



**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»

Дисциплина: «Функциональное и логическое
программирование»

Лабораторная работа №3

Студент: Левушкин И. К.

Группа: ИУ7-62Б

Преподаватели: Толпинская Н. Б.,

Строганов Ю. В.

Москва, 2020 г.

1 Составить диаграмму вычисления следующих выражений:

(equal 3 (abs - 3))

(equal (+ 1 2) 3)

(equal (4 7) 21)*

(equal (2 3) (+ 7 2))*

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

```
(defun f1 (a b) (sqrt (+ (* a a) (* b b) ) ) )
```

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

```
(defun v (x y z) (* x y z))
```

4 Каковы результаты вычисления следующих выражений?

Выражение	Результат
(list 'a c)	error: variable C has no value
(cons 'a (b c))	error: undefined function B
(cons 'a '(b c))	(A B C)
(caddr (1 2 3 4 5))	error: 1 is not a function name
(cons'a'b'c)	error: too many arguments given to CONS
(list 'a (b c))	error: undefined function B
(list a '(b c))	error: variable A has no value
(list (+ 1 '(length '(1 2 3))))	error: +: (LENGTH '(1 2 3)) is not a number

5 Написать функцию `longer_than` от двух списков-аргументов, которая возвращает `T`, если первый аргумент имеет большую длину.

```
(defun longer_than (a b) (cond ((> (length a) (length b)) T)))
```

Пример:

- (longer_than '(a b c d) '(a b c)) = T
- (longer_than '(f ss) '(r w q)) = Nil
- (longer_than '(1 (2 (3 (4)))) '(1 (2 (3)))) = Nil

6 Каковы результаты вычисления следующих выражений?

Выражение	Результат
(cons 3 (list 5 6))	(3 5 6)
(list 3 'from 9 'lives (- 9 3))	(3 FROM 9 LIVES 6)
((+ (length for 2 too)) (car '(21 22 23)))	error: (+ (LENGTH FOR 2 TOO)) is not a function name
(cdr '(cons is short for ans))	(IS SHORT FOR ANS)
(car (list one two))	error: variable ONE has no value
(cons 3 '(list 5 6))	(3 LIST 5 6)
(car (list 'one 'two))	ONE

Ответы на вопросы

1. Базис - это минимально необходимый набор конструкций, с помощью которого можно реализовать задачу.
2. Варианты классификаций функции LISP
 - чистые(математические): принимают строго определенное число аргументов и возвращают одно значение
 - формы: могут принимать разное число аргументов, в зависимости от чего по-разному себя ведет
 - функционалы: принимают функциональные описания.
3. Классификация базовых функций Lisp
 - функции-селекторы (функции доступа к данным): car, cdr
 - функции-конструкторы: cons, list
 - функции-предикаты: atom, Null, lisp и т.д.
 - функции-сравнения: eq, eql, =, equal, equalp
4. Как работают car, cdr Функции car и cdr являются базисными функциями-селекторами. car принимает в качестве аргумента точечную пару или список. Функция возвращает голову списка. В случае точечной пары или непустого списка функция вернет первый элемент, в случае пустого списка - Nil. cdr также принимает в качестве аргумента точечную пару или список. Функция возвращает хвост списка, т. е. список, состоящий из всех элементов кроме первого. Если в списке меньше двух элементов, то функция возвращает Nil.
5. Отличия list и cons Cons - базис языка, на вход принимает ровно два аргумента и создает одну списковую ячейку (расставляет указатели). List - написана на базе cons, принимает любое количество аргументов и создает список. Пример: (cons 'a (cons 'b (cons 'c Nil))) = (a b c) (list 'a 'b 'c) = (a b c)