



Министерство науки и высшего образования Российской  
Федерации  
Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Московский государственный технический университет имени  
Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №4  
По курсу: "Функциональное и Логическое программирование"

Студент \_\_\_\_\_ Сукочева Алис  
Группа \_\_\_\_\_ ИУ7-63Б  
Название предприятия \_\_\_\_\_ МГТУ им. Н. Э. Баумана, каф. ИУ7  
Тема \_\_\_\_\_ Использование управляющих структур, работа со списками.

Студент:	_____	Сукочева А.
	подпись, дата	Фамилия, И.О.
Преподаватель:	_____	Толпинская Н.Б.
	подпись, дата	Фамилия, И. О.
Преподаватель:	_____	Строганов Ю.В.
	подпись, дата	Фамилия, И. О.

## Практические задания

Что получится при вычисления каждого из выражений?

```
;; Номер 2
;; (CONS T NIL)
;; Т.к. мы блокируем вычисления 'cons - будет
;; Являться головой списка, а t и nil - хвостом.
(list 'cons t Nil)
;; (list 'cons t Nil) -> (CONS T NIL)
;; (CONS T NIL) -> (T)
;; (T) -> Error Undefined function: T
(eval (eval (list 'cons t Nil)))

;; apply (применить) - принимает два аргумента:
;; 1 - функциональный аргумент -
;; Квотируемое имя функции ('*, '+, 'cons ...)
;; 2 - Список к которому будет применен первый аргумент.
;; Пример:
;; (apply '* '(2 5)) => 10
;; #' (или FUNCTION) - функциональная блокировка.
;; говорит о том, что за ней следует имя функции.
(apply #'cons '(t Nil)) ;; (T)

;; Конструирует список, в котором
;; eval - голова, Nil - хвост.
(list 'eval Nil) ;; (eval Nil)

;; (list 'cons t Nil) -> (cons t Nil)
;; (cons t Nil) -> (t)
(eval (list 'cons t Nil)) ;; (T)

;; Nil
;; См. схему вызова eval.
(eval Nil)

;; (list 'eval Nil) -> (eval Nil)
;; (eval Nil) -> Nil
(eval (list 'eval Nil)) ;; Nil
```

Рисунок 0.1 — Задание 2

Написать функцию, которая переводит температуру в системе Фаренгейта температуру по Цельсию.

```
;; Номер 1
(defun c-to-f (temp) (+ (* 9/5 temp) 32.0))
(defun f-to-c (temp) (* 5/9 (- temp 32.0)))
;; (f-to-c 451) -> 232.77779
;; (c-to-f 232.77779) -> 451.0
```

Рисунок 0.2 — Задание 1

а) Доп 1. Написать функцию, вычисляющую катет по заданной гипотенузе и другому катету прямоугольного треугольника.

б) Доп 2. Написать функцию, вычисляющую площадь трапеции по ее основаниям и высоте.

```
;; Номер 3 (Доп. 1)
;; Дано: а - катет, с - гипотенуза
;; Найти второй катет b.
(defun cathetus (c a) (sqrt (- (* c c) (* a a))))
;; (CATHETUS 10 6) -> 8
```

```
;; Номер 4 (Доп. 2)
;; Дано: а, b, h
(defun S (a b h) (* (+ a b) h 0.5))
;; (s 2 3 1) -> 2.5
```

Рисунок 0.3 — Задание 3

## Ответы на вопросы

### Синтаксическая форма и хранение программы в памяти

В Lisp формы представления программы и обрабатываемых ею данных представляется в виде S-выражения. В памяти представляется как атом (5 указателей) или точечная пара (бинарный узел, 2 указателя)

### Трактовка элементов списка

Первый элемент списка - это имя функции, остальные аргументы.

### Порядок реализации программы

Интерпретатор ожидает ввода S-выражения, после чего предает введенное S-выражение функции eval и выводит полученный результат.

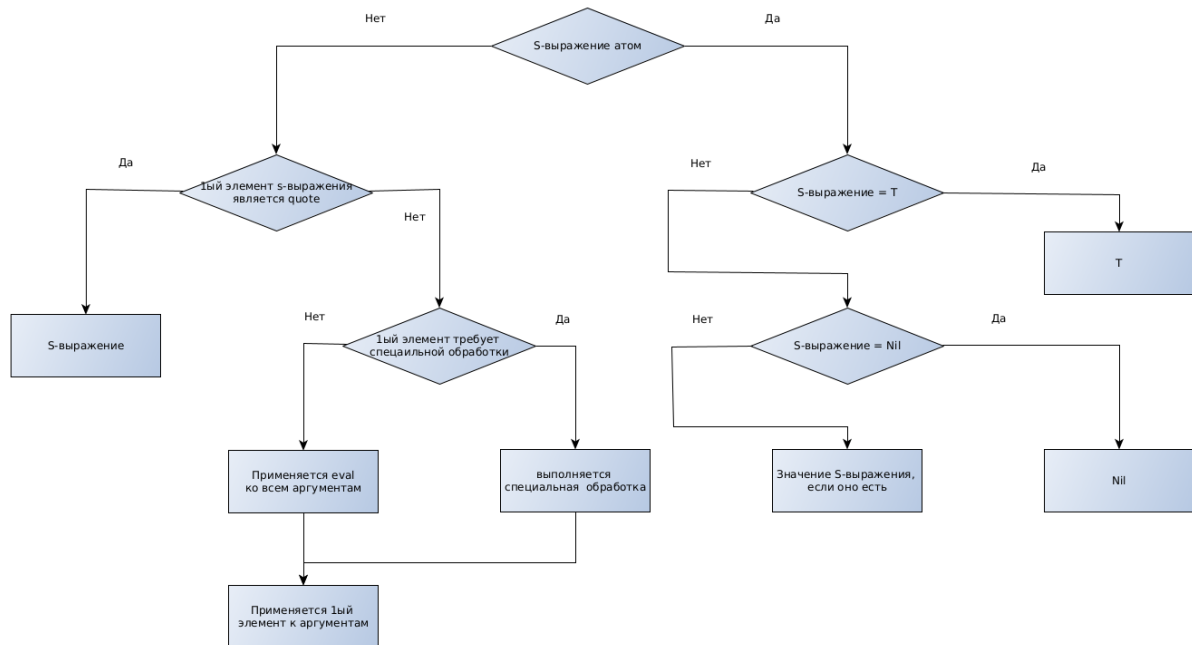


Рисунок 0.1 — Схема работы функции eval

### Способ определения функции

1. С помощью `lambda`.

Базовое определение лямбда-выражения:

```
1 (lambda лямбда-список тело_функции)
```

Пример:

```
1 (lambda (x y) (+ x y))
```

Применение лямбда-выражений:

1 (лямбда-выражение фактические\_параметры)

Пример:

1 ((lambda (x y) (+ x y)) 1 2) ;; 3

2. С помощью `defun`. Для неоднократного применения функции (а также для построения рекурсивной функции) используется встроенная функция `defun`.

Синтаксис:

1 (**defun** имя\_функции лямбда-список тело\_функции)

Пример:

1 (**defun** sum (x y) (+ x y))

Пример вызова:

1 (sum 1 2) ;; 3