

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

Студент группы М8О-306 МАИ *Дубинин Артем*, №5 по списку  
Контакты: [rusartdub@gmail.com](mailto:rusartdub@gmail.com)  
Работа выполнена: 05.04.2020

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 №2.40)

Запрограммируйте рекурсивно на языке Коммон Лисп функцию частоты, принимающую в качестве аргумента список атомов.

Функция должна возвращать список всех атомов, встречающихся в x вместе с частотой их появления, см. таблицу.

Так как результат представляет собой множество пар, несущественно, в каком порядке эти пары расположены.

x	частоты
(A B A B A C A)	((A 4) (B 2) (C 3))
(НИ ТО НИ СЁ)	((НИ 2) (ТО 1) (СЁ 1))

## 4. Оборудование студента

Ноутбук Dell Vostro 5568, процессор Intel Core i5-7200U @ 4x 3.1GHz, память: 8Gb, разрядность системы: 64.

## 5. Программное обеспечение

ОС Ubuntu 18.04 bionic, компилятор sbcl, текстовый редактор Atom.

## 6. Идея, метод, алгоритм

Заводим хэш таблицу вида  $\langle \text{Атом} \rangle \Rightarrow \langle \text{Кол-во вхождений в текст} \rangle$ , где будем считать кол-во повторяющихся атомов. Пройдем рекурсивно список, беря головной элемент списка и записывая его в хэш таблицу в виде ключа, инкрементируя кол-во вхождений этого элемента в тексте. Если до этого нам не встречался этот элемент, то записываем ноль в элемент хэш таблицы и инкрементируем на 1:  $\langle \text{Атом} \rangle \Rightarrow \langle 1 \rangle$ . Далее проходимся по хэш таблице и записываем в результат.

Например:

Число (НИ ТО НИ СЁ), берем головной элемент НИ, записываем в хэш таблицу,  $\langle \text{НИ} \rangle \Rightarrow \langle 1 \rangle$ . Далее рекурсивно запускаем нашу функцию прохождения по списку от (ТО НИ СЁ). Берем ТО,  $\langle \text{ТО} \rangle \Rightarrow \langle 1 \rangle$ . Тем же образом спускаемся к 3-му элементу списка - НИ,  $\langle \text{НИ} \rangle \Rightarrow \langle 2 \rangle$ . И в конце берем элемент СЁ,  $\langle \text{СЁ} \rangle \Rightarrow \langle 1 \rangle$ . Проходимся по хэш таблице и записываем результат.

((НИ 2) (ТО 1) (СЁ 1))

## 7. Сценарий выполнения работы

## 8. Распечатка программы и её результаты

### 8.1. Исходный код

```
(defun add-to-hash (h word)
  (setf (gethash word h)
        (+ 1 (or (gethash word h) 0))))

(defun iterate-through-list (h l)
  (when l
    (add-to-hash h (car l))
    (iterate-through-list
     h (cdr l))))

(defun word_counter (l &aux (ht (make-hash-table)))
  (iterate-through-list ht l)
  (let ((results (list)))
    (maphash #'(lambda (key val)
                  (setf results
                        (append results
```

```

                                (list (list key val))))))
                                ht)
results))

;(print (word_counter '(A B A B A C A)))

```

## 8.2. Результаты работы

```

*
* (word_counter '(A B A B A C A))
((A 4) (B 2) (C 1))

* (word_counter '(НИ ТО НИ СЁ))
((НИ 2) (ТО 1) (СЁ 1))

```

## 9. Замечания автора по существу работы

Чтобы выполнить работу, нужно было хорошо разобраться с работой со списками в лиспе, и как работать с хэш таблицей в common lisp.

## 10. Выводы

При выполнении этой работы хотелось всегда воспользоваться итеративным парадигмой, так как лисп мультипарадигмальный язык и в нем в виде макросов реализованы циклы и тд. Но работа в функциональной парадигме оказалась интересна, особенно интересно было видеть, что всё можно выполнить в функциях – циклы, запись и тд..