**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"**

Студент Сукочева Алис
Группа ИУ7-63Б
Название предприятия МГТУ им. Н. Э. Баумана, каф. ИУ7
Тема Работа интерпретатора Lisp.

| | | |
|---|---|---|
| Студент: | _____ | Сукочева А. |
| | подпись, дата | Фамилия, И.О. |
| Преподаватель: | _____ | Толпинская Н.Б. |
| | подпись, дата | Фамилия, И. О. |
| Преподаватель: | _____ | Строганов Ю.В. |
| | подпись, дата | Фамилия, И. О. |

Москва — 2021 г.

## Практические задания

```
;; № 2
;; Заданы катеты a и b.
;; Вычислить гипотенузу.
(defun hypotenuse (a b) (sqrt (+ (* a a) (* b b))))
;; (HYPOTENUSE 6 8) -> 10
```

Рисунок 0.1 — Задание 2

```
;; № 3
;; Заданы длина a, ширина b и высота c
;; Найти объем.
(defun volume (a b c) (* a (* b c)))
;; (volume 2 3 4) -> 24
```

Рисунок 0.2 — Задание 3

```
;; №4
;; Каковы результаты вычисления следующих выражений?
(list 'a 'b c)      ;; Error (The variable C is unbound).
(cons 'a (b c))     ;; Error undefined function: B
(cons 'a '(b c))    ;; (A B C)
;; Error illegal function call
;; Недопустимый вызов функции, т.к. имя функции не может быть цифрой.
(caddr (1 2 3 4 5))
;; Error The function was called with three arguments, but wants exactly two.
;; cons принимает два аргумента, а мы ей передаем три.
(cons 'a 'b 'c)
;; Error undefined function: B
;; Т.к. мы не поставили quote перед (b c)
;; Lisp воспринимает b как имя функции, а у нас такой функции нет.
(list 'a (b c))
;; Error The variable A is unbound.
;; Т.к. первый аргумент a не является списком,
;; Он проверяется на T и на nil.
;; Далее, т.к. нет совпадений возвращается значения,
;; Но так как у нас нет значения в a, возвращается ошибка.
(list a '(b c))
;; (length '(1 2 3 4)) -> 4
;; + 1 4 -> 5
;; list 5 -> (5)
(list (+ 1 (length '(1 2 3)))) ;; (4)
```

Рисунок 0.3 — Задание 4

```
;; №5
;; Два списка a и b
;; Возвращает T, если длина a больше длины b.
(defun longer_then(a b) (cond ((> (length a) (length b)) T) (Nil)))
;; (LONGER_THEN '(a b c) '(a))          ;; -> T
;; (LONGER_THEN '(a b c) '(a b c))      ;; --> nil
;; (LONGER_THEN '(a b c) '(a b c d))    ;; --> nil
```

Рисунок 0.4 — Задание 5

```
;; №6_1
;; Каковы результаты вычисления следующих выражений?
;; (list 5 6) -> (5 6)
;; (cons 3 '(5 6)) ;; -> 3 - голова, (5 6) - хвост
(cons 3 (list 5 6)) ;; (3 5 6)
;; (3 FROM 9 GIVES 6)
(list 3 'from 9 'gives (- 9 3))
;; (length '(1 foo 2 too)) -> 4
;; (car '(21 22 23)) -> 21
;; (+ 4 21) -> 25
(+ (length '(1 foo 2 too)) (car '(21 22 23))) ;; 25
;;(IS SHORT FOR ANS)
(cdr '(cons is short for ans))
;; Error undefined variable: ONE
(car (list one two))
;; (3 LIST 5 6)
(cons 3 '(list 5 6))
;; (list 'one 'two) -> (one two)
;; (car '(one two)) -> one
(car (list 'one 'two)) ;; one
```

