

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

Студент группы 8О-307Б-18 МАИ *Токарев Никита*, №21 по списку
Контакты: tokarevnikita08@mail.ru
Работа выполнена: 14.04.2021

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 № 4.9)

Запрограммировать на языке Коммон Лисп функцию с двумя параметрами: `char` - знак, `sentence` - строка предложения. Функция должна подсчитать число вхождений знака `char` в последнее слово предложения `sentence`. Сравнение как латинских букв, так и русских должно быть регистро-независимым.

4. Оборудование студента

Процессор Intel® Core™ i3-5005U CPU @ 2.00GHz × 4, 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 начать поиск от последнего элемента игнорируя различные знаки(.,?,!) в конце предложения и до первого пробела или конца предложения(если предложение состоит из одного слова). В программе одна основная функции:

- (last-word-char-count ch str) - В данной функции происходит поэлементное сравнение элементов с условиями выхода: следующий элемент пробел или начало предложения (если предложение состоит из одного слова).

7. Сценарий выполнения работы

- Анализ возможных реализаций поставленной задачи на common Lisp
- Изучение синтаксиса и основных функций работы со списками common Lisp
- Реализация поставленной задачи на common Lisp

8. Распечатка программы и её результаты

8.1. Исходный код

Значения тестовых функций представлено в исходном коде.

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

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

(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 last-word-char-count(ch str)
  (let
```

```

(
  (res 0)
  (cur-ch nil)
)
(loop for i from (- (length str) 1) downto 0 do
  (setq cur-ch (char str i))
  (if (and (char-equal ch cur-ch) (not (endSentence-char cur-ch)))
    (setq res (+ res 1))
  )
  (if (or (whitespace-char cur-ch) (equalp i 0))
    (return res)
  )
)
)
)

; (last-word-char-count #\Ы "К долинам, покоем объятм, ему не сойти с высоты.")
; (last-word-char-count #\А "\"Почему здесь никого нет, - удивилась Аленка,
- я пришла раньше или опоздала?\")
; (last-word-char-count #\а "\"Почему здесь никого нет, - удивилась Аленка,
- я пришла раньше или опоздала?\")
; (last-word-char-count #\о "Много.")
; (last-word-char-count #\к "Креветка.")
; (last-word-char-count #\к "Креветка!?")
; (last-word-char-count #\к "Двойные кавычки активно используются в
русском языке в машинном тексте!?")
; (last-word-char-count #\у "Двойные кавычки активно используются в
русском языке в машинном тексте!?")
; (last-word-char-count #\f "Hello, my friend!")

```

8.2. Результаты работы

```

* (last-word-char-count #\о "Много.")

2
* (last-word-char-count #\к "Креветка!?")

2
* (last-word-char-count #\Ы "К долинам, покоем объятм, ему не сойти с высоты.")

2
* (last-word-char-count #\у "Двойные кавычки активно используются
в русском языке в машинном тексте!?")

```

```

0
* (last-word-char-count #\a "\"Почему здесь никого нет, - удивилась Аленка,
- я пришла раньше или опоздала?\"")

2
* (last-word-char-count #\A "\"Почему здесь никого нет, - удивилась Аленка,
- я пришла раньше или опоздала?\"")

2
* (last-word-char-count #\f "Hello, my friend!")

1

```

9. Дневник отладки

Дата	Событие	Действие по исправлению	Примечание
14.04.2021	Не было поддержки русского алфавита при компиляции и использовании программы с помощью LispWorks.	Добавил необходимые функции и изменил условие в сравнении знаков	

10. Замечания автора по существу работы

Замечаний нет.

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 common Lisp. В моей программе алгоритм работает за линейное время $O(n)$, где n - длина исследуемого слова.