

Отчет по лабораторной работе №2 по курсу «Функциональное программирование»

Студент группы 8О-306 Гаврилов Максим, № по списку 7.

Контакты: sobraj@yandex.ru

Работа выполнена: 22.03.2022

Преподаватель: Иванов Дмитрий Анатольевич, доц. каф. 806

Отчет сдан:

Итоговая оценка:

Подпись преподавателя:

1. Тема работы

Простейшие функции работы со списками Коммон Лисп

2. Цель работы

Научиться конструировать списки, находить элемент в списке, использовать схему линейной и древовидной рекурсии для обхода и реконструкции плоских списков и деревьев.

3. Задание (вариант № 2.12)

Запрограммируйте рекурсивно на языке Коммон Лисп функцию, подсчитывающую число вхождений заданного целого числа в дерево.

Примеры

`(count-int 3 '((1 2) 3 (4 5 (3 6)) 7)) => 2`

4. Оборудование студента

процессор AMD FX(tm)-6300 Six-Core Processor 3.50 GHz, память 16ГБ, 64-разрядная система.

5. Программное обеспечение

ОС Windows 10, программа portacle, версия slime 2.24

6. Идея, метод, алгоритм

Древовидно-рекурсивный процесс

7. Сценарий выполнения работы

1. Изучить функции lisp для работы со списками.
2. Построить функцию, выполняющую рекурсивный просмотр дерева.
3. Провести тестирование работы построенной функции, используя пример из задания.

8. Распечатка программы и её результаты

Программа

```
(defun count-int (a lst)
  (+ (if (consp (first lst))
        ;;проверка первого элемента
        (count-int a (first lst))
        ;;если дерево ветвится дальше, переход на уровень ниже
        (if (= (first lst) a)
            ;;если на первом месте элемент, его проверка
            1
            0))
     (if (consp (rest lst));поиск в хвосте списка
         (count-int a (rest lst))
         0)))
```

Результаты

```
; SLIME 2.24
CL-USER> (defun count-int (a lst)
  (+ (if (consp (first lst))
        ;;проверка первого элемента
        (count-int a (first lst))
        ;;если дерево ветвится дальше, переход на уровень ниже
        (if (= (first lst) a)
            ;;если на первом месте элемент, его проверка
            1
            0))
     (if (consp (rest lst));поиск в хвосте списка
         (count-int a (rest lst))
         0)))

COUNT-INT
CL-USER> (count-int 3 '((1 2) 3 (4 5 (3 6)) 7)) ;пример из задания
2
CL-USER> (count-int 3 '((1 3 ((1 3) 8 3) 2) 3 (4 5 (3 6 (1 3))) 7 1
1 3)) ;более сложный пример
7
CL-USER> (count-int 1 '((1 3 ((1 3) 8 3) 2) 3 (4 5 (3 6 (1 3))) 7 1
1 3)) ;поиск другого числа
5
CL-USER> (count-int 1 '((((1 2))))
1
CL-USER> (count-int 1 '((((4 2))))
0
CL-USER>
```

9. Дневник отладки

№	Дата, время	Событие	Действие по исправлению	Примечание
1				

10. Замечания автора по существу работы

Сначала я написал программу, использующую переменную, сохраняющую количество найденных в процессе работы функции совпадений. Однако мне сразу показалось, что это как-то не соответствует стилистике Лиспа, так что я продолжил попытки выполнить задачу иначе, и в итоге у меня получилось реализовать программу, содержащую только простую рекурсию.

11. Выводы

В данной лабораторной работе я получил опыт написания функций на языке Коммон Лисп без предварительно заданных шаблонов. Я реализовал рекурсивную функцию, подсчитывающую число вхождений заданного целого числа в дерево. Программа работает правильно и прошла все тесты.