

# Scheme

Кевролетин В.В. 236гр.

24 марта 2011 г.

## Задание13

### Условие

Если мы определим следующую процедуру:

```
(define (f g)
  (g 2))
```

Что случится, если мы попробуем выполнить следующее выражение:

```
(f f)
```

### Решение

Произойдёт ошибка, потому что `f` принимает в качестве аргумента функцию 1 аргумента, и применяет её к числу.

$(f f) \rightarrow (f 2) \rightarrow (2 2) \rightarrow \text{ошибка}$

## Задание14

### Условие

`last-pair`

### Решение

```
(define (last-pair list)
  (if (or (null? list) (null? (cdr list))) list
      (last-pair (cdr list))))
```

```
(last-pair '(1 2 3)) ; (3)
(last-pair '(1))      ; (1)
(last-pair '())       ; ()
```

## Задание15

### Условие

`reverse`

### Решение

```
(define (reverse-rec list res)
  (if (null? list) res
      (reverse-rec (cdr list) (cons (car list) res))))

(define (reverse list)
  (reverse-rec list '()))
```

### Задание16

#### Условие

same-parity

#### Решение

```
(define (same-parity . list)
  (filter
    (lambda (a) (eq? (even? (car list)) (even? a)))
    list))

(same-parity 1 2 3)           ; > (1 3)
(same-parity 2 3 4 5 6)      ; > (2 4 6)
```

### Задание17

#### Условие

deep-reverse, fringe

#### Решение

```
(define (deep-reverse a)
  (if (pair? a)
      (append (deep-reverse (cdr a))
                (list (deep-reverse (car a)))))
      a))

(define x (list (list 1 2) (list 3 4)))
(deep-reverse x)
;; > ((4 3) (2 1))

(define (fringe data)
  (if (pair? data)
      (append (fringe (car data)) (fringe (cdr data)))
      (if (null? data) '()
          (list data))))

(define (fringe data)
  (define (add data res)
    (if (pair? data) (add (cdr data) (add (car data) res))
```

```
(cons data res))  
(add data '()))
```

### **Задание**

#### **Условие**

tree-map

#### **Решение**

```
(define (tree-map f tree)  
  (if (pair? tree)  
      (cons (tree-map f (car tree)) (tree-map f (cdr tree)))  
      (if (null? tree) '()  
          (f tree))))
```