0
```

Лямбда-выражение - это функция. Функция отображает параметр на значение

Итак: лямбда-выражения — это просто

```
##

function f(x:integer): integer;
begin
  result:=x*x;
end;

print(f(2))
```

```
##

function f(x:integer) := x*x;

print(f(2))
```

```
##

var f : integer -> integer := x->x*x;

print(f(2))
```

И коротко

```
##

function f(i,j:integer) :=  i*j;

MatrGen(10,10,f).Println;
```

```
##

var f:(integer,integer)->integer:= (i,j)->i*j;

MatrGen(10,10,(i,j)->f(i,j)).Println;
```

Анонимная функция

```
##

MatrGen(10, 10, (i, j)-> i * j).Println;
```

А понятно ли?

Школьники говорят:	понятно!
А мы?	А мы попробуем проверить

Генераторы массивов с лямбда-функциями

Генерация по формуле a[i] = f(i)

```
##
var a := ArrGen(10,i->i*i,1);
a.print
```

```
begin
  var a:=new integer[10];
  for var i:=0 to 9 do
    a[i]:=(i+1)*(i+1);
  a.println
end.
```

Генераторы массивов с лямбда-функциями

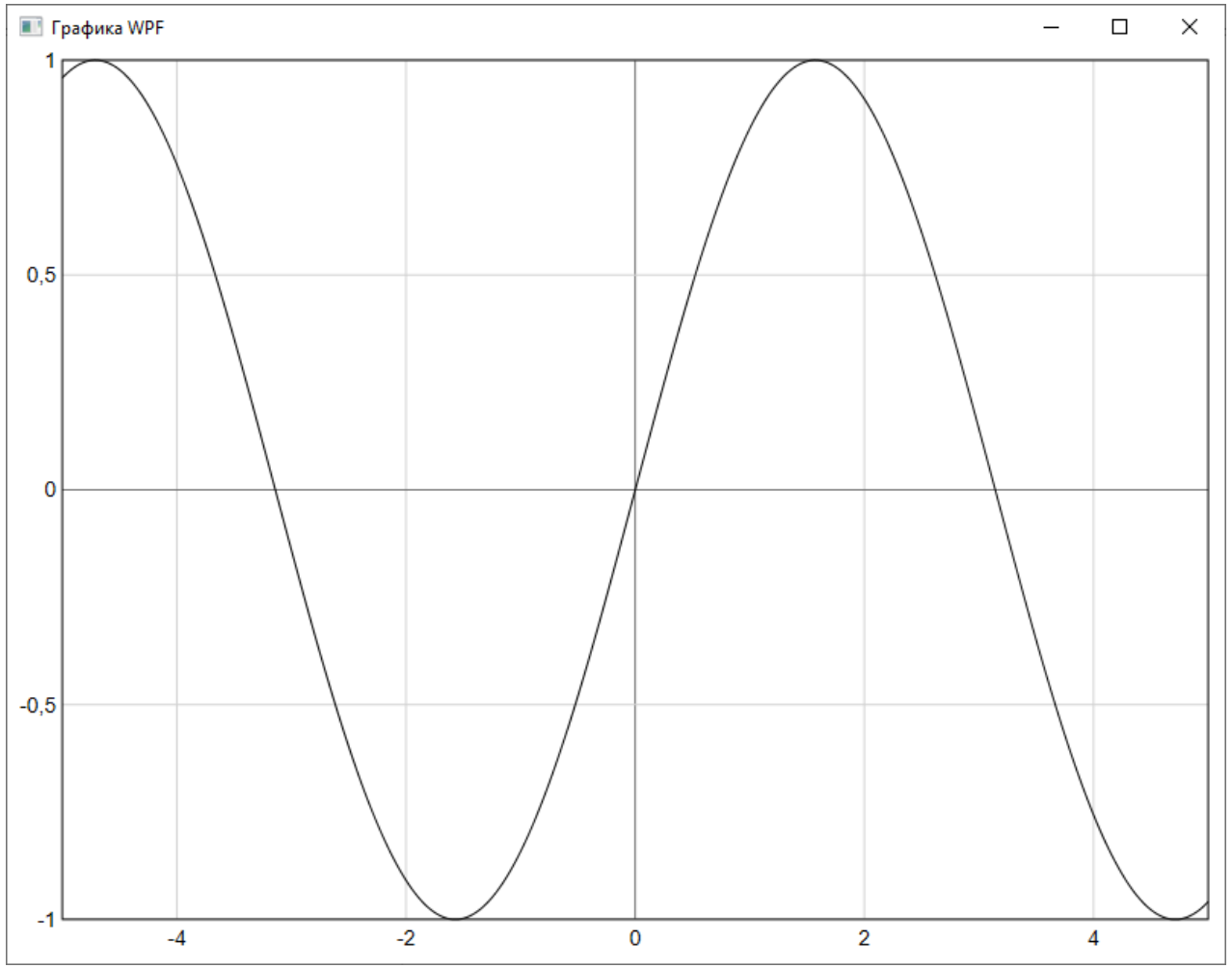
Генерация по рекуррентной формуле

