

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

Студент группы 8О-308Б Ивенкова Любовь, № по списку 6.

Контакты: lyubov.iven@mail.ru

Работа выполнена: 25.04.2022

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 № 4.37)

Запрограммировать на языке Коммон Лисп функцию `find-word`, принимающую два аргумента:

- `word` - строка, представляющая слово,
- `source` - текст.

Если слово найдено, функция должна возвращать два значения с помощью `values`:

1. индекс начала данного слова в предложении,
2. номер первого предложения в тексте, в которое входит слово (нумерация с 0).

Если слово не найдено, функция должна вернуть `NIL`.

4. Оборудование ПЭВМ студента

PC, процессор Intel Core i5-4460 @ 3,2 GHz (32 x 100), память 11,7 Gb, 64-разрядная система.

5. Программное обеспечение ЭВМ студента

ОС Windows 10, программа LispWorks 7.1 Personal (64-bit).

6. Идея, метод, алгоритм

В начале разобьём текст на предложения, а предложения на слова. От слов отсечём знаки препинания. Затем в цикле пройдемся по всем получившимся словами и будем сравнивать каждое с `word`. Когда найдём, находим также индексы первых вхождений слова и предложения и запоминаем их.

7. Сценарий выполнения работы

Я поняла задачу так, что надо реализовать регистронезависимый поиск (и, наверное, желательно и для русского, и для латиницы).

Поэтому для работы мне потребовалось взять много функций из курса – для перевода слов в нижний регистр для латиницы и русского, для сравнения для латиницы и русского, для разбивки текста на предложения и предложения на слова.

После описания всех этих функций я двумя циклами прохожусь по всем предложениям в тексте и всем словам в этих предложениях. Для обработки слов использую `string-right-trim`. Для нахождения индекса слова в предложении перевожу и то, и то в нижний регистр, чтобы обеспечить регистронезависимость поиска. Промежуточные результаты запоминаю в локальных переменных `let`.

8. Распечатка программы и её результаты

Программа

```
(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-string-downcase (string)
  (map 'string #'russian-char-downcase string))

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

(defun russian-string-equal (str1 str2)
  (and (= (length str1) (length str2))
       (loop for char1 across str1
              and char2 across str2
              always (russian-char-equal char1 char2))))

(defun whitespace-char-pp (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-pp string
                                     :start left)
                        len)
        unless (= right left)
        collect (subseq string left right)
        while (< right len)))

(defun find-word (word source)
  (let ((f -1)
        (s -1))
    (dolist (sentence source)
      (dolist (w (word-list sentence))
```

```

      (let ((wword (string-right-trim ",.!:?" (russian-string-
downcase word))))
        (setq w (string-right-trim ",.!:?" (russian-string-
downcase w)))
        (if (russian-string-equal w wword)
            (let () (setq f (search (russian-string-downcase
word) (russian-string-downcase sentence)))
              (setq s (position sentence source))))))
      (if (= f -1) nil (values f s)))

```

Результаты

```

CL-USER 1 > (find-word "ds" '("s Ds. df, d" "fg: ds"))
2
0

```

```

CL-USER 2 > (setq txt '("Вот дом,
Который построил Джек."

```

```

"А это пшеница,
Которая в темном чулане хранится
В доме,
Который построил Джек."

```

```

"А это веселая птица-синица,
Которая часто ворует пшеницу,
Которая в темном чулане хранится
В доме,
Который построил Джек.")
("Вот дом,
Который построил Джек." "А это пшеница,
Которая в темном чулане хранится
В доме,
Который построил Джек." "А это веселая птица-синица,
Которая часто ворует пшеницу,
Которая в темном чулане хранится
В доме,
Который построил Джек.")

```

```

CL-USER 3 > (find-word "Джек" txt)
26
0

```

```

CL-USER 4 > (find-word "воробей" txt)
NIL

```

9. Дневник отладки

№	Дата, время	Событие	Действие по исправлению	Примечание

10. Замечания автора по существу работы

Программа выполняет регистронезависимый поиск слова в тексте. Она работает как для русского языка, так и для английского.

11. Выводы

В процессе выполнения работы я познакомилась со знаками и строками и функциями для работы с ними в языке Common Lisp. Мне эта работа показалась довольно трудной, так как пришлось изучить множество функций и тонкостей работы с ними, часто мне было сложно понять, почему моя функция не работает.