

Отчет по лабораторной работе №2 по курсу «Функциональное программирование»

Студент группы 8О-306 Кондратьев Е.А., № по списку 14.

Контакты: egor.kondratev27@gmail.com

Работа выполнена: 22.03.2022

Преподаватель: Иванов Дмитрий Анатольевич, доц. каф. 806

Отчет сдан:

Итоговая оценка:

Подпись преподавателя:

1. Тема работы

Простейшие функции работы со списками Common Lisp.

2. Цель работы

Научиться конструировать списки, находить элемент в списке, использовать схему линейной и древовидной рекурсии для обхода и реконструкции плоских списков и деревьев.

3. Задание (вариант № 2.10)

Дан список действительных чисел $(x_1 \dots x_n)$, $n \geq 3$.

Запрограммируйте рекурсивно на языке Common Lisp функцию, вычисляющую выражение вида:

$$(x_1 + x_2 + 2 * x_n) * (x_2 + x_3 + 2 * x_{n-1}) * \dots * (x_{n-1} + x_n + 2 * x_2)$$

Всего $n-1$ сомножителей.

4. Оборудование студента

Ноутбук Asus ROG GL752VW, процессор QuadCore Intel Core i7-6700HQ, 3100 MHz , память 24474 МБ (DDR4 SDRAM), 64-разрядная система.

5. Программное обеспечение

ОС Windows 10, программа VSC

6. Идея, метод, алгоритм

Идея определить вспомогательную рекурсивную функцию *productsum3help*, которая будет принимать три аргумента: исходный список и два индекса, по которым на очередном рекурсивном вызове можно будет вычислить значение очередного сомножителя.

7. Сценарий выполнения работы

8. Распечатка программы и ее результаты

Программа

```
;;; lab2 Egor Kondratev

;;; 2.10

(defun productsum3help (a i j)
  (if (> j 1) (* (productsum3help a (+ i 1) (- j 1))
    (+ (nth i a) (nth (+ i 1) a) (* 2 (nth j a))))
    (+ (nth i a) (nth (+ i 1) a) (* 2 (nth j a))))
  )

(defun product-sum3 (a)
  (productsum3help a 0 (- (length a) 1))
)
```

Результаты

```
* (product-sum3 '(1 2 3 4 5))
28561
* (product-sum3 '(1 2 3))
81
* (product-sum3 '(32 23 32))
12019
* (product-sum3 '(1 1 1 1 1 1 1))
4096
* (product-sum3 '(2 2 2 2 2 2))
32768
```

9. Дневник отладки

№	Дата, время	Событие	Действие по исправлению	Примечание
1				

10. Замечания автора по существу работы

Замечаний и добавлений нет, очень простая и тривиальная задача, к сожалению, не успел взять сложность 3. также это можно было решить с помощью двух списков из первого брать 1 и 2 элемент, а из второго последний или развернуть его и брать первый.

11. Выводы

В данной лабораторной работе я познакомился с основными функциями обработки списков в языке Common Lisp и написал свою рекурсивную реализацию вычислений.