



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 №2  
по дисциплине «Функциональное и логическое  
программирование»

Тема Определение функций пользователя

Студент Зайцева А. А.

Группа ИУ7-62Б

Оценка (баллы) \_\_\_\_\_

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

Москва — 2022 г.

# Теоретические вопросы

## 1. Базис Lisp

Базис – это минимальный набор инструментов языка и структур данных, который позволяет решить любые задачи.

Базис Lisp :

- атомы и структуры (представляющиеся бинарными узлами);
- базовые (несколько) функций и функционалов: встроенные — примитивные функции (atom, eq, cons, car, cdr); специальные функции и функционалы (quote, cond, lambda, eval, apply, funcall).

Атомы:

- символы (идентификаторы) – синтаксически – набор литер (букв и цифр), начинающихся с буквы;
- специальные символы – T, Nil (используются для обозначения логических констант);
- самоопределимые атомы – натуральные числа, дробные числа, вещественные числа, строки – последовательность символов, заключенных в двойные апострофы (например, “abc”);

Более сложные данные – списки и точечные пары (структуры), которые строятся с помощью унифицированных структур – блоков памяти – бинарных узлов.

Определения:

Точечная пара ::= (<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>,

S-выражение ::= <атом> | <точечная пара>,

<хвост> ::= <список>.

Функцией называется правило, по которому каждому значению одного или нескольких аргументов ставится в соответствие конкретное значение результата.

Функционалом, или функцией высшего порядка называется функция, аргументом или результатом которой является другая функция.

## 2.Классификация функций

Один из вариантов классификации функций:

- чистые математические функции (имеют фиксированное количество аргументов, сначала выясняются все аргументы, а только потом к ним применяется функция);
- рекурсивные функции (основной способ выполнения повторных вычислений);
- специальные функции, или формы (могут принимать произвольное количество аргументов, или аргументы могут обрабатываться по-разному);
- псевдофункции (создают «эффект», например, вывод на экран);
- функции с вариантами значений, из которых выбирается одно;
- функции высших порядков, или функционалы – функции, аргументом или результатом которых является другая функция (используются для построения синтаксически управляемых программ);

## 3. Способы создания функций

# Практические задания

1. 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)

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

```
1 (defun hypotenuse (a b) (sqrt (+ (* a a) (* b b))))  
2 (hypotenuse 3 4) => 5.0
```

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

```
1 (defun p_volume (a b c) (* (* a b) c))  
2 (p_volume 2 3 4) => 24  
3 ;; or  
4 (defun p_volume (a b c) (* a b c))  
5 (p_volume 2 3 4) => 24
```

4. Каковы результаты вычисления следующих выражений? (объяснить возможную ошибку и варианты ее устранения)

1. (list 'a c) => The variable C is unbound.
2. (cons 'a (b c))
3. (cons 'a '(b c))
4. (caddy (1 2 3 4 5))
5. (cons 'a 'b 'c)

6. `(list 'a (b c))`
7. `(list a '(b c))`
8. `(list (+ 1 '(length '(1 2 3))))`