

# Scheme

Кевролетин В.В. 236гр.

1 июня 2011 г.

## Задание40

### Условие

Последовательность Хэмминга двумя способами.

### Решение

1 способ - отфильтровать числа, которые делятся не только на 2, 3, 5:

```
(define integers (cons-stream 1 (add-streams ones integers)))

(define (is-hamming? x)
  (cond ((= x 1) #t)
        ((= 0 (remainder x 2)) (is-hamming? (/ x 2)))
        ((= 0 (remainder x 3)) (is-hamming? (/ x 3)))
        ((= 0 (remainder x 5)) (is-hamming? (/ x 5)))
        (else #f)))

(define S (stream-filter is-hamming? integers))
```

2 способ - получать числа путём многократного умножения, начиная с 1, полученных чисел на 2, 3 и 5. Основная сложность - выводить числа без повторений и в порядке возрастания. Для этого применим потоки следующим образом:

```
(define (scale-stream stream factor)
  (stream-map (lambda (x) (* x factor)) stream))

(define (merge s1 s2)
  (cond ((stream-null? s1) s2)
        ((stream-null? s2) s1)
        (else
         (let ((s1car (stream-car s1))
               (s2car (stream-car s2)))
           (cond ((< s1car s2car)
                  (cons-stream s1car (merge (stream-cdr s1) s2)))
                 ((> s1car s2car)
                  (cons-stream s2car (merge s1 (stream-cdr s2))))
                 (else
                  (cons-stream s1car (merge s1 (stream-cdr s1)))))))))
```

```

                                (cons-stream s1 car
                                              (merge (stream-cdr s1)
                                                    (stream-cdr s2)))))))))

(define S (cons-stream 1 (merge
                          (scale-stream S 2)
                          (merge
                           (scale-stream S 3)
                           (scale-stream S 5))))))

```