

# Отчет по лабораторной работе № 3 по курсу «Функциональное программирование»

Студент группы 8О-307 МАИ *Бердикин Тимофей*, №2 по списку

Контакты: timofey.1234@mail.ru

Работа выполнена: 12.05.2020

Преподаватель: Иванов Дмитрий Анатольевич, доц. каф. 806

Отчет сдан:

Итоговая оценка:

Подпись преподавателя:

## 1. Тема работы

Последовательности, массивы и управляющие конструкции Коммон Лисп.

## 2. Цель работы

Научиться создавать векторы и массивы для представления матриц, освоить общие функции работы с последовательностями, инструкции цикла и нелокального выхода.

## 3. Задание (вариант №19)

Запрограммировать на языке Коммон Лисп функцию, принимающую в качестве аргумента двумерный массив, представляющий действительную матрицу произвольного размера.

Функция должна возвращать новую матрицу того же размера, получающуюся из данной перестановкой столбцов - первого с последним, второго с предпоследним и т.д.

Исходный массив должен оставаться неизменным.

## 4. Оборудование ПЭВМ студента

Ноутбук Asus ROG Strix, Intel® Core™ i7-7700HQ CPU @ 2.80GHz × 8, память: 11,6Gb, разрядность системы: 64.

## 5. Программное обеспечение ЭВМ студента

OS Linux Ubuntu 20.10, LispWorks.

## 6. Идея, метод, алгоритм

Идея проста – создать внутри функции новую матрицу, потом скопировать в нее элементы исходной матрицы, учитывая перестановку столбцов, после чего вернуть скопированную матрицу.

## 7. Сценарий выполнения работы

Создадим пустую результирующую матрицу нужного размера и переменные, соответствующие измерениям матрицы  $(n, m)$ .

Обойдем исходную матрицу и скопируем каждый элемент с индексами  $(i, j)$  в элемент результирующей матрицы с индексами  $(i, m - j - 1)$ .

## 8. Распечатка программы и её результаты

### 8.1. Исходный код

```
(defun swap-columns(matrix)
  (let ((n (first (array-dimensions matrix)))
        (m (second (array-dimensions matrix)))
        (result (make-array (array-dimensions matrix))))
    (loop for i upfrom 0 to (- n 1)
          do
            (loop for j upfrom 0 to (- m 1)
                  do
                    (setf (aref result i j) (aref matrix i (- m j 1))))
            )
          )
    result)
)
```

```
(print (swap-columns #2A((1 2 3)(4 5 6))))
```

### 8.2. Результаты работы

```
CL-USER 2 >
```

```
(swap-columns #2A((1 2 3)(4 5 6)))
```

```
#2A((3 2 1) (6 5 4))
```

```
CL-USER 3 > (swap-columns #2A((1 2 3 4)(5 6 7 8)(9 10 11 12)))
```

#2A((4 3 2 1) (8 7 6 5) (12 11 10 9))

## 9. Дневник отладки

Дата	Событие	Действие по исправлению	Примечание
11.05	Undefined operator SWAP-COLUNS	columns	Опечатка

## 10. Замечания автора по существу работы

Эта работа далась мне куда легче, чем предыдущая, за счёт того, что я уже более глубоко познакомился с языком Коммон Лисп и понял, что от меня требуется в целом.

## 11. Выводы

В данной работе я научился использовать циклы, а так же функции для работы с последовательностями и массивами в языке Коммон Лисп.