



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»

Тема Функции в языке Lisp

Студент Кононенко С.С.

Группа ИУ7-63Б

Оценка (баллы) \_\_\_\_\_

Преподаватели Толпинская Н.Б., Строганов Ю.В.

## Задание 1

**Постановка задачи.** написать функцию, которая переводит температуру в системе Фаренгейта в температуру по Цельсию.

**Решение.**

Листинг 1 – Решение задания 1

```
1 (defun make-far-to-cel (temp)
2   (* (/ 5 9) (- temp 32.0)))
```

## Задание 2

**Постановка задачи.** Что получится при вычислении каждого из выражений?

**Решение.**

Листинг 2 – Решение задания 2

```
1 (list 'cons t NIL) ; (CONS T NIL)
2 (eval (eval (list 'cons t NIL))) ; The function T is undefined, and its name is reserved
   by ANSI CL
3 (apply #'cons '(t NIL)) ; (T)
4 (list 'eval NIL) ; (EVAL NIL)
5 (eval (list 'cons t NIL)) ; (T)
6 (eval NIL) ; NIL
7 (eval (list 'eval NIL)) ; NIL
```

## Задание 3

**Постановка задачи.** Написать функцию, вычисляющую катет по заданной гипотенузе и другому катету прямоугольного треугольника, и составить диаграмму ее вычисления.

**Решение.**

### Листинг 3 – Решение задания 3

```
1 (defun make-cathet (hyp cath)
2   (sqrt (- (* hyp hyp) (* cath cath))))
```

(make-cathet 5 4)

5 вычисляется как 5

4 вычисляется как 4

применяется make-cathet к 5 и 4

(sqrt (- (\* hyp hyp) (\* cath cath)))

создается hyp со значением 5

создается cath со значением 4

(- (\* hyp hyp) (\* cath cath))

(\* hyp hyp)

hyp вычисляется как 5

hyp вычисляется как 5

применяется \* к 5 и 5

25

(\* cath cath)

cath вычисляется как 4

cath вычисляется как 4

применяется \* к 4 и 4

16

применяется - к 25 и 16

9

применяется sqrt к 9

3

3

## Задание 4

**Постановка задачи.** Написать функцию, вычисляющую площадь трапеции по ее основаниям и высоте, и составить диаграмму ее вычисления.

**Решение.**

## Листинг 4 – Решение задания 4

```
1 (defun make-trapezium-square (a b h)
2   (* (/ (+ a b) 2.0) h))
```

(make-trapezium-square 10 5 3)

10 вычисляется как 10

5 вычисляется как 5

3 вычисляется как 3

применяется make-trapezium-square к 10, 5 и 3

(\* (/ (+ a b) 2.0) h)

создается a со значением 10

создается b со значением 5

создается h со значением 3

(/ (+ a b) 2.0)

(+ a b)

a вычисляется как 10

b вычисляется как 5

применяется + к 10 и 5

15

применяется / к 15 и 2.0

7.5

h вычисляется как 3

применяется \* к 7.5 и 3

22.5

22.5

## Ответы на контрольные вопросы

**Вопрос 1.** Синтаксическая форма и хранение программы в памяти.

**Ответ.** В Lisp формы представления программы и обрабатываемых ею данных одинаковы – они представлены в виде S-выражений. Программы могут обрабатывать и преобразовывать другие программы или сами себя. В памяти программа представляется в виде бинарных узлов, так как она состоит из S-выражений.

**Вопрос 2.** Трактовка элементов списка.

**Ответ.** Если отсутствует блокировка вычислений, то первый элемент списка трактуется как имя функции, а остальные элементы – как аргументы функции.

**Вопрос 3.** Порядок реализации программы.

**Ответ.** Работа программы циклична: сначала программа ожидает ввода S-выражения, затем передает полученное S-выражение интерпретатору – функции `eval`, а в конце, после отработки функции `eval`, выводит последний полученный результат.

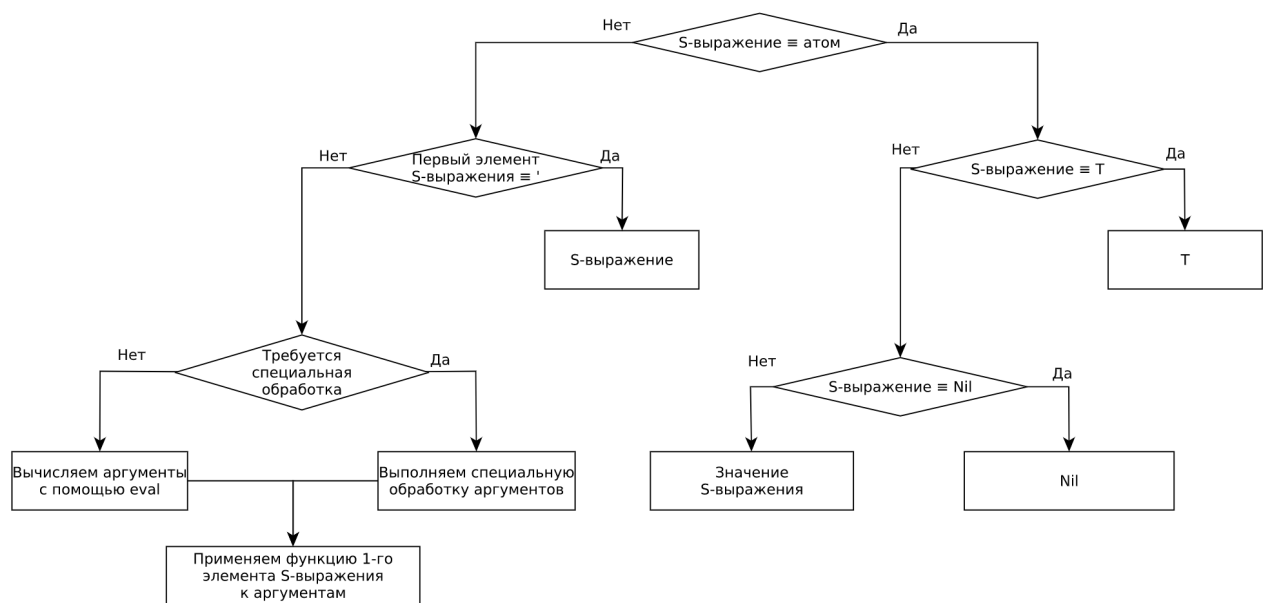


Рисунок 1 – Работа функции `eval`

**Вопрос 4.** Способы определения функции.

**Ответ.** Функцию можно задать через функцию `defun` или `lambda`.

`(defun имя_функции (список_аргументов) тело_функции)`

Например, `(defun sum (x y) (+ x y))`

Вызов функции: `(sum 2 3) -> 5`

`(lambda (список_аргументов) тело_функции)`

Например, `(lambda (x y) (+ x y))`

Вызов функции: `(lambda (x y) (+ x y) 2 3) -> 5`