

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

Студент группы М8О-307Б-18 МАИ *Скворцов Кирилл Алексеевич*, №20

Контакты: `kilyla2@yandex.ru`

Работа выполнена: 04.05.2021

Преподаватель: Иванов Дмитрий Анатольевич, доц. каф. 806

Отчет сдан:

Итоговая оценка:

Подпись преподавателя:

1. Тема работы

Знаки и строки.

2. Цель работы

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

3. Задание (вариант №4.44)

Запрограммировать на языке Коммон Лисп функцию с двумя параметрами:

- `char-bag` - список знаков,
- `text` - текст.

Функция должна вернуть преобразованный текст, во всех предложениях которого все слова "обтёсаны" т.е. в них с обоих концов, удалены знаки, упомянутые в `char-bag`.

4. Оборудование студента

Ноутбук Xiaomi mi Pro 15.6, процессор Intel Core i7-8550U CPU 1.80GHz, память: 8Gb, разрядность системы: 64.

5. Программное обеспечение

ОС Windows 10, онлайн компилятор для `common-lisp`, текстовый редактор VSCode (использовал т.к. там есть встроенный синтаксический валидатор).

6. Идея, метод, алгоритм

Осуществить проход по заданному тексту, работая с каждой строкой отдельно. Разбить каждую строку на части по делиметрам (пробел, таб и перевод строки). Удалить с начала и с конца все знаки из char-bag. Затем из получившихся слов собрать новую строку.

7. Сценарий выполнения работы

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 transform (bad_chars string)
  (loop for word in (word-list string)
        collect (string-right-trim bad_chars (string-left-trim
                                     bad_chars word))))

(defun make-string-from-list (list)
  (string-right-trim '(#\Space) (format nil "~{~A ~}" list)))

(defun get-words-lists (bad_chars text)
  (loop for string in text
        collect (transform bad_chars string)))
```

```
(defun text-trim (bad_chars text)
  (loop for list in (get-words-lists bad_chars text)
    collect (make-string-from-list list)))
```

8.2. Результаты работы

; Пробелы между " и строкой связаны с latex. Результат их не содержит.

```
(print (text-trim '(#\, #\, Б#\ б#\)
  '(" Блажен, кто смолodu был молод, "
    " Блажен, кто вовремя созрел. ")))
(" лажен кто смолodu ыл молод " " лажен кто вовремя созрел ")
```

```
(print (text-trim '()
  '(" Блажен, кто смолodu был молод, "
    " Блажен, кто вовремя созрел. ")))
(" Блажен, кто смолodu был молод, " " Блажен, кто вовремя созрел. ")
```

```
(print (text-trim '(#\, #\, Б#\ б#\)
  '(" Блажен, кто смолodu был молод, "))))
(" лажен кто смолodu ыл молод ")
```

```
(print (text-trim '(#\, #\, Б#\ б#\)
  '(" Блажен, кто смолodu был молод, "
    " Блажен, кто вовремя созрел. "
    " Блажен, кто сделал 4 ЛР. ")))
(" лажен кто смолodu ыл молод " " лажен кто вовремя созрел " " лажен
кто сделал 4 ЛР ")
```

```
(print (text-trim T'(#\ т#\ е#\ к#\ с#\)
  '(" Текст TTTTT Текст ")))
(" ")
```

9. Дневник отладки

10. Замечания автора по существу работы

Замечаний нет.

11. Выводы

Благодаря данной лабораторной работе я научился работать со строками при помощи функций обработки строк и общих функций работы со списками. Также при выполнении работы помогла функция разбиения строки на слова (<http://lisp.ystok.ru/fp> из раздела "Знаки и строки" → "Слова, предложения и тексты").