



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, искусственный интеллект и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

**Лабораторная работа № 3**  
**по дисциплине «Функциональное и**  
**логическое программирование»**

Тема Работа интерпретатора Lisp

Студент Царев А.А.

Группа ИУ7-63Б

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

Москва, 2023

# ВВЕДЕНИЕ

Целью данной работы является приобретение навыков работы в системе Common Lisp.

Задача работы: изучить работу форм — функций, которые особым образом обрабатывают свои аргументы и особенности их работы в Lisp.

# 1 Теоретические вопросы

## 1.1 Базис Lisp

Базис — минимальный набор конструкций языка, на основе которого могут быть построены вычисления.

Базис языка составляют:

- атомы;
- структуры;
- базовые функции;
- базовые функционалы.

## 1.2 Классификация функций

Функции в Lisp делятся на три класса:

- базисные,
- ядра,
- пользовательские.

Также, функции Lisp делятся на чистые (обычные, математические) и формы — функции, которые особым образом обрабатывают свои аргументы.

Базисные функции, в свою очередь, классифицируются следующим образом:

- селекторы (car, cdr и т.д.),
- конструкторы (cons),
- предикаты (atom),
- сравнения (eq).

## 1.3 Способы создания функций

В Lisp можно определять безымянные функции и функции с именем.

Безымянная функция:

```
(lambda (x1 x2 ... xn) f)
```

Функция с именем:

```
(defun <name> (x1 x2 ... xn) f)
```

Вызов:

```
(<name> a1 a2 ... an)
```

```
2 ((lambda (x1 x2 ... xn) f) a1 a2 ... an)
```

## 1.4 Работа функций `cond`, `if`, `and`, `or`

Вычисляет выражения `test` до тех пор, пока одно из них не окажется истинным. Если с ним не связаны какие-либо выражения, возвращается значение его самого, в противном случае связанные с ним выражения вычисляются по очереди и возвращается значение последнего. Если ни одно тестовое выражение не оказалось истинным, возвращается `Nil`.

```
1 (cond (test1 value1) (test2 value2) ... (testN valueN))
```

Функция `if` является сокращением `cond`. Если `test` равен `T`, то возвращается значение выражения `valT`. В ином случае возвращается значение выражения `valNil`, если оно передано функции `if` в качестве аргумента, иначе возвращается `Nil`.

```
1 (if test valT [valNil])
```

Функция `and` вычисляет выражения друг за другом до тех пор, пока одно из них не будет ложным. В таком случае возвращается `Nil`, в противном случае — значение последнего аргумента. Если выражения не заданы, возвращается `T`.

```
1 (and arg1 arg2 ... argN)
```

Функция `or` вычисляет выражения друг за другом до тех пор, пока одно из них не будет истинным. В таком случае возвращается само значение, в противном случае — `Nil`. Если выражения не заданы, возвращается `Nil`.

```
1 (or arg1 arg2 ... argN)
```

## 2 Практические задания

### 2.1 Задание 1. Написать функцию, которая принимает целое число и возвращает первое четное число, не меньшее аргумента

```
1 (defun min_even (num)
2   (if (evenp num)
3       num
4       (+ num 1)))
```

### 2.2 Задание 2. Написать функцию, которая принимает число и возвращает число того же знака, но с модулем на 1 больше модуля аргумента

```
1 (defun inc_abs (num)
2   (if (minusp num)
3       (- num 1)
4       (+ num 1)))
```

### 2.3 Задание 3. Написать функцию, которая принимает два числа и возвращает список из этих чисел, расположенный по возрастанию

```
1 (defun init_two_num_list (a b)
2   (if (> a b)
3       (cons b
4             (cons a Nil))
5       (cons a
6             (cons b Nil))))
```

## 2.4 Задание 4. Написать функцию, которая принимает три числа и возвращает Т только тогда, когда первое число расположено между вторым и третьим

```
1 (defun fst_is_med (num1 num2 num3)
2   (or (and (> num1 num2)
3           (< num1 num3))
4       (and (> num1 num3)
5           (< num1 num2))))
```

## 2.5 Задание 5. Каков результат вычисления следующих выражений?

Таблица 2.1 — Результаты вычисления 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

## 2.6 Задание 6. Написать предикат, который принимает два числа-аргумента и возвращает Т, если первое число не меньше второго

```
1 (defun ge (a b)
2   (eq (< a b)
3       Nil))
```

## 2.7 Задание 7. Какой из следующих двух вариантов предиката ошибочен и почему?

```
1 (defun pred1 (x)
2   (and (numberp x) (plusp x)))
```

или

```
1 (defun pred2 (x)
2   (and (plusp x) (numberp x)))
```

Ошибочным является предикат pred2, поскольку выражение (plusp x) будет вычислено раньше, чем выражение (numberp x). Если предикату pred2 передать в качестве аргумента символьный атом, то выражение (plusp x) не может быть вычислено, поскольку функция plusp может обрабатывать только вещественные числа.

## 2.8 Задание 8. Решить задачу 4, используя для ее решения конструкции: только IF, только COND, только AND/OR

```
1 (defun fst_is_med (num1 num2 num3)
2   (if (> num1 num2)
3       (< num1 num3)
4       (if (> num1 num3)
5           (< num1 num2)))))
```

```
1 (defun fst_is_med (num1 num2 num3)
2   (cond ((> num1 num2)
3         (cond ((< num1 num3))))
4         ((> num1 num3)
5         (cond ((< num1 num2))))))
```

```
1 (defun fst_is_med (num1 num2 num3)
2   (or (and (> num1 num2)
3           (< num1 num3))
4       (and (> num1 num3)
5           (< num1 num2))))
```

## **ЗАКЛЮЧЕНИЕ**