```
##
var a := ArrGen(10,1,x->x*2);
a.print
```

```
begin
  var a:=new integer[10];
  a[0]:=1;
  for var i:=1 to 9 do
    a[i]:=a[i-1]*2;
  a.println
end.
```

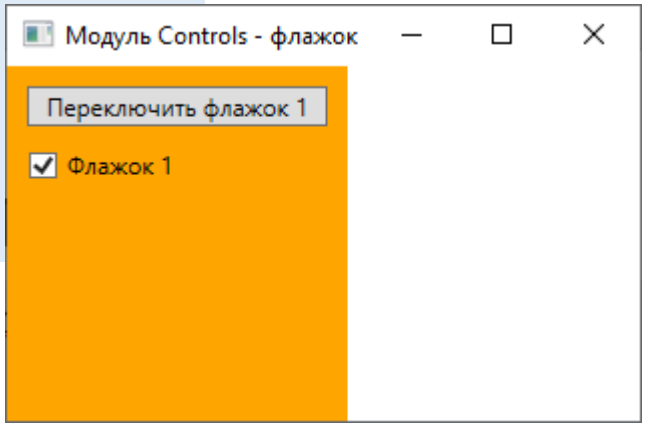
Построить график? Легко!

```
##
uses GraphWPF;
drawgraph(x->sin(x));
```



Лямбды в обработчиках

```
uses GraphWPF,Controls;

begin
  Window.Title := 'Модуль Controls - флажок';

  var p := LeftPanel(170,Colors.Orange);

  var b1 := Button('Переключить флажок 1');

  var cb1 := new CheckBoxWPF('Флажок 1');

  b1.Click := procedure -> cb1.Checked := not cb1.Checked;

end.
```



Кнопки и их обработчики

```
uses GraphWPF,Controls;

begin
  Window.Title := 'Модуль Controls - кнопки и их обработчики';
  var p := LeftPanel;
  p.Color := Colors.Orange;

  Button('Случайный цвет').Click := () -> begin
    Window.Clear(RandomColor);
  end;


  var b1 := Button('Закрыть окно');
  b1.Click := procedure -> Window.Close;
end.
```

Лямбды в обработчиках клавиатуры

```
uses WPFObjects,Controls;

begin
  Window.Title := 'Управление клавиатурой';

  var c := new CircleWPF(GraphWindow.Center,30,Colors.Blue);

  OnKeyDown := k -> begin
    case k of
      Key.Left: c.MoveBy(-2,0);
      Key.Right: c.MoveBy(2,0);
      Key.Up: c.MoveBy(0,-2);
      Key.Down: c.MoveBy(0,2);
    end;
  end;

end.
```

Лямбда-функции (результат boolean)

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

```
1 3 5 7 9 11 13 15 17 19
3
```

Сколько простых в [2, 20000]?

```
##
uses school;
(2..20000).Count( x->x.IsPrime ).Println
```

```
2262
```

Лямбды и запросы

- Запросы. А это что такое?

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
```

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)
```

Захват переменных в лямбда-выражении

```
begin
  var a := Seq(2,3,4);
  var z := 1;
  var q := a.Select(x->x+z);
  q.Println;
  z := 2;
  q.Println;
  z := 3;
  q.Println;
end.
```

```
3 4 5
4 5 6
5 6 7
```

Здесь лямбда-выражение x->x+z захватывает внешнюю переменную z.
Важно заметить, что при изменении значения переменной z запрос a.Select(x->x+z), хранящийся в переменной q, выполняется с новым значением z.

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.
```

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])
```

Задача из ЕГЭ

```
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));
  // Количество элементов этой последовательности и ее максимальный элемент
  Print(seq.Count,seq.Max);
end.
```

Проще или сложнее?
Есть ли алгоритм?
Нет привычной реализации?

Задачи на логические выражения

Для какого наибольшего натурального числа A формула
¬ДЕЛ(x, A) → (ДЕЛ(x, 6) → ¬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;
```

Можно пойти ещё дальше

Логическая функция *F* задаётся выражением $((x \wedge \neg y) \vee (w \rightarrow z)) \equiv (z \equiv x)$
На рисунке приведён частично заполненный фрагмент таблицы истинности функции *F*, содержащий **неповторяющиеся строки**. Определите, какому столбцу таблицы истинности функции *F* соответствует каждая из переменных *x*, *y*, *z*, *w*.

?	?	?	?	F
	0	0	1	1
0	1	0	0	1
0			1	1

```
##
uses school;
var tbl := TrueTable(
        (x,y,z,w) ->
        ((x and not y) or (w <= z))=(z=x));
TrueTablePrint( tbl, 1 , 'xyzw');
```

```
x y z w F
---------
0 0 0 0 1
0 1 0 0 1
1 0 1 0 1
1 0 1 1 1
1 1 0 1 1
1 1 1 0 1
1 1 1 1 1
```

zywx

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
```

А понятно ли?

Школьники говорят: А мы?	понятно!

demyana@sfedu.ru