

# Отчет по лабораторной работе № 4 по курсу «Функциональное программирование»

Студент группы 8О-307Б-18 МАИ *Тояков Артем*, №22 по списку

Контакты: `temathesuper@mail.ru`

Работа выполнена: 23.04.2021

Преподаватель: Иванов Дмитрий Анатольевич, доц. каф. 806

Отчет сдан:

Итоговая оценка:

Подпись преподавателя:

## 1. Тема работы

Знаки и строки.

## 2. Цель работы

Изучить знаки и строки, а также методы работы с ними в Коммон Лисп.

## 3. Задание (вариант № 4.11)

Запрограммировать на языке Коммон Лисп функцию, принимающую один аргумент - предложение. Функция должна возвращать число слов в этом предложении, у которых первый и последний знак совпадают. Сравнение как латинских букв, так и русских должно быть регистро-независимым.

## 4. Оборудование студента

Процессор: Intel(R) Core(TM) i7-8565U CPU @ 1.80GHz, память: 3,8 Gb, разрядность системы: 64.

## 5. Программное обеспечение

UBUNTU 18.04.5 LTS, компилятор sbcl

## 6. Идея, метод, алгоритм

Идея в том, чтобы пройти по всему предложению, разделяя его по словам, сравнивать первую и последнюю буквы всех слов, и в тех случаях, где они равны, увеличивать счётчик на 1. В конце вывести результат - переменную счётчик.

В программе есть одна основная функция (`defun count-words-with-start-eq-end(str)`), в

которой в начале объявляются переменные, а затем с помощью цикла `loop` и двух итераторов `i`, `j` реализовано разбиение на слова и сравнение букв. Один проход по внешнему циклу - проверка одного слова.

## 7. Сценарий выполнения работы

- Анализ возможных реализаций поставленной задачи на Коммон Лисп
- Изучение синтаксиса и основных функций работы со знаками и строками Коммон Лисп
- Реализация поставленной задачи на Коммон Лисп

## 8. Распечатка программы и её результаты

### 8.1. Исходный код

```
(defun whitespace-char(ch)
  (member ch '(#\Space #\Tab #\Newline))
)

(defun russian-upper-case-p (char)
  (position char "АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ")
)

(defun russian-char-downcase (char)
  (let ((i (russian-upper-case-p char)))
    (if i
      (char "абвгдеёжзийклмнопрстуфхцчшщъыьэюя" i)
      (char-downcase char))
    )
)

(defun russian-char-equal (char1 char2)
  (char-equal (russian-char-downcase char1)
    (russian-char-downcase char2))
)

(defun endSentence-char(ch)
  (member ch '(#\! #\? #\. #\"))
)

(defun count-words-with-start-eq-end(str)
  (let
```

```

(
(res 0)
      (i 0)
      (j 0)
(cur-ch-begin nil)
(cur-ch-end nil)
)

(loop while (and (not (endSentence-char (char str i))) (< i (length str))) do
(setq cur-ch-begin (char str i))
(setq j i)
      (loop while (and (< j (length str)) (and (not (whitespace-char (char str j))
(setq j (+ j 1))
      )
      (setq j (- j 1))
      (setq cur-ch-end (char str j))
(if (or (char-equal cur-ch-begin cur-ch-end) (russian-char-equal cur-ch-begin cur-ch-end))
(setq res (+ res 1))
)
(setq i (+ j 1))
      (loop while (and (< i (length str)) (and (not (endSentence-char (char str i))
(setq i (+ i 1))
      )
)
)
(write res)
)
)

```

## 8.2. Результаты работы

```

* (count-words-with-start-eq-end "а роза упала на лапу Азора")
2
2
* (count-words-with-start-eq-end "hepl fuf me")
1
1
* (count-words-with-start-eq-end "К долинам,      я крутой!")
2
2
* (count-words-with-start-eq-end "Двойные кавычки активно используются в русском языке")
3
3
* (count-words-with-start-eq-end "Ала ара как лел мор мом троп?")

```

```

5
5
* (count-words-with-start-eq-end "Америка.")
1
1
* (count-words-with-start-eq-end "На столе вкусная айва или баба пришла, где арка.")
3
3
* (count-words-with-start-eq-end "Hello, worldahaha.")
1
1

```

## 9. Дневник отладки

Дата	Событие	Действие по исправлению	Примечание

## 10. Замечания автора по существу работы

Единственный недочёт: результат выводится 2 раза.

## 11. Выводы

В ходе данной работы мне удалось познакомиться со встроенными функциями/инструментами для работы со знаками и строками. Со строками я был знаком и ранее, однако было довольно интересно увидеть применение такой структуры данных в Коммон Лисп. В моей программе алгоритм работает за линейное время  $O(n)$ , где  $n$  - длина исследуемого предложения.