



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»

## Отчёт по лабораторной работе №3 по курсу «Функциональное и логическое программирование»

Тема Работа интерпретатора Lisp

Студент Богаченко А. Е.

Группа ИУ7-66Б

Оценка (баллы) \_\_\_\_\_

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

# Лабораторная работа №1

## Задание 1

Составить диаграмму вычисления следующих выражений:

Листинг 1 – Задание 1

```
1 (equal 3 (abs -3))  
2 (equal (+ 1 2) 3)  
3 (equal (* 4 7) 21)  
4 (equal (* 2 3) (+ 7 2))  
5 (equal (- 7 3) (* 3 2))  
6 (equal (abs (- 2 4)) 3)
```

1. (equal 3 (abs -3))

3 вычисляется как 3

(abs -3)

-3 вычисляется как -3

abs применяется к -3

3

equal применяется к 3 и 3

T

2. (equal (+ 1 2) 3)

(+ 1 2)

1 вычисляется как 1

2 вычисляется как 2

+ применяется к 1 и 2

3

3 вычисляется как 3

equal применяется к 3 и 3

T

3. (equal (\* 4 7) 21)

(\* 4 7)

4 вычисляется как 4

7 вычисляется как 7  
\* применяется к 4 и 7  
28

21 вычисляется как 21  
equal применяется к 28 и 21  
Nil

4. (equal (\* 2 3) (+ 7 2))

(\* 2 3)

2 вычисляется как 2  
3 вычисляется как 3  
\* применяется к 2 и 3  
6

(+ 7 2)

7 вычисляется как 7  
2 вычисляется как 2  
+ применяется к 7 и 2  
9

equal применяется к 6 и 9  
Nil

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

(- 7 3)

7 вычисляется как 7  
3 вычисляется как 3  
- применяется к 7 и 3  
4

(\* 3 2)

3 вычисляется как 3  
2 вычисляется как 2  
\* применяется к 3 и 2  
6

equal применяется к 4 и 6  
Nil

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

```
(abs (- 2 4))
(- 2 4)
  2 вычисляется как 2
  4 вычисляется как 4
  - применяется к 2 и 4
  -2
abs применяется к -2
2
3 вычисляется как 3
equal применяется к 2 и 3
Nil
```

## Задание 2

Написать функцию, вычисляющую гипотенузу прямоугольного треугольника по заданным катетам и составить диаграмму её вычисления.

Листинг 2 – Задание 2

```
1 (defun calc-triangle-hyp (a b)
2   "Task 2"
3   (let* ((a-square (* a a))
4          (b-square (* b b))
5          (square-sum (+ a-square b-square)))
6     (sqrt s)))
```

```
(a-square (* a a))
(* a a)
  a вычисляется как a
  a вычисляется как a
  * применяется к a и a
   $a^2$ 
привязка  $a^2$  к a-square
(b-square (* b b))
(* b b)
  b вычисляется как b
  b вычисляется как b
```

$*$  применяется к  $b$  и  $b$   
 $b^2$   
 привязка  $b^2$  к  $b$ -square  
 $(\text{square-sum } (+ \text{ a-square } b\text{-square}))$   
 $(+ \text{ a-square } b\text{-square})$   
 $+$  применяется к  $a$ -square и  $b$ -square  
 $a^2 + b^2$   
 привязка  $a^2 + b^2$  к  $\text{square-sum}$   
 $(\text{sqrt square-sum})$   
 $\text{sqrt}$  применяется к  $\text{square-sum}$   
 $\sqrt{a^2 + b^2}$

## Задание 3

Написать функцию, вычисляющую объём параллелепипеда по 3-м его сторонам и составить диаграмму её вычисления.

### Листинг 3 – Задание 3

```

1 (defun calc-par-vol (a b h)
2   "Task 3"
3   (* a b h))

```

$(* \text{ a } \text{ b } \text{ h})$   
 $a$  вычисляется как  $a$   
 $b$  вычисляется как  $b$   
 $h$  вычисляется как  $h$   
 $*$  применяется к  $a$ ,  $b$ ,  $h$   
 $a \cdot b \cdot h$

## Задание 4

Каковы результаты вычисления следующих выражений?

### Листинг 4 – Задание 4

```

1 (list 'a 'b c) ; -> the variable C is unbound

```

```

2 (cons 'a (b c)) ; -> the variable C is unbound
3 (cons 'a '(b c)) ; -> (a b c)
4 (caddr (1 2 3 4 5)) ; -> illegal function call
5 (cons 'a 'b 'c) ; -> invalid number of arguments: 3
6 (list 'a (b c)) ; -> the variable C is unbound
7 (list a '(b c)) ; -> the variable A is unbound
8 (list (+ 1 '(length '(1 2 3)))) ; -> the value (LENGTH '(1 2 3)) is not of type NUMBER

```

## Задание 5

Написать функцию `longer-than` от двух списков-аргументов, которая возвращает Т, если первый аргумент имеет большую длину.

Листинг 5 – Задание 5

```

1 (defun longer-than (a b)
2   "Task 5"
3   (let ((len-a (length a))
4         (len-b (length b)))
5     (> len-a len-b)))

```

## Задание 6

Каковы результаты вычисления следующих выражений?

