



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 дисциплине «Функциональное и логическое  
программирование»**

Тема Работа интерпретатора Lisp

Студент Романов А.В.

Группа ИУ7-63Б

Оценка (баллы)                     

Преподаватель Толпинская Н.Б.

# 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

Решение оформлено на тетрадном листке бумаге. К отчету прилагается.

# Задание 2

## Постановка задачи

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

## Решение

Листинг 1: Решение задания №2

```
1 (defun hypot (x y) (sqrt (+ (* x x) (* y y))))
```

# 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 volume (x y z) (* x y 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; (list 'a 'b 'c) -> (ABC)
2 (cons 'a (b c)); THE VARIABLE C IS UNBOUND; (cons 'a '(bc)) -> (ABC)
3 (cons 'a '(b c)) -> (A B C)
4 (caddr (1 2 3 4 5)) -> 3
5 (cons 'a 'b 'c); INVALID NUMBER OF ARGUMENTS; (cons 'a 'b) -> (A . B)
6 (list 'a (b c)); THE VARIABLE C IS UNBOUND; (list 'a '(b c)) -> (A (BC))
7 (list a '(b c)); THE VARIABLE A IS UNBOUND; (list 'a '(b c)) -> (A (BC))
8 (list (+ 1 '(length '(1 2 3)))) ; (LENGTH '(1 2 3)) is not of type NUMBER; (
  list (+1 (length '(123)))) -> 4
```

## 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 longer_than (l1 l2) (> (length l1) (length l2)))
```

## Задание 6

### Постановка задачи

Каковы результаты вычисления следующих выражений?

### Решение

Листинг 5: Решение задания №6

```
1 (cons 3 (list 56)) -> (356)
2 (cons 3 '(list 56)) -> (3 LIST 56)
3 (list 3 'from 8 'gives(-9 3)) -> 3 FROM 9 GIVES 6
```

```

4 (+ (length '(1 foo 2 too)) (car '(21 22 23))) -> 25
5 (cdr '(cons is short for ans)) -> (IS SHORT FOR ANS)
6 (car (list one two)); VARIABLE ONE IS UNBOUND; (car (list 'one 'two)); ->
  ONE
7 (cons 3 '(list 5 6)) -> (3 LIST 5 6)
8 (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; (mystery '(free)) -> (
  NIL FREE)
3 (mystery (last 'one 'two)); The value ONE is not of type LIST when binding
  LIST; (mystery (last ('one 'two))) -> (NIL TWO)
4 (mystery 'one 'two); INVALID NUMBER OF ARGUMENTS: 2; (mystery '(one two)) ->
  (TWO ONE)

```

## Контрольные вопросы

**Вопрос 1.** Базис языка Lisp.

**Ответ.** Базис языка представлен:

- структурами и атомами;
- функциями;

Функции, входящие в базис языка:

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

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

**Ответ.**

- 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. Любая структура (точечная пара или список) заключаются в круглые скобки:

- (A . B) – точечная пара;
- (A) – список из одного элемента;
- Nil или () – пустой список;
- (A . (B . (C . (D ()))) или (A B C D) – непустой список;
- Элементы списка могут являться списками: ((A)(B)(CD))

Любая непустая структура в Lisp, в памяти представлена списковой ячейкой, хранящей два указателя: на голову и хвост.

**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.