



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»

---

## Отчёт

### по лабораторной работе №3

Название «Работа интерпретатора Lisp»

---

Дисциплина «Функциональное и логическое программирование»

---

Студент ИУ7-65Б

---

(подпись, дата)

Клименко А.К.

---

(Фамилия И.О.)

Преподаватель

---

(подпись, дата)

Толпинская Н.Б.

---

(Фамилия И.О.)

Москва, 2022

## Содержание

Введение . . . . .	3
1 Практические задания . . . . .	4
1.1 Задание 1 . . . . .	4
1.2 Задание 2 . . . . .	4
1.3 Задание 3 . . . . .	5
1.4 Задание 4 . . . . .	5
1.5 Задание 5 . . . . .	5
1.6 Задание 6 . . . . .	5
1.7 Задание 7 . . . . .	6
1.8 Задание 8 . . . . .	6
1.9 Задание 9 . . . . .	7
Заключение . . . . .	8

## Введение

Программа на Lisp представляет собой вызов функции на верхнем уровне. Функции в Lisp делятся на *типичные* (математические) функции и *формы* — функции, которые особым образом обрабатывают свои аргументы, т. е. требуют специальной обработки. Кроме этого, функции в Lisp носят *частичный характер* т. е. по разному, иногда не корректно работают на множестве S-выражений.

Синтаксически программа оформляется в виде *S-выражения* (обычно – списка). S-выражение, попавшее на вход системы анализирует функция eval. S-выражение очень часто может быть *структурированным*.

**Цель работы:** приобрести навыки работы в системе Common Lisp.

Для достижения поставленной цели необходимо выполнить следующие задачи:

- изучить работу форм;
- изучить правила работы функций cond, if, and/or на различных списках-аргументах;
- изучить особенности работы форм в Lisp;
- проанализировать эффективность работы разных реализаций.

# 1 Практические задания

## 1.1 Задание 1

Написать функцию, которая принимает целое число и возвращает первое четное число, не меньшее аргумента.

```
1 ; вариант 1
2 (defun first-even-ge (num)
3   (if (= (mod num 2) 0)
4       num
5       (+ num 1)
6   )
7 )
8
9 ; вариант 2
10 (defun first-even-ge (num)
11   (+ num (mod num 2))
12 )
```

## 1.2 Задание 2

Написать функцию, которая принимает число и возвращает число того же знака, но с модулем на 1 больше модуля аргумента.

```
1 ; вариант 1
2 (defun plus-mod (num)
3   (if (< num 0)
4       (- num 1)
5       (+ num 1)
6   )
7 )
8
9 ; вариант 2
10 (defun plus-mod (num)
11   ((if (< num 0) - +) num 1)
12 )
```

### 1.3 Задание 3

Написать функцию, которая принимает два числа и возвращает список из этих чисел, расположенный по возрастанию.

```
1 (defun sorted-2list (a b)
2   (if (< a b)
3       (list a b)
4       (list b a))
5 )
6 )
```

### 1.4 Задание 4

Написать функцию, которая принимает три числа и возвращает Т только тогда, когда первое число расположено между вторым и третьим.

```
1 (defun betweenp (a b c)
2   (or (and (< b a) (< a c))
3       (and (< c a) (< a b)))
4 )
5 )
```

### 1.5 Задание 5

Каков результат вычисления следующих выражений?

Выражение	Результат
(and 'fee 'fie 'foe)	FOE
(or nil 'fie 'foe)	FIE
(and (equal 'abc 'abc) 'yes)	YES
(or 'fee 'fie 'foe)	FEE
(and nil 'fie 'foe)	NIL
(or (equal 'abc 'abc) 'yes)	T

### 1.6 Задание 6

Написать предикат, который принимает два числа-аргумента и возвращает Т, если первое число не меньше второго.

```
1 (defun gr-eq (a b)
2   (>= a b)
3 )
```

## 1.7 Задание 7

Какой из следующих двух вариантов предиката ошибочен и почему?

- 1) (defun pred1 (x) (and (numberp x) (plusp x)))
- 2) (defun pred2 (x) (and (plusp x) (numberp x)))

Второй вариант ошибочен, так как при вычислении формы `and` первым будет вычисляться предикат `plusp` и в случаях, когда аргумент `x` не будет являться числом, возникнет ошибка.

## 1.8 Задание 8

Решить задачу 4, используя для ее решения конструкции IF, COND, AND/OR.

```
1 ; IF
2 (defun betweenp (a b c)
3   (if (< b c)
4     (if (< b a) (< a c))
5     (if (< c a) (< a b))
6   )
7 )
8
9 ; COND
10 (defun betweenp (a b c)
11   (cond
12     ((< b a) (< a c))
13     ((< c a) (< a b))
14   )
15 )
16
17 ; AND/OR
18 (defun betweenp (a b c)
19   (or (and (< b a) (< a c))
20       (and (< c a) (< a b))
21   )
22 )
```

## 1.9 Задание 9

Переписать функцию `how-alike`, приведенную в лекции и использующую `COND`, используя только конструкции `IF`, `AND/OR`.

## **Заключение**

В ходе работы была изучена работа форм в языке Lisp. Изучены правила работы функций `cond`, `if`, `and/or` на различных списках-аргументах. Была проанализирована эффективность работы разных реализаций функций.