Листинг 6 – Задание 6

```

1 (cons 3 (list 5 6)) ; -> (3 5 6)
2 (list 3 'from 9 'gives (- 9 3)) ; -> (3 from 9 gives 6)
3 (+ (length '(1 foo 2 too)) (car '(21 22 23))) ; -> 25
4 (cdr '(cons is short for ans)) ; -> (is short for ans)
5 (car (list one two)) ; -> variable ONE is unbound
6 (cons 3 '(list 5 6)) ; -> (3 list 5 6)
7 (car (list 'one 'two)) ; -> one

```

## Задание 7

Дана функция

## Листинг 7 – mystery

```
1 (defun mystery (x)
2   (list (second x) (first x)))
```

Каковы результаты вычисления следующих выражений?

## Листинг 8 – Задание 7

```
1 (mystery '(one two)) ; -> (two one)
2 (mystery 'free) ; -> the value FREE is not of type LIST
3 (mystery (last 'one 'two)) ; -> the value ONE is not of type LIST when binding LIST
4 (mystery 'one 'two) ; -> invalid number of arguments: 2
```

## Задание 8

Написать функцию, которая переводит температуру в системе Фаренгейта в температуру по Цельсию.  $f = \frac{9}{5} \cdot c + 32$

## Листинг 9 – Задание 8

```
1 (defun f-to-c (temp)
2   "Task 7"
3   (float (* (- temp 32) (/ 5 9))))
```

Как бы назывался роман Р. Бредбери «+451 по Фаренгейту» в системе по Цельсию?

– «232.78 по Цельсию»

## Задание 9

Что получится при вычислении каждого из выражений?

## Листинг 10 – Задание 9

```
1 (list 'cons T Nil) ; -> (cons T Nil)
2 (eval (eval (list 'cons T Nil))) ; -> undefined function
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
```

## Дополнительное задание 1

Написать функцию, вычисляющую катет по заданной гипотенузе и другому катету прямоугольного треугольника и составить диаграмму её вычисления.

Листинг 11 – Дополнительное задание 1

```
1 (defun find-triangle-side (hyp side)
2   "Extra 1"
3   (let* ((hyp-square (* hyp hyp))
4          (side-square (* side side))
5          (square-sub (- hyp-square side-square)))
6     (sqrt square-sub)))
```

(hyp-square (\* hyp hyp))

  (\* hyp hyp)

    hyp вычисляется как hyp

    hyp вычисляется как hyp

    \* применяется к hyp и hyp

$hyp^2$

  привязка  $hyp^2$  к hyp-square

(side-square (\* side side))

  (\* side side)

    side вычисляется как side

    side вычисляется как side

    \* применяется к side и side

$side^2$

  привязка  $side^2$  к side-square

(square-sub (- hyp-square side-square)))

  (- hyp-square side-square)

    - применяется к hyp-square и side-square

$hyp^2 - side^2$

  привязка  $hyp^2 - side^2$  к square-sub

(sqrt square-sub)

  sqrt применяется к square-sub

$\sqrt{hyp^2 - side^2}$



# Контрольные вопросы

## 1. Базис языка Lisp

Базис языка представлен:

- структурами, атомами;
- функциями:  
atom, eq, cons, car, cdr,  
cond, quote, lambda, eval, label.

## 2. Классификация функций языка Lisp

Функции в языке Lisp:

- чистые (с фиксированным количеством аргументов) – математические функции;
- рекурсивные функции;
- специальные функции – формы (принимают произвольное количество аргументов или по разному обрабатывают аргументы);
- псевдофункции (создающие «эффект» - отображающие на экране процесс обработки данных и т. п.);
- функции с вариативными значениями, выбирающие одно значение;
- функции высших порядков – функционалы (используются для построения синтаксически управляемых программ).

### 3. Синтаксис элементов языка и их представление в памяти

Точечные пары ::= (<атом>, <атом>) |

(<атом>, <точечная пара>) |

(<точечная пара>, <атом>) |

(<точечная пара>, <точечная пара>)

Список ::= <пустой список> | <непустой список>), где

<пустой список> ::= () | Nil,

<непустой список> ::= (<первый элемент>, <хвост>) ,

<первый элемент> ::= (S-выражение),

<хвост> ::= <список>

Список – частный случай S-выражения.

Синтаксически любая структура (точечная пара или список) заключается в ():

(A . B) – точечная пара

(A) – список из одного элемента

Пустой список изображается как Nil или ()

Непустой список может быть изображён: (A. (B . (C ()))) или (A B C)

Элементы списка могут являться списками: ((A) (B) (C))

Любая непустая структура `Lisp` в памяти представлена списковой ячейкой, хранящей два указателя: на голову (первый элемент) и хвост (всё остальное).

#### 4. Функции `car`, `cdr`

Являются базовыми функциями доступа к данным. `car` принимает точечную пару или список в качестве аргумента и возвращает первый элемент или `Nil`, `cdr` – возвращает все элементы, кроме первого или `Nil`.

#### 5. Функции `list`, `cons`

Являются функциями создания списков (`cons` – базовая, `list` – нет). `cons` создаёт списочную ячейку и устанавливает два указателя на аргументы. `list` принимает переменное число аргументов и возвращает список, элементами которого являются аргументы, переданные в функцию.