



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

*к лабораторной работе №3*

*По курсу: «Функциональное и логическое  
программирование»*

Студентка ИУ7-66Б  
Барсуков Н.М.

Преподаватель  
Строганов Ю.В.

2020 г.

**Цель работы:** приобрести навыки использования стандартных функций Lisp.

**Базис** – это минимально необходимый набор конструкций, с помощью которого можно реализовать задачу.

### **Классификация функций:**

- Базовые функции
  - Селекторы (car, cdr)
  - Конструкторы (cons)
  - Предикаты (atom, Null, lisp, ..)
  - Функции сравнения (eq, eql, =, equal, equalp)
- Формы
- Функционалы

**Списки в ОП** представлены с помощью списковых ячеек. В списковой ячейке хранится два указателя: на голову и хвост.

**CAR и CDR** являются базовыми функциями доступа к данным. CAR принимает точечную пару или пустой список в качестве аргумента и возвращает первый элемент или nil, соответственно. CDR принимает точечную пару или пустой список и возвращает список состоящий из всех элементов, кроме первого. Если в списке меньше двух элементов, то возвращается Nil.

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

Например список '(open close halph) из задания 1 можно представить как: (cons 'open (cons 'close (cons 'halph nil))) или (list 'open 'close 'halph).

### 1. Составить диаграмму вычисления следующих выражений:

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

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

$$c = \sqrt{a^2 + b^2}$$

```
(defun hyp (a b)  
  (sqrt (+ (* a a) (* b b))))  
  
> (hyp 3 4) -> 5.0
```

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

$$V = a * b * h$$

```
(defun vol (a b h)  
  (* a b h))  
  
> (vol 2 3 4) -> 24
```

### 4. Каковы результаты вычисления следующих выражений?

#### Возможные ошибки:

The variable C is unbound – возникает при попытке получить значение символа не связанного со значением.

The function B is undefined – не была определена вызываемая функция.

Invalid number of arguments – в функцию было передано неверное количество аргументов.

Illegal function call – возникает при попытке в качестве имени функции передать не символ (например если написать число или список).

```
(list 'a c) -> The variable C is unbound.
```

(list 'a 'c) -> (a c)

(cons'a'b'c) -> invalid number of arguments: 3

(cons'a'b) -> (a . b)

(cons'a (b c)) -> The variable C is unbound.

(cons 'a ' (b c)) -> (a b c)

(list 'a (b c)) -> The variable C is unbound.

(list 'a (b 'c)) -> The function COMMON-LISP-USER::B is undefined.

(list 'a '(b c)) -> (a (b c))

(cons 'a '(b c)) -> (a b c)

(list a '(b c)) -> The variable A is unbound.

(list 'a '(b c)) -> (a (b c))

(caddy (1 2 3 4 5)) -> Execution of a form compiled with errors. Form: (1 2 3 4 5)

Compile-time error: illegal function call

(caddy ' (1 2 3 4 5)) -> The function COMMON-LISP-USER::CADDY is undefined.

(caddr '(1 2 3 4 5)) -> 3

(list (+ 1 '(length '(1 2 3)))) -> The value (LENGTH '(1 2 3)) is not of type NUMBER

(list (+ 1 (length '(1 2 3)))) -> (4)

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

(defun longer\_then (a1 a2)

(> (length a1) (length a2)))

> (longer\_then '(1 2 3 4) '(1 2)) -> T

> (longer\_then '(1 2 3 4) '(1 2 3 4 5)) -> NIL

```
(defun longer_then2 (a1 a2)

  (cond ((null a2) (null (null a1)))

  (T (longer_then2 (cdr a1) (cdr a2))))

)
```

## 6. Каковы результаты вычисления следующих выражений?

```
(cons 3 (list 5 6)) -> (3 5 6)
(cons 3 '(list 5 6)) -> (3 list 5 6)
(list 3 'from 9 'lives (- 9 3)) -> (3 from 9 lives 6)
(+ (length for 2 too)) -> The variable FOR is unbound
(+ (length '(for 2 too))) -> 3
(car '(21 22 23))) -> unmatched close parenthesis
(car '(21 22 23)) -> 21
(cdr ' (cons is short for ans)) -> (is short for ans)
(car (list one two)) -> The variable one is unbound
(car (list 'one 'two)) -> one
```

## 7. Дана функция (defun mystery (x) (list (second x) (first x))). Какие результаты вычисления следующих выражений?

```
(mystery (one two)) -> The variable TWO is unbound.
(mystery (one 'two)) -> The function COMMON-LISP-USER::ONE is undefined.
(mystery '(one two)) -> (two one)
(mystery one 'two)) -> The variable ONE is unbound.
(mystery 'one 'two)) -> invalid number of arguments: 2
(mystery ('one 'two)) -> illegal function call
(mystery (last one two)) -> The variable ONE is unbound.
(mystery '(last one two)) -> (one last)
(mystery free) -> The variable FREE is unbound.
(mystery 'free) -> The value FREE is not of type LIST
(mystery '(free)) -> (nil free)
```