

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

Студент группы 80-306Б Пивницкий Даниэль Сергеевич, № по списку 13.

Контакты: pivnitskiydaniel@gmail.com

Работа выполнена: 06.06.2022

Преподаватель: Иванов Дмитрий Анатольевич, доц. каф. 806

Отчет сдан:

Итоговая оценка:

Подпись преподавателя:

1. Тема работы

Знаки и строки.

2. Цель работы

Научиться работать с литерами (знаками) и строками при помощи функций обработки строк и общих функций работы с последовательностями.

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

Ноутбук Lenovo Legion 5 Pro, процессор Ryzen 5 5600h, память 16ГБ, 64-разрядная система.

5. Программное обеспечение

OS Windows 10, программа LispWorks Personal Edition 7.1.2

6. Идея, метод, алгоритм

Разбиение предложения на слова, сравнение первого и последнего символа каждого слова, подсчет количества совпавших пар символов. Учитывается, что слово может быть как на русском, так и на английском языках.

7. Сценарий выполнения работы

1. Изучение функций для работы со строками в Коммон Лисп.
2. Изучение функций для работы с русским языком.
3. Написание программы для подсчета количества слов, начинающихся и заканчивающихся на одну и ту же букву.

8. Распечатка программы и её результаты

Программа

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

(defun word-list (string)
  ;; Разбить строки на слова, разделённые знаками whitespace
  ;; A la (split-seq-if #'whitespace-char-p 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) ; исключить пустые слова
        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 count-words-with-start-eq-end (sentence)
  (let ((cnt 0))
    (dolist (word (word-list sentence))
      (let ((first NIL) (last NIL))
        (setf first (char word 0))
        (setf last (char word (- (length word) 1)))
        (if (or (russian-char-equal first last) (char-equal first
                                                                last))
            (setf cnt (+ cnt 1))))))
  cnt))
```

Результаты

```
CL-USER 1 > (defun whitespace-char-p (char)
  (member char '(#\Space #\Tab #\Newline)))
```

Error: Redefining function WHITESPACE-CHAR-P visible from package

```
LISPWORKS { *handle-warn-on-redefinition* is :ERROR }
  1 (continue) Redefine it anyway.
  2 (abort) Return to top loop level 0.
```

Type :b for backtrace or :c <option number> to proceed.
Type :bug-form "<subject>" for a bug report template or :? for other options.

```
CL-USER 2 : 1 > :c 1
WHITESPACE-CHAR-P
```

```
CL-USER 3 > (defun word-list (string)
  ;; Разбить строки на слова, разделённые знаками whitespace
  ;; A la (split-seq-if #'whitespace-char-p 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) ; исключить пустые слова
    collect (subseq string left right)
    while (< right len)))
```

WORD-LIST

```
CL-USER 4 >
(defun russian-upper-case-p (char)
  (position char "АВВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧЩЪЫЬЭЮЯ"))
RUSSIAN-UPPER-CASE-P
```

```
CL-USER 5 >
(defun russian-char-downcase (char)
  (let ((i (russian-upper-case-p char)))
    (if i
      (char "абвгдеёжзийклмнопрстуфхцчщъыьэюя" i)
      (char-downcase char))))
RUSSIAN-CHAR-DOWNCASE
```

```
CL-USER 6 >
(defun russian-char-equal (char1 char2)
  (char-equal (russian-char-downcase char1)
              (russian-char-downcase char2)))
RUSSIAN-CHAR-EQUAL
```

```
CL-USER 7 >

(defun count-words-with-start-eq-end (sentence)
  (let ((cnt 0))
    (dolist (word (word-list sentence))
      (let ((first NIL) (last NIL))
        (setf first (char word 0))
        (setf last (char word (- (length word) 1)))
        (if (or (russian-char-equal first last) (char-equal first
                                                                last))
            (setf cnt (+ cnt 1))))))
  cnt))
```

COUNT-WORDS-WITH-START-EQ-END

```
CL-USER 8 > (count-words-with-start-eq-end "Hello, nothing here")
0
```

```
CL-USER 9 > (count-words-with-start-eq-end "ogo OGO ogO OGo
neogO")
4
```

```
CL-USER 10 >
(count-words-with-start-eq-end "OGO oGoro oГОгO работает")
3
```

```
CL-USER 11 > (count-words-with-start-eq-end "а роза упала на лапу
Азора")
2
```

```
CL-USER 12 > (count-words-with-start-eq-end "Englishe Русскийр")
2
```

9. Дневник отладки

№	Дата, время	Событие	Действие по исправлению	Примечание
1				

10. Замечания автора по существу работы

Стоило учесть, что Коммон Лисп не умеет работать с русскими символами, поэтому их пришлось рассмотреть отдельно. Хорошо, что на лекции были даны реализации всех необходимых для этого функций. Это облегчило мою задачу.

11. Выводы

Выполняя данную работу, я научился работать со строками на языке Коммон Лисп. Этот язык поддерживает все стандартные операции над строками - выделение *i*-ого символа, вычисление длины строки, итд. Поэтому задание не вызвало затруднений.