

## ЛАБОРАТОРНАЯ РАБОТА 2

### *Описание простейших рекурсивных функций в языке Лисп*

#### 1. Цель и задачи.

Целью работы является изучение основных правил написания рекурсивных функций в функциональном языке.

Основные задачи :

- На примере muLISP'a (newLISP'a) научиться формулировать условие завершения рекурсии, описывать формирование результата функции и новых значений аргументов для рекурсивного вызова;
- Получить практические навыки работы со списочными структурами в выбранной реализации языка Лисп.

#### 2. Краткие сведения о рекурсивных функциях.

В ЛИСПе рекурсия - основной метод программирования. Как и в других языках программирования, описание рекурсивных функций требует особой тщательности. Во-первых, функция обязательно должна содержать хотя бы одно условие окончания рекурсии, во-вторых, значение аргумента при рекурсивных вызовах должно изменяться.

Очень часто условием окончания рекурсии является пустой список-аргумент. В процессе работы все элементы исходного списка обрабатываются последовательно до тех пор, пока не будет достигнут конец списка, т.е. пока список не пуст. Так, в muLISP'e указанное типовое условие окончания рекурсивной обработки можно записать как :

((NULL lst) <действие>)

Второе требование к рекурсивной функции, касающееся изменения аргумента при рекурсивных вызовах, чаще всего обеспечивается рекурсивным вызовом с аргументом-хвостом списка.

Рассмотрим пример описания рекурсивной функции в muLISP'e.

Пример. Описать функцию, которая проверяла бы поэлементную эквивалентность двух списков.

Из задачи ясно, что формальных параметров у функции должно быть два и оба – списки. Назовем параметры lst1 и lst2.

Функция должна возвращать значение Т, если длина списков одинакова и совпадают все элементы: первые, вторые, и т. д.

Определим сколько и каких условий окончания рекурсии должно присутствовать в функции. Благополучный исход, при котором функция возвращает Т, достигается при поэлементном совпадении lst1 и lst2. При этом должно выполняться условие :

(and (null lst1)(null lst2))

Два условия завершения рекурсивного вызова надо предусмотреть для неблагоприятного исхода, когда функция возвращает NIL. Это произойдет, во-первых, если один список короче другого :

(or (and (null lst1)(not (null lst2)))  
(and (not (null lst1))(null lst2)))

И, во-вторых, если найдены несовпадающие элементы :

(NOT (EQUAL (CAR lst1)(CAR lst2)))

Если ни одно из условий окончания рекурсии не выполняется, то функция вызывает себя же уже для хвостов списка.

Описание функции на muLISPe :

```
(defun eqlists (lst1 lst2)
  ((and (null lst1)(null lst2)) T)
  ((or (and (null lst1)(not (null lst2)))
        (and (not (null lst1))(null lst2))
        (not (equal (car lst1)(car lst2)))) nil)
  (eqlists (cdr lst1)(cdr lst2)))
```

Для сравнения описание той же функции на newLISP-tk.

```
(define (eqlists lst1 lst2)
  (cond
    ((and (null? lst1)(null? lst2)) true)
    ((or (and (null? lst1)(not (null? lst2)))
          (and (not (null? lst1))(null? lst2))
          (not (= (first lst1)(first lst2))))
     ) nil)
  (true (eqlists (rest lst1)(rest lst2))
  )
  )
)
```

### 3. Задание на лабораторную работу.

- Ознакомиться по лекционному материалу с описанием рекурсивных функций в Лиспе. Выполнить примеры.
- Описать функцию в соответствии со своим вариантом задания из Таблицы 1, вариант выдает преподаватель.

Таблица 1. Варианты заданий.

