

Лабораторная работа 5

Методические указания

Использование функционалов

Цель работы: приобрести навыки использования функционалов.

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

Краткие теоретические сведения

Для организации многократных вычислений в Lisp могут быть использованы функционалы — функции (формы), которые особым образом обрабатывают свои аргументы. Функционалы это функции более высокого порядка, т.к. 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, функционалы, являющиеся предикатами, функционалы, использующие предикаты в качестве функционального объекта.

Указания к выполнению работы

При выполнении лабораторной работы следует

- изучить правила и особенности работы функционалов: `apply`, `funcall`, `mapcar`, `maplist` и др.

Электронная версия лабораторной работы защищается по окончании работы.

Common Lisp

Используя функционалы:

1. Напишите функцию, которая уменьшает на 10 все числа из списка-аргумента этой функции, проходя по верхнему уровню списковых ячеек. (* Список смешанный структурированный)
2. Написать функцию которая получает как аргумент список чисел, а возвращает список квадратов этих чисел в том же порядке.
3. 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.

4. Написать функцию, которая по своему списку-аргументу `lst` определяет является ли он палиндромом (то есть равны ли `lst` и `(reverse lst)`), для одноуровневого смешанного списка.
5. Используя функционалы, написать предикат `set-equal`, который возвращает `t`, если два его множества-аргумента (одноуровневые списки) содержат одни и те же элементы, порядок которых не имеет значения.
6. Напишите функцию, `select-between`, которая из списка-аргумента, содержащего только числа, выбирает только те, которые расположены между двумя указанными числами - границами-аргументами и возвращает их в виде списка (упорядоченного по возрастанию (+ 2 балла)).
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 , принадлежит B .)
8. Почему так реализовано `reduce`, в чем причина?
`(reduce #' + ()) -> 0`
`(reduce #' * ()) -> 1`
9. * Пусть `list-of-list` список, состоящий из списков. Написать функцию, которая вычисляет сумму длин всех элементов `list-of-list` (количество атомов), т.е. например для аргумента
`((1 2) (3 4)) -> 4.`