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

Студент группы 8О-306 Плешков Алексей, № по списку 22.

Контакты: pleshkov911@yandex.ru
Работа выполнена: 12.06.2022
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 № 4.41)

Запрограммировать на языке Коммон Лисп функцию, принимающую один аргумент - текст. Функция должна возвращать два значения:

1. длину самого длинного слова, состоящего только из цифр,
2. само это слово.

Если таких слов несколько, функция должна вернуть последнее из них.

```
(max-digital-word-length '("Один 1 одиннадцать 11 пятнадцать 15"))
 => 2, "15"
```

4. Оборудование студента

Ноутбук HP, процессор Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz   1.80 GHz, память 8ГБ, 64-разрядная система.

5. Программное обеспечение

ОС Windows 10, программа LispWorks Personal Edition 6.1.1

6. Идея, метод, алгоритм

Разобьем текст на предложения, а предложения на слова. Для каждого слова проверим, состоит ли оно полностью из цифр. Если нет, переходим к следующему слову, если да, проверяем его длину, если она больше или равна текущей максимальной, то запоминаем позицию этого слова. В конце возвращаем из функции длину слова, состоящего только из цифр и само это слово.

7. Сценарий выполнения работы

8. Распечатка программы и её результаты

Программа

```
(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 max-digital-word-length (text)
    (let ((maxlen 0)
          (pos -1)
          (str ""))
            (dolist (sentense text)
                (let ((list-words (word-list sentense)))
                     (loop for i from 0 to (- (length list-words) 1)
                           do (if (every #'digit-char-p (nth i list-words))
                                  (if (>= (length (nth i list-words))
maxlen) (setq maxlen (length (nth i list-words))

pos i

str (nth i list-words))))))
                (if (/= pos -1) (values maxlen str)))))
```

Результаты

CL-USER 1 > (max-digital-word-length '("Один 1 одиннадцать 11 пятнадцать 15"))

2

"15"

CL-USER 2 > (max-digital-word-length '("Один одиннадцать пятнадцать"))

NIL

CL-USER 3 > (max-digital-word-length '("Сто 100 одиннадцать 11 пятнадцать 15"))

3

"100"

CL-USER 4 > (max-digital-word-length '("Один два три 123456789 четыре 1543 "))

9

"123456789"

9. Дневник отладки

№	Дата, время	Событие	Действие по исправлению	Примечание
1				

10. Замечания автора по существу работы

Задача не вызвала затруднений и показалась довольно интересной. Со строками работать понравилось больше, чем с массивами.

11. Выводы

В третьей лабораторной работе по курсу «Функциональное Программирование» я познакомился с символами и строками. Благодаря тому, что тип данных string является подтипом vector, то к строкам применимы и все функции работы с векторами и последовательностями, а это заметно облегчает работу с данным типом данных, поскольку vector мы уже изучали ранее.