

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

Студент группы 8О-307 МАИ *Ефимов Александр*, №7 по списку  
Контакты: `aleks.efimov2011@yandex.ru`  
Работа выполнена: 02.04.2021

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: №4.34

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

Если в тексте нет знака \*, то функция должна вернуть этот текст без изменения. В противном случае функция должна вернуть копию текста, в котором все малые буквы, предшествующие первому вхождению \*, заменены на цифру 3.

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

## 4. Оборудование студента

Процессор Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz, память: 7.6Gi, разрядность системы: 64.

## 5. Программное обеспечение

ОС Arch Linux, система CLisp.

## 6. Идея, метод, алгоритм

Сначала реализовать функцию, которая ищет в строки позицию первого вхождения \*. Если она не найдена, вернуть строку. Иначе, разбить строку на две подстроки до и с \* в начале. После замены в первой строки всех букв нижнего регистра на З вернуть объединение этой подстроки со второй (где \*).

Обобщить эту функцию для текста: найти предложение, где содержится \* и сделать объединение трех списков

1. Список предложений до \*, где все буквы нижнего регистра заменены на З;
2. Список с предложением с \*, обработанным ранее описанной функцией;
3. Список со всеми оставшимися предложениями.

Предложение с \* необходимо обернуть в список для корректной работы функции *concatenate*.

## 7. Сценарий выполнения работы

## 8. Распечатка программы и её результаты

### 8.1. Исходный код

---

```
1 (defun check-lowercase (letter)
2   "Checks if letter is lowercase"
3   (cond ((position letter "абвгдеёжзийклмнопрстуфхцчщзьёя") T)
4         ((lower-case-p letter) T)
5         (T NIL)))
6
7 (defun edit-lower (letter)
8   "If letter is lowercase, returns З (the letter otherwise)"
9   (if (check-lowercase letter)
10      #\З
11      letter))
12
13 (defun edit-string (str)
14   "Replaces lower case letters before * with З"
15   (let ((star-pos (position #\* str)))
16     (if star-pos
17         (concatenate
18          'string
19          (map 'string #'edit-lower (subseq str 0 star-pos))
```

```

20         (subseq str star-pos))
21     str)))
22
23
24 (defun edit-full-string (str)
25   "Same as edit-string but for the whole string"
26   (map 'string #'edit-lower str))
27
28 (defun edit-text (text)
29   "Replaces lower case letters in strings before * with 3"
30   (let ((star-line (position-if (lambda (str) (position #\* str)) text)))
31     (if star-line
32       (concatenate
33         'list
34         (map 'list #'edit-full-string (subseq text 0 star-line))
35         (list (edit-string (nth star-line text)))
36         (subseq text (+ 1 star-line)))
37     text)))

```

---

## 8.2. Результаты работы

---

```

[1]> (load "editor.lisp")
;; Loading file editor.lisp ...
;; Loaded file editor.lisp
[2]> (edit-text '())
NIL
[3]> (edit-text '("This Is A SINGLE sentence"))
("This Is A SINGLE sentence")
[4]> (edit-text '("This Is A SINGLE sentence with a * star"))
("T333 I3 A SINGLE 33333333 3333 3 * star")
[5]> (edit-text '("Now" "We" "Are" "Not" "Alone"))
("Now" "We" "Are" "Not" "Alone")
[6]> (edit-text '("Now" "We" "Are" "Not" "Alone" "But * starred"))
("N33" "W3" "A33" "N33" "A3333" "B33 * starred")
[7]> (edit-text '("And the Star" "CAN appear" "in r*ndom places"))
("A33 333 S333" "CAN 333333" "33 3*ndom places")

```

---

## 9. Дневник отладки

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

## 10. Замечания автора по существу работы

## 11. Выводы

- Манипуляции над строками облегчаются за счет применения функций *map* и *position*;
- Строки можно разбивать на подстроки и работать с ними без ограничения общности.