



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 №3 по курсу «Функциональное и логическое программирование»

Тема Работа интерпретатора Lisp

Студент Кононенко С.С.

Группа ИУ7-63Б

Оценка (баллы) \_\_\_\_\_

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

# Задание 1

**Постановка задачи.** Составить диаграмму вычисления следующих выражений:

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)`

**Решение.**

1. `(equal 3 (abs -3))`  
3 вычисляется как 3  
(abs -3)  
-3 вычисляется как -3  
применяется abs к -3  
3  
применяется equal 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

применяется equal 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

применяется equal 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

применяется equal 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

применяется equal 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

применяется abs к -2

2

3 вычисляется как 3

применяется equal к 2 и 3

NIL

## Задание 2

**Постановка задачи.** Написать функцию, вычисляющую гипотенузу прямоугольного треугольника по заданным катетам и составить диаграмму ее вычисления.

**Решение.**

Листинг 1 – Решение задания 2

```
1 (defun make-hypot (x y)
2   (sqrt (+ (* x x) (* y y))))
```

(sqrt (+ (\* x x) (\* y y)))

(+ (\* x x) (\* y y))

(\* x x)

х вычисляется как х

х вычисляется как х

применяется \* к х и х

$x^2$

(\* y y)

у вычисляется как у

у вычисляется как у

применяется \* к у и у

$y^2$

применяется + к  $x^2$  и  $y^2$

$x^2 + y^2$

применяется sqrt к  $x^2 + y^2$

$\sqrt{x^2 + y^2}$

## Задание 3

**Постановка задачи.** Написать функцию, вычисляющую объем параллелепипеда по 3-м его сторонам, и составить диаграмму ее вычисления.

**Решение.**

### Листинг 2 – Решение задания 3

```
1 (defun make-par-volume (x y z)
2   (* x y z))
```

`(* x y z)`

`x` вычисляется как `x`

`y` вычисляется как `y`

`z` вычисляется как `z`

`*` применяется к `x`, `y`, `z`

$x \times y \times z$

## Задание 4

**Постановка задачи.** Каковы результаты вычисления следующих выражений?

**Решение.**

### Листинг 3 – Решение задания 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)) ; INVALID FUNCTION CALL
5 (cons 'a 'b 'c) ; INVALID NUMBER OF ARGUMENTS
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)))) ; (LENGTH '(1 2 3)) is not of type NUMBER
```

## Задание 5

**Постановка задачи.** Написать функцию `longer_then` от двух списков-аргументов, которая возвращает Т, если первый аргумент имеет большую длину.

**Решение.**

### Листинг 4 – Решение задания 5

```
1 (defun is-longer-than (l1 l2)
2   (> (length l1) (length l2)))
```

## Задание 6

**Постановка задачи.** Каковы результаты вычисления следующих выражений?

**Решение.**

### Листинг 5 – Решение задания 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

**Постановка задачи.** Дана функция (defun mystery (x) (list (second x) (first x))). Какие результаты вычисления следующих выражений?

**Решение.**

### Листинг 6 – Решение задания 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
```

## Ответы на контрольные вопросы

**Вопрос 1.** Базис языка Lisp.

**Ответ.** Базис языка представлен:

- структурами, атомами;

- Функциями:

atom, eq, cons, car, cdr;  
cond, quote, lambda, eval, label.

**Вопрос 2.** Классификация функций языка Lisp.

**Ответ.** 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);

**Вопрос 3.** Синтаксис элементов языка и их представление в памяти.

**Ответ.**

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 круглые скобки:



(A . B) – точечная пара.

(A) – список из одного элемента.

Пустой список изображается как Nil или ().

Непустой список может быть изображен: (A . (B . (C . (D ())))),  
(A B C D)

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

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

**Вопрос 4.** Функции `car`, `cdr`.

**Ответ.** Функции `car`, `cdr` являются базовыми функциями доступа к данным. `car` принимает точечную пару или список в качестве аргумента и возвращает первый элемент или Nil, соответственно. `cdr` принимает точечную пару или список в качестве аргумента и возвращает все элементы кроме первого или Nil, соответственно.

**Вопрос 5.** Функции `list`, `cons`.

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