

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 (txt)`), в

которой в начале объявляются переменные, а затем с помощью дополнительной функции word-list реализовано разбиение на 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-p (char)
  (member char '(#\Space #\Tab #\Newline)))

(defun word-list (string)
  (loop with len = (length string)
        for left = 0 then (1+ right)
        for right = (or (position-if #'whitespace-char-p string :start left) len)
        unless (= right left)
          collect (subseq string left right)
        while (< right len)
  )
)

(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 english-upper-case-p (char)
  (position char "ABCDEFGHIJKLMNOPQRSTUVWXYZ")
)

(defun english-char-downcase (char)
  (let ((i (english-upper-case-p char)))
    (if i
      (char "abcdefghijklmnopqrstuvwxyz" i)
      (char-downcase char))
  )
)

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

(defun count-words-with-start-eq-end (txt)
  (let ((found 0))
    (dolist (sentence txt)
      (dolist (word (word-list sentence))
        (let ((first-char NIL) (last-char NIL))
          (setf first-char (char word 0))
          (setf last-char (char word (- (length word) 1)))

          (let ((rus-res(russian-char-equal first-char last-char)) (eng-res(english
            (if (or rus-res eng-res) (setf found (+ found 1)))
          )
        )
      )
    )
    found)
)

```

8.2. Результаты работы

```
* (count-words-with-start-eq-end '("а роза упала на лапу Азора"))
```

2

```
* (count-words-with-start-eq-end '("Ала ара как лел мор мом троп?"))
```

5

```
* (count-words-with-start-eq-end '("Двойные кавычки активно используются в русском я
```

3

```
* (count-words-with-start-eq-end '("Hello, world ahah."))
```

0

9. Дневник отладки

Дата	Событие	Действие по исправлению	Примечание
06.05.2021	Ошибка: выход за границу памяти массива	Довольно сильное изменение программы, ввод структуры данных: list	

10. Замечания автора по существу работы

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.