|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **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**

**Дисциплина: «Функциональное и логическое программирование»**

Выполнила: Овчинникова А.П.

Группа: ИУ7-65Б

Преподаватель: Толпинская Н.Б.

Москва, 2020

**Теоретическая часть.**

Списки также можно рассматривать как последовательности элементов, следующих друг за другом в фиксированном порядке. Длина последовательности определяется с помощью *length.*

Функция *list-length* возвращает количество ячеек в списке или *nil*, если список является циклическим. Не работает с точечными парами, только со списками.

Способы определения функций в Лисп:

* *(defun имя (список\_аргументов) (тело\_функции))*
* *(lambda (список\_аргументов) (тело\_функции)).*

Запустить функции на выполнение можно с помощью функционалов:

* *(apply #'function arg1 … argn)*. Функция *apply* применяет функцию *function* к аргументам *arg1 … argn.*
* *(funcall #’function (список\_аргументов)).*

Функция *(eval expression)* вычисляет заданное выражение *expression* и возвращает его значение. Схема алгоритма работы функции *eval* представлен на рисунке 1.

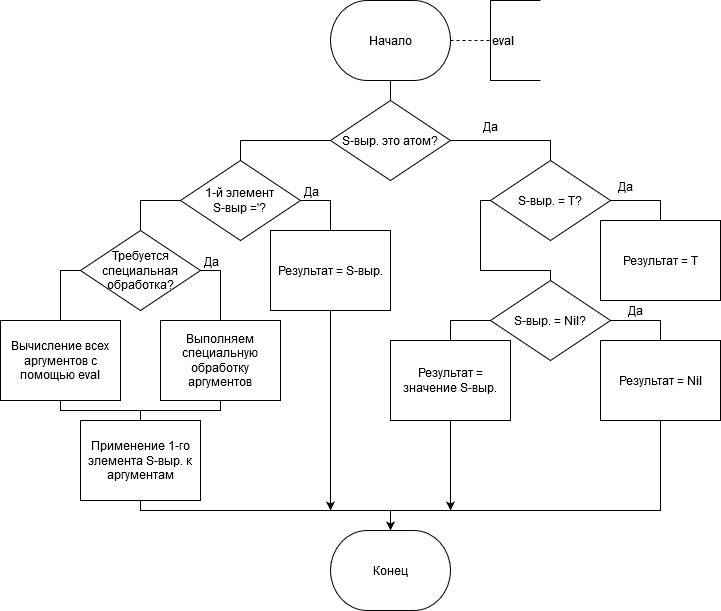


Рис. 1. Схема алгоритма работы функции *eval*.

**Практическая часть.**

**Задание 1.**

Написать функцию longer\_than от двух списков-аргументов, которая возвращает *T*, если первый аргумент имеет большую длину.

*(defun longer\_than (l1 l2)  
 (> (list-length l1) (list-length l2)))*

**Задание 2.**

*(cons 3 (list 5 6)) = (3 5 6)*

*(list 3 ‘from 9 ‘gives (- 9 3)) = (3 FROM 9 GIVES 6)*

*(+ (length ‘(1 foo 2 too)) (car ‘(21 22 23))) = 25*

*(cdr ‘(cons is short for ans)) = (IS SHORT FOR ANS)*

*(car (list one two)) = Error. One* не имеет значения.

*(cons 3 ‘(list 5 6)) = (3 LIST 5 6)*

*(car (list ‘one ‘two)) = ONE*

**Задание 3.**

*(mystery ‘(one two)) = (TWO ONE)*

*(mystery ‘free)) = Error. FREE –* не список.

*(mystery (last ‘one ‘two)) – Error. TWO* не является неотрицательным целым.

*(mystery ‘one ‘two)) – Error.* Слишком много аргументов.

**Задание 4.**

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

*(defun f-to-c (temp)*

*(\* (/ 5.0 9.0) (- temp 32.0)))*

Роман Р. Брэдбери «+451 по Фаренгейту» в системе по Цельсию назывался бы «+232,777779 по Фаренгейту».

**Задание 5.**

*(list ‘cons t Nil) = (CONS T NIL)*

*(eval (eval (list ‘cons t Nil))) = Error.* Функция *T* не определена.

*(apply #’cons ‘(t Nil)) = (T)*

*(list ‘eval Nil) = (EVAL NIL)*

*(eval (list ‘cons t Nil)) = (T)*

*(eval Nil) = NIL*

*(eval (list ‘eval NIL)) = NIL.*

**Задание 6.**

Написать функцию, вычисляющую катет по заданной гипотенузе и другому катету прямоугольного треугольника.

*(defun cathet (cat1 hyp)  
 (sqrt (- (\* hyp hyp) (\* cat1 cat1))))*

Диаграмма вычисления:

(*cathet 3 5)*

Вычисляется 3 к 3

Вычисляется 5 к 5

Функция *cathet* к 3 и 5

Создание переменной *cat1* со значением 3

Создание переменной *hyp* со значением 5

*(sqrt (- (\* hyp hyp) (\* cat1 cat1)))*

*(- (\* hyp hyp) (\* cat1 cat1))*

*(\* hyp hyp)*

Вычисляется *hyp* к 5

Вычисляется *hyp* к 5

Функция \* к 5 и 5

25

*(\* cat1 cat1)*

Вычисляется *cat1* к 3

Вычисляется *cat1* к 3

Функция \* к 3 и 3

9

Функция – к 25 и 9

16

Функция *sqrt* к 16

4

Возврат 4

**Задание 7.**

Написать функцию, вычисляющую площадь трапеции по ее основаниям и высоте.

*(defun trapeze (base1 base2 height)  
 (/ (\* (+ base1 base2) height) 2))*

Диаграмма вычисления:

*(trapeze 2 5 4)*

*(/ (\* (+ base1 base2) height) 2)*

*(\* (+ base1 base2) height)*

*(+ base1 base2)*

Вычисляется *base1* к 2

Вычисляется *base2* к 5

Функция + к 2 и 5

7

Вычисляется *height* к 4

Функция \* к 7 и 4

28

Вычисление 2 к 2

Функция / к 28 и 2

14

Возврат 14