# Отчет по лабораторной работе №1 по курсу «Функциональное программирование»

Студент группы 8О-306 Семенов Илья, № по списку 20.

Контакты: ilya.semenov89099@yandex.ru  
Работа выполнена: 26.02.2021  
Преподаватель: Иванов Дмитрий Анатольевич, доц. каф. 806  
Отчет сдан:   
Итоговая оценка:   
Подпись преподавателя:

### 1. Тема работы

Простейшие функции работы со списками Коммон Лисп

### 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 № 2.43)

Запрограммировать рекурсивно на языке Коммон Лисп функцию, принимающую в качестве аргумента список целых чисел . Функция должна вернуть список , где чётные элементы исходного списка, взятые в порядке следования, нечетные элементы исходного списка, взятые в порядке следования,  
 т.е. результирующий список должен содержать только первые k пар.

### 4. Оборудование студента

Ноутбук Huawei matebook d 14, процессор Intel® Core™ i5-10210U CPU 1.60GHz, память 8ГБ, 64-разрядная система.

### 5. Программное обеспечение

ОС Windows 10, программа LispWorks Personal Edition 6.1.1

### 6. Идея, метод, алгоритм

Обработка исходного списка с помощью рекурсии и функций работы со списками

### 7. Сценарий выполнения работы

Сначала создать списки из нечетных и четных элементов исходного списка, потом соединить их в один список.

### 8. Распечатка программы и её результаты

**Программа**

(defun rearrange (input-list)

(multiple-value-bind (odd-list even-list) (rearrange-impl '() '() input-list)

(combine-lists odd-list even-list '())

)

)

(defun rearrange-impl (odd-list even-list input-list)

(if (null input-list)

(values (reverse odd-list) (reverse even-list))

(if (oddp (first input-list))

(rearrange-impl (cons (first input-list) odd-list) even-list (rest input-list))

(rearrange-impl odd-list (cons (first input-list) even-list) (rest input-list))

)

)

)

(defun combine-lists (odd-list even-list result)

(if (or (null odd-list) (null even-list))

result

(combine-lists (rest odd-list)

(rest even-list)

(append result (list (first even-list) (first odd-list)))

)

)

)

**Результаты**

CL-USER 1 > (rearrange '(1 10 4 13 7))

(10 1 4 13)

CL-USER 2 > (rearrange '(1 3 2 5 6 2 8 8 0))

(2 1 6 3 2 5)

### 9. Дневник отладки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Дата, время | Событие | Действие по исправлению | Примечание |
| 1 |  |  |  |  |

### 10. Замечания автора по существу работы

### 11. Выводы

Благодаря данной работе я научился обрабатывать списки с помощью средств языка Лисп. С помощью рекурсии это делается гораздо проще, нежели без последней.