Lecture #9

Основы программирования
Последовательности

Основы алгоритмизации и программирования

Mayer Svetlana Fyodorovna

Методы Where, Select, OrderBy

Where, Select, OrderBy – являются основными методами для запросов к последовательностям (из языка SQL).

a.Where(условие) – фильтрация последовательности

a.Select(lambda) – проекция последовательности элементов

a.OrderBy(проекция по ключу) – сортировка последовательности

Каждый из запросов возвращает последовательность

Цепочки последовательностей

```
begin
  var a := Arr(7,5,3,2,9,6,15,4);
  var q := a.Where(x -> x mod 2 = 0).Select(x -> x*x).OrderBy(x -> x);
  q.Println; // 4 16 36
end.
```

«Ленивые» запросы

Запросы к последовательностям являются ленивыми:

```
var a := Arr(7,5,3,2,9,6,15,4);
var q := a.Where(x -> x mod 2 = 0).Select(x -> x*x).OrderBy(x -> x);
```

Запрос q начинает выполнение после :

```
q.Println; // 4 16 36
foreach var x in q do
  Print(x); // 4 16 36
a := q.ToArray;
var l := q.ToList;
var s := q.Sum;
```

Примеры

Задача 1. Является ли последовательность симметричной?

```
var a := Arr(7,5,3,2,9,6,15,4);
print(a.SequenceEqual(a.Reverse)); // false
```

Задача 2. Все ли элементы последовательности уникальны?

```
print(a.SequenceEqual(a.Distinct)); // true
```

Задача 3. n

```
var n2 := n div 2;
a.Take(n2).SequenceEqual(a.Skip(n2));
```

Задача 4. Второй максимум в последовательности

```
a.OrderDescending.Distinct.Skip(1).First;
```

Использование Агрегации

Агрегация – это наиболее распространенный метод сворачивания последовательности в скалярную

Расчет произведения:

```
var a := Arr(3,5,2);
var p := a.Aggregate(1,(p,x)->p*x); // 30
```

Перевод в строку:

```
var a := Arr(3,5,2);
var s := a.Aggregate('',(s,x)->s + x + ' '); // '3 5 2'
```

Диапазон в качестве альтернативы циклу for

Генератор диапазона (a, b) может использоваться вместо цикла for

Является ли x простым числом? Все ли i=2…x-1 - не делители числа x?

```
var x := 17;
Range(2,x-1).All(i -> x mod i <> 0).Println;
```

Сумма квадратов нечетных двухразрядных чисел:

```
Range(11,99,2).Sum(x -> x*x).Println;
```

Табуляция функции

```
PartitionPoints(1,2,10).PrintLines(x -> $'{x,4:f1} {Sin(x),7:f4}');
```

```
 1.0  0.8415
 1.1  0.8912
 1.2  0.9320
 1.3  0.9636
 1.4  0.9854
 1.5  0.9975
 1.6  0.9996
 1.7  0.9917
 1.8  0.9738
 1.9  0.9463
 2.0  0.9093
```

Q & A