



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 №5  
по курсу «Функциональное и логическое программирование»  
на тему: «Использование функционалов»

Студент ИУ7-61Б  
(Группа)

\_\_\_\_\_  
(Подпись, дата)

Постнов С. А.  
(Фамилия И. О.)

Преподаватель

\_\_\_\_\_  
(Подпись, дата)

Толпинская Н. Б.  
(Фамилия И. О.)

Преподаватель

\_\_\_\_\_  
(Подпись, дата)

Строганов Ю. В.  
(Фамилия И. О.)

2024 г.

# СОДЕРЖАНИЕ

<b>1</b>	<b>Практические задания</b>	<b>3</b>
1.1	Задание 1 . . . . .	3
1.2	Задание 2 . . . . .	3
1.3	Задание 3 . . . . .	3
1.4	Задание 4 . . . . .	4
1.5	Задание 5 . . . . .	4
1.6	Задание 6 . . . . .	4
1.7	Задание 7 . . . . .	5
1.8	Задание 8 . . . . .	5
1.9	Задание 9 . . . . .	5

# 1 Практические задания

## 1.1 Задание 1

В листинге 1.1 представлена функция, которая уменьшает на 10 все числа из списка-аргумента, проходя по верхнему уровню.

Листинг 1.1 – Функция, которая уменьшает на 10 все числа из списка-аргумента

```
1 (defun minus_10 (lst)
2     (mapcar #'(lambda (x)
3                 (cond
4                     ((numberp x)
5                      (- x 10))
6                     (t
7                      x)))) lst))
```

## 1.2 Задание 2

В листинге 1.2 представлена функция, которая получает как аргумент список чисел, а возвращает список квадратов этих чисел в том же порядке.

Листинг 1.2 – Функция, которая возвращает список квадратов чисел

```
1 (defun quad (lst)
2     (mapcar #' * lst lst))
```

## 1.3 Задание 3

В листинге 1.3 представлена функция, которая умножает на заданное число-аргумент все числа из заданного списка-аргумента, когда все элементы — числа или любые объекты.

Листинг 1.3 – Функция, которая умножает на заданное число-аргумент все числа из заданного списка-аргумента

```
1 (defun multi (lst a)
2     (mapcar #'(lambda (x)
3                 (cond
4                     ((numberp x)
5                      (* x a))
6                     (t
7                      x)))) lst))
```

## 1.4 Задание 4

В листинге 1.4 представлена функция, которая по своему списку-аргументу `lst` определяет, является ли он палиндромом (то есть равны ли `lst` и `(reverse lst)`), для одноуровневого смешанного списка.

Листинг 1.4 – Функция, которая по своему списку-аргументу `lst` определяет, является ли он палиндромом

```
1 | (defun is_palindrome (lst)
2 |     (equal (reverse lst) lst))
```

## 1.5 Задание 5

В листинге 1.5 представлен предикат `set-equal`, который возвращает `t`, если два его множества-аргумента содержат одни и те же элементы, порядок которых не имеет значения.

Листинг 1.5 – Предикат `set-equal`, который возвращает `t`, если два его множества-аргумента содержат одни и те же элементы

```
1 | (defun set-equal (lst1 lst2)
2 |     (and (= (length lst1) (length lst2))
3 |         (= (length lst1) (length (intersection lst1 lst2)))))
```

## 1.6 Задание 6

В листинге 1.6 представлена функция `select-between`, которая из списка-аргумента, содержащего только числа, выбирает только те, которые расположены между двумя указанными числами — границами - аргумента и возвращает их в виде списка (упорядоченного по возрастанию).

Листинг 1.6 – Функция `select-between`

```
1 | (defun select-between (lst a b)
2 |     (cond
3 |         ((null lst) Nil)
4 |         (t (reduce #'(lambda (x y)
5 |                         (cond ((and (> y a) (< y b)) (cons y x))
6 |                             (t x))) lst :initial-value nil))))
```

## 1.7 Задание 7

В листинге 1.7 представлена функция, вычисляющая декартово произведение двух своих списков-аргументов.

Листинг 1.7 – Функция, вычисляющая декартово произведение

```
1 (defun cartesian (lstX lstY)
2   (mapcar #'(lambda (x)
3     (mapcar #'(lambda (y)
4       (cons x
5         (cons y
6           Nil)))) lstX)) lstY))
```

## 1.8 Задание 8

Описание поведения функции **reduce**:

- 1) (**reduce** #'**+** '()) возвращает 0 в силу начального значения по умолчанию, равного 0;
- 2) (**reduce** #'**\*** '()) возвращает 1 в силу начального значения по умолчанию, равного 1.

## 1.9 Задание 9

В листинге 1.8 представлена функция, которая вычисляет сумму длин всех элементов-списков списка **list-of-list**.

Листинг 1.8 – Функция, которая вычисляет сумму длин всех элементов

```
1 (defun lst_sum (lst)
2   (reduce #'(lambda (sum l)
3     (+ sum (length l))) lst :initial-value 0))
4
5 (setf list-of-list '((1 2) (3 4)))
6
7 (lst_sum list-of-list) -> 4
```