Вариант.	Задача.
1.	Описать функцию, которая для заданного списка lst формирует список-результат путем объединения результата реверсирования lst, результата реверсирования хвоста lst, результата реверсирования хвоста хвоста lst и так далее. Пример : для списка '(1 2 3 4 5 6) результатом будет : '(6 5 4 3 2 1 6 5 4 3 2 6 5 4 3 6 5 4 6 5 6).
2.	Есть список lst и два произвольных лисповских объекта obj1 и obj2. Описать функцию, которая формирует новый список путем замены в списке lst всех вхождений объекта obj1 объектом obj2.
3.	Описать функцию, которая находила бы сумму всех числовых элементов списка с учетом наличия подсписков. Пример : для списка '(1 ((2 3) 4) 5 6) результатом будет 21.
4.	Описать функцию, которая на основе списка чисел формирует список-результат следующим образом : первый элемент есть произведение элементов списка, второй – произведение элементов хвоста, третий – произведение элементов хвоста хвоста и так далее. Пример : для списка '(1 2 3 4 5 6) результатом будет : '(720 720 360 120 30 6).
5.	Реализовать функцию включения объекта на заданное место в списке (нумерация элементов – от начала списка).

Продолжение таблицы 1.

Вариант.	Задача.
6.	Реализовать функцию, которая в исходном списке заменяет все элементы-списки результатами их реверсирования. Реверсирование производить на всех уровнях вложения. Пример : для списка '(1 ((2 3) 4) 5 6) результатом будет : '(1 (4 (3 2)) 5 6).
7.	Реализовать функцию, которая выдавала бы элемент списка по заданному номеру с конца.
8.	Дан список lst и число n. Реализовать функцию, которая удаляет все i+n – e элементы списка.
9.	Даны списки lst1 и lst2. Реализовать функцию, которая удаляет из lst1 все элементы-списки, которые соответствуют тому же множеству, что и lst2. Пример : для списков : lst1='(1 (2 2 3) 4 (3 2 3) 5), lst2='(3 2 3 2) результатом будет '(1 4 5).
10.	Реализовать функцию, возвращающую Т в том случае, если одинаковые атомы расположены в исходных списках в одном и том же порядке.
11.	Реализовать функцию, меняющую местами первый и последний элементы исходного списка.
12.	Описать функцию, которая, выдавала бы атомарный элемент списка по заданному номеру n, считая от начала. Пример : для списка '((2) (3) 4 5 a (e r) g) и n=3 результатом будет a.
13.	Написать функцию подсчета числа элементов-списков исходного списка на всех уровнях вложения.
14.	Описать функцию, которая создавала бы список только из числовых элементов списка-аргумента. Список может содержать подсписки произвольной глубины.
15.	Опишите функцию, которая из исходного списка формирует список, содержащий только символьные атомы.
16.	Описать функцию, которая вставляла бы на заданное место элементы второго списка-аргумента.
17.	Описать функцию, аргументами которой являются два списка, а результатом список, содержащий элементы первого списка, не принадлежащие второму списку.
18.	Описать функцию, которая в заданном списке заменяет все элементы-списки значениями сумм входящих в них числовых элементов с учетом вложенных подсписков.
19.	Описать функцию, которая в заданном списке заменяет все элементы-списки значениями количества входящих в них элементов-символов с учетом вложенных подсписков.
20.	Описать функцию, которая для заданного списка проверяет, является ли он отсортированным по возрастанию (убыванию).
21.	Опишите функцию, аргументами которой являются два множества, а результатом - множество, содержащее элементы, принадлежащие только одному из исходных множеств.

## **5. Содержание отчета по лабораторной работе.**

Отчет по лабораторной работе должен содержать :

- формулировку цели и задач;
- результаты выполнения заданий по пунктам, обоснование выбранных структур функций, включая условие окончания рекурсии в каждом случае и формирование новых значений аргументов при рекурсивном вызове;
- выводы по проделанной реализации.

## **Литература.**

Хювенен Э., Сеппянен Й. Мир Лиспа. В 2-х т. Пер. с финск. – М.: Мир, 1990.

Lutz Mueller newLISP For BSDs, Linux, Mac OS X, Solaris and Win32. Users Manual and Reference // [http://www.newlisp.org/downloads/manual\\_frame.html](http://www.newlisp.org/downloads/manual_frame.html)