

Лабораторная работа 4.

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: and, or, append, или особым образом обрабатывающие свои аргументы: функции cond, if, 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.

Указания к выполнению работы

При выполнении лабораторной работы следует

- изучить правила и особенности работы функций: cond, if, and, or, append, remove, reverse, substitute и др.

Отчет по лабораторной сдается в письменной форме по окончании работы.

Задания:

Теор вопросы:

1. синтаксическая форма и хранение программы в памяти,
2. Трактровка элементов списка,
3. Порядок реализации программы,
4. Способы определения функции

Практические задания (Common Lisp):

1. Чем принципиально отличаются функции cons, list, append?

Пусть (setf lst1 '(a b))
(setf lst2 '(c d)).

Каковы результаты вычисления следующих выражений?

(cons lst1 lst2)
(list lst1 lst2)
(append lst1 lst2)

2. Каковы результаты вычисления следующих выражений, и почему?

(reverse |())
(last ())
(reverse '(a))
(last '(a))
(reverse '((a b c)))
(last '((a b c)))

3. Написать, по крайней мере, два варианта функции, которая возвращает последний элемент своего списка-аргумента.

4. Написать, по крайней мере, два варианта функции, которая возвращает свой список-аргумент без последнего элемента.

5. Написать простой вариант игры в кости, в котором бросаются две правильные кости. Если сумма выпавших очков равна 7 или 11 -- выигрыш, если выпало (1,1) или (6,6) --- игрок право снова бросить кости, во всех остальных случаях ход переходит ко второму игроку, но запоминается сумма выпавших очков. Если второй игрок не выигрывает абсолютно, то выигрывает тот игрок, у которого больше очков. Результат игры и значения выпавших костей выводить на экран с помощью функции print.