

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

Студент группы 8О-308Б Ивенкова Любовь, № по списку 6.

Контакты: lyubov.iven@mail.ru

Работа выполнена: 04.04.2022

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 № 2.38)

Запрограммируйте рекурсивно на языке Коммон Лисп функцию индивидуумы (l), которая принимает в качестве аргумента список атомов l и выдаёт в результате список всех атомов, которые встречаются в l ровно один раз.

Примеры

```
(индивидуумы ' (a b c a d c e) ) => ' (b d e)
```

4. Оборудование ПЭВМ студента

PC, процессор Intel Core i5-4460 @ 3,2 GHz (32 x 100), память 11,7 Gb, 64-разрядная система.

5. Программное обеспечение ЭВМ студента

OS Windows 10, программа LispWorks 7.1 Personal (64-bit).

6. Идея, метод, алгоритм

Будем просматривать список поэлементно. Если изначальный список пустой, то возвращается nil. Иначе, берём голову списка и смотрим, встречается ли она у нас в хвосте. Если нет, то рекурсивно задаём новый список: голова + (индивидуумы хвост). Если же символ головы встречается в хвосте, то удаляем все эти вхождения из него и задаём новый список: (индивидуумы (хвост-вхождения)).

7. Сценарий выполнения работы

Я не была уверена, можно ли в этой лабораторной работе использовать функцию remove, так что на всякий случай реализовала свой аналог этой функции также с помощью рекурсии.

8. Распечатка программы и её результаты

Программа

```
(defun myremove (item lst)
```

```

(cond ((atom lst)
      lst)
      ((equal item (first lst))
       (myremove item (rest lst)))
      (t
       (cons (myremove item (first lst))
              (myremove item (rest lst))))))

(defun индивидуумы (lst)
  (if (null lst) nil
      (if (member (first lst) (rest lst))
          (индивидуумы (myremove (first lst) (rest lst)))
          (cons (first lst) (индивидуумы (rest lst))))))

```

Результаты

```

CL-USER 1 > (индивидуумы '(a b c a d c e))
(B D E)
CL-USER 2 > (индивидуумы '(A B 1 2 1 B))
(A 2)
CL-USER 3 > (индивидуумы '())
NIL

```

9. Дневник отладки

№	Дата, время	Событие	Действие по исправлению	Примечание

10. Замечания автора по существу работы

Программа проходит по всему списку, для каждого элемента проверяя его вхождение в хвост с помощью функции `member`, имеющую линейную сложность. Также иногда происходит удаление, работающее также за $O(n)$. В итоге получаем сложность $O(n^3)$.

11. Выводы

В процессе выполнения работы я познакомилась со списками и простейшими функциями работы с ними в языке Common Lisp. Работать с ними было довольно легко, т.к. синтаксис и определения очень похожи на работу со списками в языке Prolog.