

Лабораторная работа 6

Методические указания

Рекурсивные функции

Цель работы: приобрести навыки организации рекурсии в Lisp

Задачи работы: изучить способы организации хвостовой, дополняемой, множественной, взаимной рекурсии и рекурсии более высокого порядка в Lisp.

Краткие теоретические сведения

Рекурсия — это ссылка на определяемый объект во время его определения. Т.к. в Lisp используются рекурсивно определенные структуры, то рекурсия — это естественный принцип обработки таких структур. Существуют типы рекурсивных функций: хвостовая, дополняемая, множественная, взаимная рекурсия и рекурсия более высокого порядка.

При организации рекурсии можно использовать как функции с именем, так и локально определенные с помощью лямбда выражений. Кроме этого, при организации рекурсии можно использовать функционалы или использовать рекурсивную функцию внутри функционала.

Указания к выполнению работы

При выполнении лабораторной работы следует

- изучить правила и особенности организации рекурсивных функций разных видов: хвостовая, дополняемая, множественная, взаимная рекурсия и рекурсия более высокого порядка.

Электронная версия лабораторной работы защищается по окончании работы.

Common Lisp

Используя рекурсию:

1. Написать хвостовую рекурсивную функцию `my-reverse`, которая развернет верхний уровень своего списка-аргумента `lst`.
3. Написать функцию, которая возвращает первый элемент списка -аргумента, который сам является непустым списком.
4. Написать функцию, которая выбирает из заданного списка только те числа, которые больше 1 и меньше 10.
(Вариант: между двумя заданными границами.)

ThisCanThrow simply dec
takes a form which begin
complete the operation be
along its execution path tha
test the strong guarantee: 12

```
extern int gThrowCounte  
(void thisCanThrow()  
if (gThrowCounter--  
throw 0;  
// the class value, clas  
// back (const value
```

7. Напишите рекурсивную функцию, которая умножает на заданное число-аргумент все числа из заданного списка-аргумента, когда
 - а) все элементы списка --- числа,
 - б) элементы списка -- любые объекты.
8. Напишите функцию, select-between, которая из списка-аргумента, содержащего только числа, выбирает только те, которые расположены между двумя указанными границами-аргументами и возвращает их в виде списка (упорядоченного по возрастанию списка чисел (+ 2 балла)).
8. Написать рекурсивную версию (с именем rec-add) вычисления суммы чисел заданного списка:
 - а) одноуровневого смешанного,
 - б) структурированного.
9. Написать рекурсивную версию с именем recnth функции nth.
10. Написать рекурсивную функцию allodd, которая возвращает t когда все элементы списка нечетные.
11. Написать рекурсивную функцию, которая возвращает первое нечетное число из списка (структурированного), возможно создавая некоторые вспомогательные функции.
12. Используя cons-дополняемую рекурсию с одним тестом завершения, написать функцию которая получает как аргумент список чисел, а возвращает список квадратов этих чисел в том же порядке.