

ЗАДАЧИ ЗА ЗАДЪЛЖИТЕЛНА
САМОПОДГОТОВКА
ПО
Структури от данни и програмиране
“Реактивни” безкрайни потоци

email: kalin@fmi.uni-sofia.bg

21 ноември 2017 г.

ВНИМАНИЕ: Решението на следите задачи е публикувано към кода от лекции. Разгледайте решенията, осмислете ги и ги пресъздайте самостоятелно (или предложете друго решение на задачата за реализация на операции над безкрайни потоци).

1. Към класа **StreamBase** от реализираната на лекции йерархия да се добави метод за печатане на първите n елемента от потока. Методът да връща остатъка от потока.

Следният пример отпечатва първите 5 нечетни числа:

```
ints.filter(odd).print(2).print(3);
```

2. Да се дефинира поток **RepeatStream** състоящ се от безкрайно повторение на дадено число.

Следният пример отпечатва 5 единици:

```
RepeatStream ones(1);  
ones.print(5);
```

3. Да се дефинира клас **SumStream**, който позволява по дадени два потока **A** и **B** да се генерира нов поток, всеки от елементите на който е сумата (получена с оператора $+$) на двата съответни елемента на **A** и **B**.

Следният пример отпечатва първите 5 четни числа:

```
ints.sum(ints).print (5);
```

Следният пример отпечатва 5 двойки:

```
ones.sum(ones).print (5);
```

4. Да се дефинира клас `ZipStream`, който позволява по дадени два потока $A = (a_i)$ и $B = (b_i)$ да се генерира нов поток $C = (c_i)$, всеки от елементите на който е резултат от приложението на някаква функция $f : int \times int \rightarrow int$ над двата съответни елемента на A и B, т.е. $c_i = f(a_i, b_i)$.

Следният пример отпечатва първите 5 точни квадрата:

```
ints.zip(ints,mult).print (5);
```

Където $mult(x, y) = x * y$

5. Какво трябва да се промени в йерархията така, че да може да се конструират потоци с елементи, различни от *int*?