

Московский Государственный Технический Университет им. Н. Э. Баумана

Факультет: Информационные системы и управление

Кафедра: Программное обеспечение ЭВМ и информационные технологии

---

Функциональное и логическое программирование

Отчет по лабораторной работе №4

Выполнил: Зыкин Д.А

Группа: ИУ7-63

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

Москва, 2019

## 1. Теоретическая часть

1.1. Как синтаксически представляется программа в Lisp, и как она хранится в памяти ?

Программа Lisp представляется в виде списка. Хранится в виде списковых ячеек.

1.2. Как трактуются элементы списка ?

Первый элемент списка - имя функции, остальные элементы - аргументы.

Примеры:

- (+ 1 2)
- (+ 1 2 3 4)
- (+)
- (+ 1 (/ 4 2))

1.3. Порядок реализации программы.

Цикл работы Lisp-машины:

1. read S-выражение
2. eval S-выражение
3. print
4. GOTO 1

В любых императивных системах система печатает только по запросу. Lisp работает по указателям, в процессе работы может преобразовывать значения, и на экран всегда выводится последнее вычисленное значение. Не всегда оно может соответствовать тем преобразованиям, которые прошли в памяти.

Схема работы функции eval.

(eval s-выражение)

1. S-выражение атом ? -yes->2 -no->7
2. S-выражение = t ? -yes->3 -no->4
3. печать T \*
4. S-выражение = nil? ->yes->5 -no->6
5. печать nil \*
6. значение s-expr если есть
7. является 1м элементом quote? -yes->8 -no->9
8. s-expr без символа кавычки \*
9. требует особой обработки? -да->12 -нет->10
10. применить eval ко всем элементам S-выражение кроме первого

11. применить первый элемент к остальным
12. провести обработку в зависимости от функции
13. применить первый элемент к остальным

## 2. Практическая часть

### 2.1.1. (equal 3 (abs - 3))

3 вычисляется к 3  
(abs -3)  
-3 вычисляется к -3  
применение функции abs к -3  
возвращается 3  
применение функции equal к 3 и 3  
возвращается T

### 2.1.2. (equal (+ 1 2) 3)

(+ 1 2)  
1 вычисляется к 1  
2 вычисляется к 2  
применение функции + к 1 и 2  
возвращается 3  
3 вычисляется к 3  
применение функции equal к 3 и 3  
возвращается T

### 2.1.3. (equal (\* 4 7) 21)

(\* 4 7)  
4 вычисляется к 4  
7 вычисляется к 7  
применение функции \* к 4 и 7  
возвращается 28  
21 вычисляется к 21  
применение функции equal к 28 и 21  
возвращается nil

### 2.1.4. (equal (\* 2 3) (+ 7 2))

(\* 2 3)  
2 вычисляется к 2  
3 вычисляется к 3  
применяется функция \* к 2 и 3  
возвращается 6  
(+ 7 2)  
7 вычисляется к 7  
2 вычисляется к 2  
применение функции + к 7 и 2

возвращается 9  
применение функции equal к 6 и 9  
возвращается nil

2.1.5. (equal (- 7 3) (\* 3 2))

(- 7 3)  
7 вычисляется к 7  
3 вычисляется к 3  
применение функции - к 7 и 3  
возвращает 4  
(\* 3 2)  
3 вычисляется к 3  
2 вычисляется к 2  
применение функции \* к 3 и 2  
возвращает 6  
применение функции equal к 4 и 6  
возвращает nil

2.1.6. (equal (abs(- 2 4)) 3)

(abs(- 2 4)) 3)  
(- 2 4)  
2 вычисляется к 2  
4 вычисляется к 4  
применение функции - к 2 и 4  
возвращает -2  
error!

2.2. (defun a (c b) (sqrt (- (\* c c) (\* b b))))

(a 10 8)  
10 вычисляется к 10  
8 вычисляется к 8  
вызов функции a с аргументами 10 и 8  
создается переменная c со значением 10  
создается переменная b со значением 8  
(sqrt (- (\* c c) (\* b b)))  
(- (\* c c) (\* b b))  
(\* c c)  
c вычисляется к 10  
c вычисляется к 10  
применяется функция \* к аргументам  
возвращается 100  
(\* b b)  
b вычисляется к 8  
b вычисляется к 8

применяется функция \* к аргументам  
 возвращается 64  
 применяется функция - к аргументам  
 возвращается 36  
 применяется функция sqrt к аргументам  
 возвращается 6  
 возвращается 6

2.3. (defun S(h a b)(\* h (/ 1.0 2.0) (+ a b)))  
 (S 5 1 3)

5 вычисляется к 5  
 1 вычисляется к 1  
 3 вычисляется к 3  
 вызов функции S с аргументами 5, 1, 3  
 создается переменная h со значением 5  
 создается переменная a со значением 1  
 создается переменная b со значением 3  
 (\* h (/ 1.0 2.0) (+ a b))  
 h вычисляется к 5  
 (/ 1.0 2.0)  
 1.0 вычисляется к 1.0  
 2.0 вычисляется к 2.0  
 применение функции / к аргументам  
 возвращается 0.5  
 (+ a b)  
 a вычисляется к 1  
 b вычисляется к 3  
 применяется функция + к аргументам  
 возвращается 4  
 применяется функция \* к аргументам  
 возвращается 10  
 возвращается 10