



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 №6  
по дисциплине «Функциональное и логическое  
программирование»**

Тема Использование функционалов

Студент Криков А.В.

Группа ИУ7-63Б

Оценка (баллы) \_\_\_\_\_

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

# Практические задания

## Задание 1

Напишите функцию, которая уменьшает на 10 все числа из списка-аргумента этой функции

```
1 (defun minus-10 (lst)
2   (mapcar (lambda (e) (if (numberp e) (- e 10) e)) lst))
```

## Задание 2

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

1. Все элементы списка — числа,
2. Элементы списка — любые объекты.

```
1 (defun mult-all (lst n)
2   (mapcar (lambda (e) (if (numberp e) (* e n) e)) lst))
3
4
5 (defun mult-all-deep (lst n)
6   (mapcar (lambda (e) (cond ((numberp e) (* e n))
7                             ((listp e) (mult-all-deep e n))
8                             (T e)))
9     lst))
```

## Задание 3

Написать функцию, которая по своему списку-аргументу lst определяет является ли он палиндромом (то есть равны ли lst и (reverse lst)).

```
1 (defun my-reverse (lst)
2   (reduce (lambda (res e) (cons e res)) lst :initial-value ()))
3
4 (defun palindromp (lst)
5   (equal lst (my-reverse lst)))
```

## Задание 4

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

```
1 (defun in-set (el lst)
2   (reduce (lambda (a b) (or a b))
3     (mapcar (lambda (e) (equal el e)) lst)))
4
5 (defun my-subsetp (l1 l2)
6   (reduce (lambda (a b) (and a b))
7     (mapcar (lambda (e) (in-set e l2)) l1)))
8
9 (defun set-equal (l1 l2)
10  (and (my-subsetp l1 l2) (my-subsetp l2 l1)))
```

## Задание 5

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

```
1 (defun sqr-lst (lst)
2   (mapcar (lambda (e) (* e e)) lst))
```

## Задание 6

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

```
1 (defun select-between (lst n1 n2)
2   (mapcan (lambda (e) (and (< n1 e n2) (list e)))
3     lst))
4
5 (defun select-between-sorted (lst n1 n2)
6   (sort (select-between lst n1 n2) #'<))
```

## Задание 7

Написать функцию, вычисляющую декартово произведение двух своих списков аргументов.

```
1 (defun decart-mult (l1 l2)
2   (mapcan (lambda (e1)
3     (mapcar (lambda (e2)
```

```

4      (list e1 e2))
5      l2))
6  l1))

```

## Задание 8

Почему так реализовано `reduce`, в чем причина? `(reduce #'+ ()) -> 0`

## Решение

Функция `+` — функционал, который при 0 количестве аргументов возвращает значение 0. Если подать на вход `reduce` функцию, которая не может обработать 0 аргументов, то вызов `reduce` с пустым списком в качестве второго аргумента вернет ошибку (`invalid number of arguments: 0`). При этом, если подано более одного аргумента, то `reduce` выполняет действия:

1. Сохраняет первый элемент списка в область памяти;
2. Для всех остальных элементов списка выполняет переданную в качестве первого аргумента функцию, подавая на вход 2 аргумента и сохраняя результат в асс.

## Задание 9

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

```

1 (defun sum-of-len (lst)
2   (reduce (lambda (res e) (+ res (length e)))
3     lst
4     :initial-value 0))

```