Рисунок 0.5 — Задание 6.1

```
;; №6_2
;; Дана функция
;; (first list)    ==   (car list)
;; (second list)   ==   (car (cdr list))
;; Принимает список x
;; Возвращает список, содержащий второй и первый элемент
;; (Именно в таком порядке).
(defun mystery (x) (list (second x) (first x)))

;; Какие результаты вычислений следующий выражений?
(mystery '(one two))    ;; (TWO ONE)
;; Error The value FREE is not of type LIST
;; (cdr 'free) - выдаст ошибку.
(mystery 'free)
;; Error The value ONE is not of type LIST when binding LIST
;; (last 'one 'two) - выдаст ошибку.
(mystery (last 'one 'two))
;; Error invalid number of arguments: 2
;; Должен быть один аргумент.
(mystery 'one 'two)
```

Рисунок 0.6 — Задание 6.2

## Ответы на вопросы

### Базис лиспа.

**Базис** - минимальный набор средств для решения любой задачи.

Базис:

1) атомы и бинарные узлы;

2) atom, eq, cons, car, cdr, cond, quote, eval.

atom проверяет, является ли объект, переданный в качестве аргумента, атомом.

```
1 (atom 'a) ;; t
2 (atom '(a b c)) ;; nil
```

eq проверяет идентичность двух символов.

```
1 (eq 'a 'b) ;; nil
2 (eq 'a 'a) ;; t
```

cond - сокращение от англ. condition – условие. Не имеет фиксированного количества аргументов. Каждый аргумент - это список, голова которого рассматривается как условие, и если оно истинно, то результатом будет хвост рассматриваемого списка.

```
1 (cond ((eq 'A 'B) 'are_equal)
2       (T 'not_equal)) ;; NOT_EQUAL
```

eval - выполняет двойное вычисление своего аргумента.

```
1 (eval (cons (quote car) (quote ('(A B))))) => A
2          |---------(car '(A B))---------|
```

### Классификация функций.

а) Чистые математические функции - имеет фиксированное количество аргументов и один результат.

б) Специальны функции - произвольное количество аргументов.

в) Псевдофункции - создают эффект на экране.

г) Рекурсивные функции.

д) Функции с вариантными значениями, которые возвращают одно значение.

е) Функции высших порядков - используются для построения синтаксически управляемых программ.

**Список** - динамическая структура данных, которая может быть пустой или непустой. Если она не пустая, то состоит из двух элементов:

1. Головы - любая структура.

2. Хвоста - список.

Список представляет из себя заключенную в скобки последовательность из атомов, разделенных пробелами, или списков. Любой список является программой - его нужно вычислять.

### Как выполняются функции car и cdr

**car** - возвращает голову. **cdr** - возвращает хвост.

### Отличие list и cons

cons - имеет фиксированное количество аргументов (два). В случае, когда аргументами являются атомы создает точечную пару. В случает, когда первый аргумент атом а второй список, атом становится головой, а второй аргумент (список) становится хвостом.

```
1 (cons 'a 'b)           ;; (A . B)
2 (cons 'a '(a b c))     ;; (A A B C)
3 (cons '(a c) '(b d)) ;;((A C) B D)
4 (cons 'a 'v 'd)        ;; Error (invalid number of arguments: 3)
```

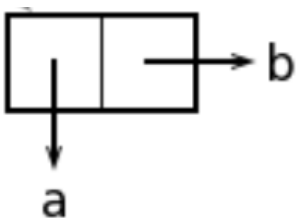


Рисунок 0.1 — Результат cons.

list - не имеет фиксированное количество аргументов. Создает список, у которого голова - это первый аргумент, хвост - все остальные аргументы.

```
1 (list 'a 'b)                  ;; (A B)
2 (list 'a 'b 'v '(c d) 'd)     ;; (A B V (C D) D)
```

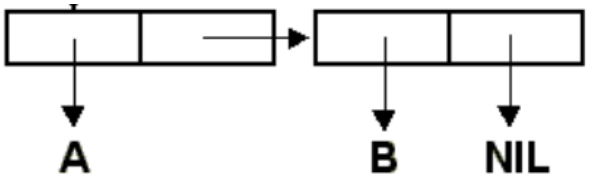


Рисунок 0.2 — Результат list.

**cons** - имеет фиксированное число аргументов и более экономный по памяти.