

Лабораторная работа 5.

Методические указания

Использование управляющих структур, работа со списками

Цель работы: приобрести навыки работы с управляющими структурами Lisp.

Задачи работы: изучить работу функций с произвольным количеством аргументов, функций разрушающих и неразрушающих структуру исходных аргументов.

Краткие теоретические сведения

Многие стандартные функции Lisp являются формами и реализуют особый способ работы со своими аргументами. К таким функциям относятся функции, позволяющие работать с произвольным количеством аргументов или особым образом обрабатывающие свои аргументы: функции cond, and, or, if, let, append, remove, reverse, substitute и др.

Если на вход функции подается структура данных (список), то возникает вопрос: сохранится ли возможность в дальнейшем работать с исходными структурами, или они изменятся в процессе реализации функции. В Lisp существуют функции, использующие списки в качестве аргументов и разрушающие или не разрушающие структуру исходных аргументов при этом часть из них позволяет использовать произвольное количество аргументов, а часть нет. Списки-аргументы могут быть одноуровневыми, смешанными или структурированными.

Указания к выполнению работы

При выполнении лабораторной работы следует

- изучить правила и особенности работы функций: append, remove, reverse, substitute и др.

Отчет по лабораторной сдается в письменной форме по окончании работы.

Задания:

Теор вопросы:

1. структуроразрушающие и не разрушающие структуру списка функции,
2. отличие в работе функций cons, list, append, ncons и в их результате

Common Lisp

1. Написать функцию, которая по своему списку-аргументу lst определяет

является ли он палиндромом (то есть равны ли `lst` и `(reverse lst)`)).

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

3. Напишите свои необходимые функции, которые обрабатывают таблицу из 4-х точечных пар:

(страна . столица), и возвращают по стране - столицу, а по столице — страну .

4. Напишите функцию `swap-first-last`, которая переставляет в списке-аргументе первый и последний элементы.

5. Напишите функцию `swap-two-element`, которая переставляет в списке- аргументе два указанных своими порядковыми номерами элемента в этом списке.

6. Напишите две функции, `swap-to-left` и `swap-to-right`, которые производят одну круговую перестановку в списке-аргументе влево и вправо, соответственно.

7. Напишите функцию, которая добавляет к множеству двухэлементных списков новый двухэлементный список, если его там нет.

8. Напишите функцию, которая умножает на заданное число-аргумент первый числовой элемент списка из заданного 3-х элементного списка-аргумента, когда

а) все элементы списка --- числа,

б) элементы списка -- любые объекты.

9. Напишите функцию, `select-between`, которая из списка-аргумента из 5 чисел выбирает только те, которые расположены между двумя указанными границами-аргументами и возвращает их в виде списка (упорядоченного по возрастанию списка чисел (+ 2 балла)).