

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

Студент группы М8О-307 МАИ *Безлуцкая Елизавета*, №2 по списку

Контакты: lizabeklutskaya@gmail.com

Работа выполнена: 23.05.2019

Преподаватель: Иванов Дмитрий Анатольевич, доц. каф. 806

Отчет сдан:

Итоговая оценка:

Подпись преподавателя:

Тема работы

Знаки и строки.

Цель работы

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

Задание (вариант №36)

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

Если в тексте нет малых букв, функция должна вернуть этот текст без изменения. В противном случае функция должна вернуть копию текста, в котором после всех слов, состоящих из малых букв, вставлен знак пунктуации , (запятая).

Функция должна работать как для малых латинских, так и малых русских букв.

Оборудование студента

Процессор Intel Core i5-3230M 4 @ 2.6GHz, память: 16Gb, разрядность системы: 64.

Программное обеспечение

Ubuntu 16.04 LTS, clisp compiler

Идея, метод, алгоритм

Основная функция `text-processing` вызывает вспомогательную функцию `sentence-processing`, которая обрабатывает каждое предложение текста по отдельности.

Функция `sentence-processing` работает следующим образом:

- если встретилось слово, в нём проверяется наличие знака нижнего регистра
 - встретился хоть один знак нижнего регистра – в возвращаемую коллекцию добавляется данное слово плюс знак запятой после слова
 - иначе – слово без изменений
- если встретились знаки-разделители (один или более) – в возвращаемую коллекцию добавляется вся последовательность разделителей

Результат обработки предложения должен быть строкой, а функция `sentence-processing` возвращает последовательность строк. Для преобразования «последовательность строк – одна строка» вводится функция `concat-string`.

Сценарий выполнения работы

Распечатка программы и её результаты

Исходный код

```
1 (defun whitespace-char-p (char)
2   (member char '(#\Space #\Tab #\Newline)))
3
4 (defun russian-lower-case-p (char)
5   (position char абвгдеёжзиклмнопрстуфхцчшщъыьэюя""))
6
7 (defun text-processing (text)
8   (loop for sentence in text
9         collect (concat-string (sentence-processing sentence))))
10
11 (defun sentence-processing (string)
12   (loop with len = (length string)
13         for left = 0 then (or (position-if-not #'whitespace-char-p string :
14 start right) len)
15         for right = (or (position-if #'whitespace-char-p string :start left)
16 len)
17         if (or (some #'lower-case-p (subseq string left right))
18               (some #'russian-lower-case-p (subseq string left right)))
19         collect (concatenate 'string (subseq string left right) ","))
```

```

19         if (not (or (some #'lower-case-p (subseq string left right))
20                     (some #'russian-lower-case-p (subseq string left right))))
21             collect (subseq string left right)
22
23         collect (subseq string right
24                 (or (position-if-not #'whitespace-char-p string :start right) len)
25                 )
26         while (< right len)))
27
28 (defun concat-string (list)
29   "A non-recursive function that concatenates a list of strings."
30   (if (listp list)
31       (let ((result ""))
32         (dolist (item list)
33           (if (stringp item)
34               (setq result (concatenate 'string result item))))
35       result)))
36   result)))

```

Результаты работы

```

1 (setq txt A (" СУДЬИ КТо? - За древностию лет"К
2
3 " свободной жизни их вражда непримирима,Сужденья
4 черпают из забытых газетВремён
5 Очаковских и покоренья Крыма;"))
6
7 (print (text-processing txt))А
8
9 (" СУДЬИ КТо?, - За, древностию, лет,"
10 К" свободной, жизни, их, вражда, непримирима,,Сужденья
11 , черпают, из, забытых, газет,Времён
12 , Очаковских, и, покоренья, Крыма;,")
13
14
15 (setq txt ГДЕ'("? УКАЖИТЕ НАМ, ОТЕЧЕСТВА ОТЦЫ,"КОТОРЫХ
16
17 " МЫ ДОЛЖНЫ ПРИНЯТЬ ЗА ОБРАЗЦЫ?"))
18
19 (print (text-processing txt))ГДЕ
20
21 ("? УКАЖИТЕ НАМ, ОТЕЧЕСТВА ОТЦЫ," КОТОРЫХ" МЫ ДОЛЖНЫ ПРИНЯТЬ
    ЗА ОБРАЗЦЫ?")

```

Дневник отладки

Дата	Событие	Действие по исправлению	Примечание
22.05.19	Излишняя реализация функции для проверки слова на наличие знаков нижнего регистра	Использован предикат <code>some</code>	
28.05.19	Функция <code>lower-case-p</code> гарантированно работает только для латинских букв	Добавлена функция <code>russian-lower-case-p</code> для корректной работы с русскими буквами	

Замечания автора по существу работы

Данная лабораторная работа далась мне с трудом, так как я до сих пор неосознанно переключаюсь на язык C++.

Выводы

При выполнении работы пришлось написать много излишнего кода: чтобы посмотреть, как работает множество новых для меня конструкций, исправить не изящно написанные и неработающие функции. В очередной раз нахожу язык Common Lisp неудобным для меня, порой даже сложным. Возникает вопрос, обосновано ли данное неудобство. Пока что определенного ответа у меня нет.