



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 по курсу «Функциональное и логическое программирование»

Тема Работа интерпретатора Lisp

Студент Богаченко А. Е.

Группа ИУ7-66Б

Оценка (баллы) \_\_\_\_\_

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

## Задание 1

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

Листинг 1 – Задание 1

```
1 (defun make-first-even (num)
2   (if (evenp num)
3       num
4       (+ num 1)))
```

## Задание 2

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

Листинг 2 – Задание 2

```
1 (defun make-abs-greater-num (num)
2   (if (> num 0)
3       (+ num 1)
4       (- num 1)))
```

## Задание 3

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

Листинг 3 – Задание 3

```
1 (defun make-asc-pair (a b)
2   (list (min a b) (max a b)))
```

## Задание 4

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

Листинг 4 – Задание 4

```
1 (defun is-between (a b c)
2   (if (and (> a b) (< a c))
3       T
4       Nil))
```

## Задание 5

Каков результат вычисления следующих выражений?

Листинг 5 – Задание 5

```
1 (and 'fee 'fie 'foe) ; FOE
2 (or 'fee 'fie 'foe) ; FEE
3 (and (equal 'abc 'abc) 'yes) ; YES
4 (or nil 'fie 'foe) ; FIE
5 (and nil 'fie 'foe) ; NIL
6 (or (equal 'abc 'abc) 'yes) ; T
```

## Задание 6

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

Листинг 6 – Задание 6

```
1 (defun gep (a b)
2   (if (>= a b)
3       T
4       Nil))
```

## Задание 7

Какой из следующих двух вариантов предиката ошибочен и почему?

Листинг 7 – mystery

```
1 (defun pred1 (x)
2   (and (numberp x) (plusp x))) ; Good
3
4 (defun pred2 (x)
5   (and (plusp x) (numberp x))) ; Bad
```

Второй предикат ошибочен, потому что в случае, если на вход функции будет подан аргумент, не являющийся числом, к нему будет применена функция `plusp`, определяющая, является ли число положительным. Она определена только для чисел, поэтому в случае нечислового аргумента интерпретатор выдаст ошибку.

## Задание 8

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

Листинг 8 – Задание 8

```
1 ;; IF variant
2 (defun is-between-if (a b c)
3   (if (and (> a b) (< a c))
4       T
5       Nil))
6
7 ;; COND variant
8 (defun is-between-cond (a b c)
9   (cond
10    ((and (> a b) (< a c)) T)
11    (T Nil)))
12
13 ;; AND/OR variant
14 (defun is-between-andor (a b c)
15   (and (> a b) (< a c)))
```

## Задание 9

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

### Листинг 9 – Задание 9

```
1 ;; COND variant
2 (defun how-alike-cond (x y)
3   (cond
4     ((or (= x y) (equal x y)) 'the_same)
5     ((and (oddp x) (oddp y)) 'both_odd)
6     ((and (evenp x) (evenp y)) 'both_even)
7     (T 'difference)))
8
9 ;; IF variant
10 (defun how-alike-if (x y)
11   (if (if (= x y)
12         (equal x y))
13       'the_same
14       (if (if (oddp x)
15               (oddp y))
16           'both_odd
17           (if (if (evenp x)
18                   (evenp y))
19               'both_even
20               'difference)))))
21
22 ;; AND/OR variant
23 (defun how-alike-andor (x y)
24   (or (and (or (= x y) (equal x y)) 'the_same)
25       (and (oddp x) (oddp y) 'both_odd)
26       (and (evenp x) (evenp y) 'both_even)
27       'difference))
```

# Контрольные вопросы

## 1. Базис языка Lisp

Базис языка представлен:

- структурами, атомами;
- функциями:  
atom, eq, cons, car, cdr,  
cond, quote, lambda, eval, label.

## 2. Классификация функций языка Lisp

Функции в языке Lisp:

- чистые (с фиксированным количеством аргументов) – математические функции;
- рекурсивные функции;
- специальные функции – формы (принимают произвольное количество аргументов или по разному обрабатывают аргументы);
- псевдофункции (создающие «эффект» - отображающие на экране процесс обработки данных и т. п.);
- функции с вариативными значениями, выбирающие одно значение;
- функции высших порядков – функционалы (используются для построения синтаксически управляемых программ).

По назначению функции разделяются следующим образом:

1. конструкторы — создают значение (`cons`, например);
2. селекторы — получают доступ по адресу (`car`, `cdr`);
3. предикаты — возвращают `Nil`, `T`.

### 3. Способы создания функций

С помощью макро определения `defun` или с использованием Лямбда-нотации (функция без имени).

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

##### Функция `and`

Синтаксис: `(and expression-1 expression-2 ... expression-n)`

Функция возвращает первое `expression`, результат вычисления которого `= Nil`. Если все не `Nil`, то возвращается результат вычисления последнего выражения.

Примеры:

```
(and 1 Nil 2) -> Nil
```

```
(and 1 2 3) -> 3
```

##### Функция `or`

Синтаксис: `(or expression-1 expression-2 ... expression-n)`

Функция возвращает первое `expression`, результат вычисления которого не `Nil`. Если все `Nil`, то возвращается `Nil`.

Примеры:

```
(or Nil Nil 2) -> 2
```

```
(or 1 2 3) -> 1
```

##### Функция `if`

Синтаксис: `(if condition t-expression f-expression)`

Если вычисленный предикат не `Nil`, то выполняется `t-expression`, иначе - `f-expression`.

Примеры:

```
(if Nil 2 3) -> 3
```

```
(if 0 2 3) -> 2
```

## Функция cond

Синтаксис:

```
(cond  
  (condition-1 expression-1)  
  (condition-2 expression-2)  
  ...  
  (condition-n expression-n))
```

По порядку вычисляются и проверяются на равенство с `Nil` предикаты. Для первого предиката, который не равен `Nil`, вычисляется находящееся с ним в списке выражение и возвращается его значение. Если все предикаты вернут `Nil`, то и `cond` вернет `Nil`.

Примеры:

```
(cond (Nil 1) (2 3)) -> 3
```

```
(cond (Nil 1) (Nil 2)) -> Nil
```