



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 №6 по курсу «Функциональное и логическое программирование»

Тема Использование функционалов

Студент Богаченко А. Е.

Группа ИУ7-65Б

Оценка (баллы) _____

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

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

Листинг 1 – Задание 1

```
1 (defun subtract-10 (lst)
2   (mapcar #'(lambda (x) (- x 10)) lst))
```

2. Напишите функцию, которая умножает на заданное число-аргумент все числа из заданного списка-аргумента, когда все элементы числа и любые объекты

Листинг 2 – Задание 2

```
1 (defun mult-all (multiplier lst)
2   (mapcar #'(lambda (x) (* x multiplier)) lst))
3
4 (defun mult-all-nested (multiplier lst)
5   (mapcar #'(lambda (x)
6               (if (listp x)
7                   (mult-all-nested multiplier x)
8                   (* x multiplier)))
9         lst))
```

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

Листинг 3 – Задание 3

```
1 (defun palindromep (lst)
2   (equal lst (reverse lst)))
```

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

Листинг 4 – Задание 4

```
1 (defun set-equal (lst1 lst2)
2   (and (subsetp lst2 lst1) (subsetp lst1 lst2)))
```

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

Листинг 5 – Задание 5

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

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)

Листинг 6 – Задание 6

```
1 (defun select-between (from to lst)
2   (sort (mapcan #'(lambda (x)
3                     (and (>= x from) (<= x to) (list x)))
4                     lst)
5         #'<))
```

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

Листинг 7 – Задание 7

```
1 (defun decart-mult (a b)
2   (mapcan #'(lambda (outer)
3               (mapcar #'(lambda (inner)
4                           (count outer inner))
5                           b))
6           a))
```

8. Почему так реализовано `reduce`, в чем причина?

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

Поведение в данном примере обусловлено работой функции `+`. Эта функция — функционал, который при 0 количестве аргументов возвращает значение 0. Если подать на вход `reduce` функцию, которая не может обрабатывать 0 аргументов (например, математическая функция `cons`), то вызов `reduce` с пустым списком в качестве второго аргумента вернет ошибку (`invalid number of arguments: 0`).

9. Написать функцию, которая вычисляет сумму длин всех элементов вложенного списка списков

Листинг 8 – Задание 9

```
1 (defun length! (lol)
2   (reduce #'(lambda (acc x)
3             (+ acc (length x)))
4   (cons 0 lol)))
```