



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 функционало»

Студент ИУ7-61Б
(Группа)

(Подпись, дата)

Д. В. Шубенина
(И. О. Фамилия)

Преподаватель

(Подпись, дата)

Н. Б. Толпинская
(И. О. Фамилия)

Преподаватель

(Подпись, дата)

Ю. В. Строганов
(И. О. Фамилия)

2024 г.

СОДЕРЖАНИЕ

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

1 Практические задания

1.1 Задание 1

Напишите функцию, которая уменьшает на 10 все числа из списка-аргумента этой функции, проходя по верхнему уровню списковых ячеек.

```
1 | (defun f (lst)
2 |   (mapcar #'(lambda (x) (cond ((numberp x) (- x 10))
3 |                               (t x))) lst))
```

1.2 Задание 2

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

```
1 | (defun square (lst)
2 |   (mapcar #'(lambda (x) (* x x)) lst))
```

1.3 Задание 3

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

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

```
1 | (defun times-all-num (num lst)
2 |   (mapcar #'(lambda (x) (* num x)) lst))
3 |
4 | (defun times-mixed (num lst)
5 |   (mapcar #'(lambda (x) (cond ((numberp x) (* num x))
6 |                               (t x))) lst))
```

1.4 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.

```
1 | (defun is-palindrome (lst)
2 |   (equal lst (reverse lst)))
```

1.5 Задание 5

Используя функционалы, 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 set-equal (s1 s2)
2   (every #'(lambda (x) x)
3     (mapcar #'(lambda (x)
4       (some #'(lambda (y) y)
5         (mapcar #'(lambda (y)
6           (equal x y))
7             s1)))
8       s2)))
```

1.6 Задание 6

Написать 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).

```
1 (defun select-between (lst a b)
2   (cond ((null (car lst)) nil)
3     (T (reduce #'(lambda (x y)
4       (if (< a y b)
5         (cons y x)
6         x))
7       lst :initial-value nil))))
```

1.7 Задание 7

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

```
1 (defun decart (lx ly)
2   (mapcar #'(lambda (x)
3     (mapcar #'(lambda (y)
4       (cons x (cons y nil)))
5       ly))
6     lx))
```

1.8 Задание 8

Почему так реализовано `reduce`, в чем причина?

```
(reduce #'+ ()) -> 0
```

```
(reduce #'* ()) -> 1
```

При передаче пустого списка в функцию `reduce` она возвращает значение `initial-value`, которое для сложения равно 0, а для умножения — 1.

1.9 Задание 9

Пусть `list-of-list` список, состоящий из списков. Написать функцию, которая вычисляет сумму длин всех элементов.

```
1 | (defun s (lst)
2 |   (cond
3 |     ((null lst) 0)
4 |     ((atom lst) 1)
5 |     (t (+ (s (car lst))
6 |           (s (cdr lst))))))
```