

	<p>Министерство образования и науки 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)</p>
---	--

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Отчет по лабораторной работе № 8

По курсу: "Функциональное и логическое
программирование"

Студент:

Турсунов Жасурбек Рустамович

Группа: ИУ7-66Б

Преподаватели:

Толпинская Наталья Борисовна

Строганов Юрий Владимирович

Москва, 2021 г.

Содержание

Введение	2
Задание 2	4
Задание 3	5
Задание 4	6
Задание 5	7
Ответы на вопросы	8

Цель работы: приобрести навыки использования функционалов.

Задачи работы: описать и сравнить работу функционалов: `apply`, `funcall`, `mapcar`, `maplist`

Введение

Для организации многократных вычислений в Lisp могут быть использованы функционалы - функции, которые особым образом обрабатывают свои аргументы. Функционалы - это функции более высокого порядка, так как они в качестве своего первого аргумента принимают функциональный объект - функцию, имеющую имя(глобально определенную функцию), или функцию, не имеющую имени(локально определенную функцию). При использовании функционального объекта должно быть использовано замыкание контекста функции, которым обеспечивается связывание свободных переменных со значениями. В Lisp используются применяющие и отображающие функционалы, являющиеся предикатами, функционалы, использующие предикаты в качестве функционального объекта

Задание 1

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

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

```
1 (defun mult_num (lst k)
2   (mapcar
3     #'(lambda (el) (* el k))
4     lst
5   )
6 )
```

Листинг 1: Функция умножения для списка из чисел

lst - входной список, **k** - число, на которое выполняется умножение.

С помощью `mapcar` осуществляется формирование списка, состоящего из элементов списка-аргумента, умноженных на число-аргумент.

```
1 (defun mult_all (lst k)
2   (mapcar #'(lambda (el)
3     (cond
4       (
5         (numberp el)
6         (* el k)
7       )
8       (
9         (atom el)
10        el
11      )
12      (
13        (mult_all el k)
14      )
15    )
16   ) lst
17 )
```

Листинг 2: Функция умножения для списка из любых объектов

lst - входной список, **k** - число, на которое выполняется умножение.

С помощью `mapcar` осуществляется проход по всему списку и проверка типа каждого элемента:

- если он является числом, то производится умножение на число-аргумент функции и возвращается результат;
- если он является атомом, то он возвращается;
- если он является списком, то к нему рекурсивно применяется текущая функция.

Примеры работы:

```

1      > (mult_num nil 1)
2      NIL
3      > (mult_num '(1) 3)
4      (3)
5      > (mult_num '(1 2 3 4 5) 3)
6      (3 6 9 12 15)
7      > (mult_all nil 2)
8      NIL
9      > (mult_all '(1) 2)
10     (2)
11     > (mult_all '(1 2 3) 3)
12     (3 6 9)
13     > (mult_all '((1 2) 3 (4 (5))) 3)
14     ((3 6) 9 (12 (15)))
15     > (mult_all '(1 2 (a)) 3)
16     (3 6 (A))

```

Задание 2

Напишите функцию, `select-between`, которая из списка-аргумента, содержащего только числа, выбирает только те, которые расположены между двумя указан-

ными границами-аргументами и возвращает их в виде списка.

```
1 (defun select-between (lst a b)
2   (remove-if #'(lambda (x)
3     (and
4       (or (< x a) (> x b))
5       (or (> x a) (< x b))
6     )
7   ) lst
8 )
9 )
```

Листинг 3: Функция выборки из списка по границам

lst - входной список, **a**, **b** - границы выбора.

С помощью функционала `remove-if` осуществляется проход по всему списку-аргументу и удаление тех элементов списка, которые не входят в заданные границы-аргументы.

Примеры работы:

```
1 > (select-between nil 1 2)
2 NIL
3 > (select-between '(1) 0 2)
4 (1)
5 > (select-between '(1 2 3) 4 5)
6 NIL
7 > (select-between '(1 2 3 4 5) 2 4)
8 (2 3 4)
9 > (select-between '(1 2 3 4 5) 4 2)
10 (2 3 4)
```

Задание 3

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

Реализация с помощью функционалов и рекурсии:

```

1 (defun dec_all (lst)
2   (mapcar #'(lambda (el)
3     (cond
4       (
5         (numberp el)
6         (- el 10)
7       )
8       (
9         (atom el)
10        el
11      )
12      (
13        (dec_all el)
14      )
15    )
16  ) lst
17 )
18 )

```

Листинг 4: Функция уменьшения всех чисел списка на 10

lst - список.

В реализации осуществляется проход по всему списку, проверка типа каждого элемента:

- если он является числом, то из него вычитается 10 и выделяется списочная ячейка;
- если он является атомом, то под него выделяется списочная ячейка;
- если он является списком, то к нему рекурсивно применяется текущая функция.

Задание 4

Написать функцию, которая возвращает первый аргумент списка-аргумента, который сам является непустым списком.

Реализация с помощью рекурсии:

```
1 (defun get_first_list (lst)
2   (cond
3     (
4       (and
5         (listp (car lst))
6         (not (null (car lst))))
7     )
8     (car lst)
9   )
10  (
11    (get_first_list (cdr lst))
12  )
13 )
14 )
```

Листинг 5: Функция поиска первого непустого списка

lst - список.

Реализация с помощью рекурсии эффективнее и по памяти, и по скорости, так как не создает вспомогательных списков и возвращает непустой список сразу же, как он встретится.

Примеры работы:

```
1 > (get_first_list '(1 2 (3) (4 5 (6))))
2 (3)
3 > (get_first_list '(((1 2) 3) 4 5))
4 ((1 2) 3)
5 > (get_first_list '(nil 2 (nil (5)) 6))
6 (NIL (5))
```

Задание 5

Что будет результатом (mapcar 'вектор '(570-40-8))?

Результат: (#(|570-40-8|))

Ответы на вопросы:

1) Варианты использования функционалов

Функционалы используются для вычисления одной функции с разными входными значениями. Например, для применения какой-то операции ко всему списку. Некоторые функционалы используют повторные вычисления, что позволяет организовывать такие вычисления на этих функционалах (например, `mapcar`)