

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

Студент группы М8О-307-19 МАИ *Ежов Никита Павлович*, №9 по списку
Контакты: nikita.ejov2012@yandex.ru
Работа выполнена: 25.05.2022

Преподаватель: Иванов Дмитрий Анатольевич, доц. каф. 806
Отчет сдан:
Итоговая оценка:
Подпись преподавателя:

1. Тема работы

Последовательности, массивы и управляющие конструкции Common Lisp.

2. Цель работы

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

3. Задание (вариант № 4.13)

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

Функция должна возвращать копию исходного предложения, в которой из каждого слова удалена первая литера. Полученные пустые слова отбрасываются и оставшиеся непустые слова должны разделяться лишь одним пробелом.

(trim-first-char "это ушла срезка с плугом") => "то шла резка лугом"

4. Оборудование студента

Процессор Intel i7-4770 (8) @ 3.9GHz, память: 16 Gb, разрядность системы: 64.

5. Программное обеспечение

ОС Kubuntu 20.04.4 LTS, компилятор GNU CLISP 2.49.92, текстовый редактор Visual Studio Code 1.67.1

6. Идея, метод, алгоритм

Мы разбиваем строку на слова, при этом удаляя из этих слов первый символ. Потом обратно склеиваем слова в конечную строку и возвращаем её.

7. Сценарий выполнения работы

8. Распечатка программы и её результаты

8.1. Исходный код

```
;4.13

; Returns string copy
(defun copy-string (str)
  (let* ((n (length str))
        (res (make-sequence 'string n :initial-element #\.)))
    (loop for i from 0 to (- n 1)
          do (setf (aref res i) (aref str i)))
    res))

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

(defun word-list (stringArg)
  (let* ((str (copy-string stringArg)))
    ;; Разбить строки на слова, разделённые знаками whitespace
    ;; A la (split-seq-if #'whitespace-char-p string)
    (loop with len = (length str)
          for left = 0 then (1+ right)
          for right = (or (position-if #'whitespace-char-p str
                                       :start left)
                          len)
          unless (= right left) ; из всех слов исключить первый
            СИМВОЛ
            collect (subseq str (+ left 1) right)
            while (< right len))))

(defun trim-first-char (stringArg)
  (let* (
    (words (word-list stringArg))
    (res (make-sequence 'string 0)))
```

```

(loop for i from 0 to (- (length words) 2)
  do (if (< 0 (length (elt words i)))
    (setq res (concatenate 'string res (elt words i)
      (string #\Space)))))
(if (< 0 (length words))
  (setq res (concatenate 'string res (elt words (- (length
words) 1)))))
res
))

```

8.2. Результаты работы

[1]> (trim-first-char «это ушла срезка с плугом»)
«то шла резка лугом»

9. Дневник отладки

Дата	Событие	Действие по исправлению	Примечание
------	---------	-------------------------	------------

10. Замечания автора по существу работы

Сложность полученного решения $O(n)$. Пространственная сложность тоже $O(n)$, поэтому добиться лучшей асимптотики нельзя.

11. Выводы

Я познакомился со строками и литерами в Common Lisp. Очень приятно использовать высокоуровневые функции, например `subseq`, встроенные в язык. Чем-то напоминает Python, в котором так же легко работать со строками