

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

Студент группы 8О-306 МАИ *Наумов Дмитрий*, №15 по списку

Контакты: dandachok@gmail.com

Работа выполнена: 27.04.2020

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 №43)

Запрограммировать на языке Коммон Лисп функцию, принимающую один аргумент - текст.

Функция должна удалить все слова, в которых встречается не более двух различных букв, и вернуть новый текст. Сравнение как латинских букв, так и русских должно быть регистро-независимым.

```
(remove-two-char-words '(" Оно_скрылось_за_деревьями . "
                          " Мы_прошли , _не_заметив_его . " ) )
=> ( " скрылось_деревьями . " " прошли , _заметив_его . " )
```

4. Оборудование студента

Процессор Intel Core i3-2100 4 @ 2.2GHz, память: 8Gb, разрядность системы: 64.

5. Программное обеспечение

ОС Ubuntu 18.04.4 LTS, clisp 2.49.60

6. Идея, метод, алгоритм

Функция `russian-upper-case-p` проверяет, находится ли буква в верхнем регистре, если да, то возвращает ее номер.

`russian-char-downcase` возвращает символ в нижнем регистре.

`russian-string-downcase` возвращает строку в нижнем регистре.

Функция `more-two-symbol-p` возвращает Т, если строка содержит больше двух букв, NIL в противном случае. Реализация довольно простая, помещаем каждую букву в список, если ее там еще нет. Затем проверяем размер списка.

Функция `remove-two-char-words` принимает список строк. В каждой строке выделяем слова разделенные пробелом и проверяем их на количество букв, если букв больше двух, то добавляем слово в результирующую строку.

7. Распечатка программы и её результаты

7.1. Исходный код

```
(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 more-two-symbol-p (string)
  (let ((str (string-downcase (string-right-trim " ,.:!?"
    string))) (symbols (list)))
    (let ((len (length str)))
      (do ((i 0 (+ i 1)))
        ((>= i len))
        (if (not (member (char str i) symbols))
          (setf symbols (append symbols (list (char str i))))))
      (> (length symbols) 2))))

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

```

(defun remove-two-char-words (text)
  (let (ans (list))
    (dolist (sentence text)
      (let ((len (length sentence))
            (right 0)
            (ans-str "")
            (word "")
            (more-two NIL))
        (do ((left 0 (+ left 1)))
          ((>= left len))
            (setf right (or (position-if #'space-p sentence
                                         :start left) len))
            (setf word (subseq sentence left right))
            (setf more-two (more-two-symbol-p word))
            (if more-two
              (setf ans-str (concatenate 'string ans-str word)))
            (if (and (< right len) (= right left))
              (setf ans-str (concatenate 'string ans-str
                                           (make-string 1 :initial-element (char sentence
                                                         right))))))
          (setf left right))
        (setf ans (append ans (list ans-str))))
    ans))

```

7.2. Результаты работы

```

dandachok@dpc:~/Documents/Study/sem6/FP/lab4$ clisp
[1]> (load "solution.lisp")
;; Loading file solution.lisp ...
;; Loaded file solution.lisp
#P"/home/dandachok/Documents/Study/sem6/FP/lab4/solution.lisp"
[2]> (remove-two-char-words '(" Оно скрылось за деревьями." " Мы
прошли, не заметив его."))
(" скрылось деревьями." " прошли, заметив его.")
[7]> (remove-two-char-words '(" Оно она онн"))
(" она ")
[7]> (remove-two-char-words '(" Оно она онн. ja jj jaa adskf."))
(" она adskf.")
[7]> (remove-two-char-words '(" Оно она онн. ja jJ jaA adskf."))
(" она adskf.")

```

8. Дневник отладки

| Дата | Событие | Действие по исправлению | Примечание |
|------------|----------------------------------|---|------------|
| 06.05.2020 | Некорректная работа с кириллицей | Добавлены функции для работы с кириллицей | |

9. Замечания автора по существу работы

10. Выводы

Сложность программы по времени и по памяти линейная.