

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

Студент группы 8О-307Б-18 МАИ *Токарев Никита*, №21 по списку  
Контакты: tokarevnikita08@mail.ru  
Работа выполнена: 21.03.2021

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

Изучить простейшие функции работы со списками common Lisp.

## 3. Задание (вариант № 2.9)

Дан список действительных чисел  $(x_1 \dots x_n)$ ,  $n \geq 2$ . Запрограммируйте рекурсивно на языке common Lisp функцию, вычисляющую выражение вида:

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

## 4. Оборудование студента

Процессор Intel® Core™ i3-5005U CPU @ 2.00GHz × 4, память: 3,8 Gb, разрядность системы: 64.

## 5. Программное обеспечение

UBUNTU 18.04.5 LTS, компилятор sbcl

## 6. Идея, метод, алгоритм

Идея в том, чтобы посчитать произведение  $n$ -го элемента списка `list` и реверсированного списка `list` и просуммировать полученные значения, используя рекурсию. В данной программе реализованы две ключевые функции (`sum-product1 list1 list2`) и (`sum-product2 list`).

- (sum-product2 list) - вызов функции (sum-product1 list1 list2), где list = list1, list2 = (reverse list), при это длина входного списка не менее 2 и элементы списка действительные числа.
- (sum-product1 list1 list2) - рекурсивная функция: произведение n-х элементов списков list1 и list2 суммируется с результатом данной вызванной функции, где входными данными являются списки: list1 = list1(n + 1,...), list2 = list2(n + 1,...).

## 7. Сценарий выполнения работы

- Анализ возможных реализаций поставленной задачи на common Lisp
- Изучение синтаксиса и основных функций работы со списками common Lisp
- Реализация поставленной задачи на common Lisp

## 8. Распечатка программы и её результаты

### 8.1. Исходный код

```
(defun sum-product1 (list1 list2)
  (cond
    ((null list1) 0)
    (t (+ (* (first list1)(first list2))
          (sum-product1 (rest list1)(rest list2)))))
  )
)
```

```
(defun sum-product2 (list)
  (if (>= (list-length list) 2)
      (sum-product1 list (reverse list))
      )
)
```

### 8.2. Результаты работы

```
* (sum-product2 '(1 2))

4
* (sum-product2 '(1 2 3 4 5))

35
* (sum-product2 '(1 -2 1 -5 7))

35
* (sum-product2 '(1))
```

NIL

```
* (sum-product2 '(1 2 1 2 1 2 1 2 1 2))
```

20

```
* (sum-product2 '())
```

NIL

## 9. Дневник отладки

Дата	Событие	Действие по исправлению	Примечание
21.03.2021	Ошибка в sum-product2: The value NIL is not of type NUMBER.	Изменил условие: ((null list1) NIL) в sum- product2 на ((null list1) 0) .	

## 10. Замечания автора по существу работы

Замечаний нет.

## 11. Выводы

В ходе данной работы я познакомился с представлением и основными особенностями списков в common Lisp. Список в common Lisp представляет собой S-выражение вида: атом | список(хвост). Также хотелось бы отметить: в (sum-product1 list1 list2) реализована хвостовая рекурсия.