



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

Название «Использование функционалов»

Дисциплина «Функциональное и логическое программирование»

Студент ИУ7-65Б

(подпись, дата)

Бугаенко А.П.

(Фамилия И.О.)

Преподаватель

(подпись, дата)

Толпинская Н.Б.

(Фамилия И.О.)

Москва, 2022

1 Цели и задачи работы

Цель работы — приобрести навыки использования функционалов. Задачи работы — изучить работу и методы использования применяющих и отображающих функционалов: `apply`, `funcall`, `mapcar`, `maplist`.

2 Практические задания

Используя функционалы решите описанные ниже задачи.

2.1 Задание 1

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

```
1 (defun add_10 (x)
2   (+ x 10)
3 )
4
5 (defun add_10_lst (x)
6   (mapcar 'add_10 x)
7 )
```

2.2 Задание 2

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.

Функция, если все элементы - числа:

```
1 (defun func_mult_x (n)
2   #'(lambda (x) ( * x n))
3 )
4
5 (defun mult_numbers (lst x)
6   (mapcar (func_mult_x x) lst)
7 )
```

Функция, если элементы - любые объекты:

```
1 (defun func_mult_check_x (n)
2   #'(lambda (x)
3     (if (numberp x)
4         ( * x n)
5         x
6       )
7   )
8 )
9
```

```

10 (defun mult_check_numbers (lst x)
11   (mapcar (func_mult_check_x x) lst)
12 )

```

2.3 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 check_palindrom (lst)
2   (reduce #'(lambda (x y) (and x y)) (mapcar #'(lambda (x y) (eql x y)) lst (reverse
3     lst))))
4 \end{lstlisting}
5
6 \section{Задание 4}
7 Написать предикат set-equal, который возвращает t, если два его множества-аргумента с
   одержат одни и те же элементы, порядок которых не имеет значения.
8
9 \begin{lstlisting}
10 (defun set-equal (set_1 set_2)
11   (eval '(and ,@(mapcar #'(lambda (x)
12     (eval '(or ,@x))
13     )
14     (mapcar #'(lambda (x)
15       (mapcar #'(lambda (y)
16         (eql x y)
17         )
18         set_2
19       )
20     )
21     set_1
22   )
23   ))
24 )
25 )

```

2.4 Задание 5

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

```

1 (defun square_lst (lst)
2   (mapcar (lambda (x) x x) lst))

```

2.5 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.

```
1 (defun select-between (lst a b)
2   (mapcan #'(lambda (x) (if (and (> x a) (< x b)) (list x))) lst)
3 )
```

2.6 Задание 7

Написать функцию, вычисляющую декартово произведение двух своих списков-аргументов. (Напомним, что $A \times B$ это множество всевозможных пар (a, b) , где a принадлежит A , b принадлежит B .)

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

2.7 Задание 8

Почему так реализовано `reduce`, в чем причина? `(reduce #' + 0) -> 0` - не работает, второй аргумент `reduce` должен быть последовательностью

`(reduce #' + ()) -> 0` - так как `(+)` -> 0, аналогично `(reduce #' * ()) -> 1` так как `(*)` -> 1

2.8 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 lst-sub-len (lst)
2   (reduce #' + (mapcar 'length lst))
3 )
```