



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»

---

## Отчёт по лабораторной работе № 5 по курсу «Функциональное и логическое программирование»

Тема Использование функционалов

Студент Калашков П. А.

Группа ИУ7-66Б

Оценка (баллы) \_\_\_\_\_

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

# 1 Практические задания

Задание 1. Напишите функцию, которая уменьшает на 10 все числа из списка-аргумента этой функции, проходя по верхнему уровню списковых ячеек. ( \* Список смешанный структурированный)

Листинг 1.1 – Задание 1, mapcar

```
1 (defun minus-ten (lst) (  
2   mapcar #'(lambda (el)  
3     (cond ((numberp el) (- el 10))  
4       (T el))  
5   ) lst  
6 )  
7 )
```

Листинг 1.2 – Задание 1, mapcan

```
1 (defun minus-ten (lst) (  
2   mapcan #'(lambda (el)  
3     (cons  
4       (cond ((numberp el) (- el 10))  
5         (T el))  
6     Nil)  
7   )  
8   lst)  
9 )
```

**Задание 2.** Написать функцию которая получает как аргумент список чисел, а возвращает список квадратов этих чисел в том же порядке.

Листинг 1.3 – Задание 2, mapcar

```
1 (defun sqr-lst (lst) (  
2   mapcar #'(lambda (el)  
3     (* el el)) lst  
4   )  
5 )
```

Листинг 1.4 – Задание 2, mapcan

```
1 (defun sqr-lst (lst) (  
2   mapcan #'(lambda (el)  
3     (cons (* el el) Nil))  
4   lst)  
5 )
```

**Задание 3.** Напишите функцию, которая умножает на заданное число-аргумент все числа из заданного списка-аргумента, когда

- а) все элементы списка — числа,
- б) элементы списка — любые объекты.

Листинг 1.5 – Задание 3, а, mapcar, mapcon

```
1 (defun mul-lst (lst num) (  
2   mapcar #'(lambda (el)  
3     (* el num)) lst  
4   )  
5 )  
6  
7 (defun mul-lst (lst num) (  
8   mapcan #'(lambda (el)  
9     (cons (* el num) Nil))
```

|    |   |      |  |
|----|---|------|--|
| 10 |   | 1st) |  |
| 11 | ) |      |  |

## Листинг 1.6 – Задание 3, b, mapcar, mapcon

```

1 (defun mul-lst (lst num) (
2   mapcar #'(lambda (el)
3     (cond ((numberp el) (* el num))
4       (T el))
5   ) lst
6 )
7 )
8
9 (defun mul-lst (lst num) (
10  mapcon #'(lambda (el)
11    (cons
12      (cond ((numberp el) (* el num))
13        (T el))
14      Nil)
15    )
16  lst)
17 )

```

**Задание 4.** Написать функцию, которая по своему списку-аргументу `lst` определяет является ли он палиндромом (то есть равны ли `lst` и `(reverse lst)`), для одноуровневого смешанного списка.

## Листинг 1.7 – Задание 4

```

1 (defun is-palindrom (lst)
2   (cond ((eql
3     (find-if #'evenp
4       (mapcar #'(lambda (elem reverse-elem)
5         (cond ((eql elem reverse-elem) 1)
6           (T 0))
7       ) lst (reverse lst))
8     ) Nil) T)
9     (T Nil))
10 )

```

**Задание 5.** Используя функционалы, написать предикат `set-equal`, который возвращает `t`, если два его множества-аргумента (одноуровневые списки) содержат одни и те же элементы, порядок которых не имеет значения.

Листинг 1.8 – Задание 5

```
1 (defun set-equal (set1 set2) (  
2   and (eq (length set1) (length set2))  
3       (every #'(lambda (el)  
4         (member el set2 :test #'equal)  
5       ) set1)  
6       (every #'(lambda (el)  
7         (member el set1 :test #'equal)  
8       ) set2)  
9 ))
```

## Задание 6.

Напишите функцию, `select-between`, которая из списка-аргумента, содержащего только числа, выбирает только те, которые расположены между двумя указанными числами - границами-аргументами и возвращает их в виде списка (упорядоченного по возрастанию (+ 2 балла))

Листинг 1.9 – Задание 6

```
1 (defun select-between (lst left right) (  
2   cond ((< right left) Nil)  
3       (T (  
4         sort (reduce #'(lambda (result el)  
5           (cond ((and (> el left) (> right el))  
6             (cons el result))  
7           (T result)  
8         )  
9         ) lst :initial-value ()) #'<  
10  ))  
11 ))
```

## Задание 7. Написать функцию, вычисляющую декартово произведение двух своих списков-аргументов.

Листинг 1.10 – Задание 7

```
1 (defun decart (lst1 lst2) (  
2   mapcan #'(lambda (x)  
3     (mapcar #'(lambda (y)  
4       (list x y)) lst2)) lst1  
5 ))
```

## Задание 8. Почему так реализовано reduce, в чем причина?

Листинг 1.11 – Задание 8

```
1 (reduce #'+ ()) ; -> 0  
2 (reduce #'* ()) ; -> 1
```

Если список пуст, а начальное значение не задано, то вызывается функция без аргументов, а reduce возвращает то, что вернёт функция. Функция сложения без аргументов возвращает 0, а функция умножения возвращает 1.

## Задание 9.

Пусть list-of-list список, состоящий из списков. Написать функцию, которая вычисляет сумму длин всех элементов list-of-list (количество атомов), т.е. например для аргумента ((1 2) (3 4)) -> 4.

## Листинг 1.12 – Задание 9

```
1 (defun len-list-of-lists (lst) (  
2   apply #'+ (  
3     mapcar #'(lambda (el)  
4       (cond ((listp el) (len-list-of-lists el))  
5         (T 1)  
6       )  
7     ) lst  
8   )  
9 ))
```