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

Студент группы 8О-308 Иванов Федор, № по списку 5.

Контакты: kenola82007@gmail.com  
Работа выполнена: 02.06.2022

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 №** 3.26**)**

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

Функция должна возвращать новую матрицу B того же размера, каждый элемент которой bij равен наибольшему из элементов матрицы A, расположенных в области, определяемой индексами i и j и заштрихованной на рисунке (строка i и столбец j принадлежат области).

### **4. Оборудование студента**

Ноутбук HP-Probook 440G5, Intel Core i5-8250U: 1,6 ГГц, до 3,4 ГГц при использовании технологии Intel Turbo Boost 2.0, 4 ядра, разрядность системы: 64

### **5. Программное обеспечение**

ОС Linux Mint, программа LispWorks Personal Edition 6.1.1

### **6. Идея, метод, алгоритм**

Идея заключалась в следующем: делаем двойной цикл по матрице, начиная от левого нижнего края, проверяя два условия, является ли элемент ниже или левее больше чем текущий, если так, делаем замену, продолжаем пока не окажемся в правом верхнем углу.

### **7. Сценарий выполнения работы**

### **8. Распечатка программы и её результаты**

**Программа**

(defun copy-array (arr)

(let\* ((dimensions (array-dimensions arr)) (new-arr (make-array dimensions)))

(dotimes (i (array-total-size arr))

(setf (row-major-aref new-arr i)

(row-major-aref arr i)))

new-arr))

(defun fun (mat)

(let ((size (array-dimensions mat)))

(let ((lines (first size)) (columns (second size)) (ans (copy-array mat)))

(do ((i (- lines 1) (- i 1))) ((< i 0) ans)

(do ((j 0 (+ j 1))) ((>= j columns) 'done\_str)

(if (and (> j 0) (> (aref ans i (- j 1)) (aref ans i j)))

(setf (aref ans i j) (aref ans i (- j 1))))

(if (and (< i (- lines 1)) (> (aref ans (+ i 1) j) (aref ans i j)))

(setf (aref ans i j) (aref ans (+ i 1) j))

))))))

**Результаты**

(print (fun (make-array '(3 3) :initial-contents '((0 0 0) (0 1 0) (0 0 0)))))

#2A((0 1 1) (0 1 1) (0 0 0))

(print (fun (make-array '(3 3) :initial-contents '((0 1 2) (3 4 5) (6 7 8)))))

#2A((6 7 8) (6 7 8) (6 7 8))

(print (fun (make-array '(3 3) :initial-contents '((3 4 5) (2 3 4) (1 2 3)))))

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

### **9. Дневник отладки**

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

### **10. Замечания автора по существу работы**

Написал алгоритм на листочке, чтобы убедиться в корректности алгоритма. Долго пришлось повозиться, чтоб разобраться с матрицами и как они работают. Наблюдал аналогию с ЯП Си в создании циклов.

### **11. Выводы**

В процессе выполнения лабораторной работы, я научился создавать векторы и массивы для представления матриц, освоить общие функции работы с последовательностями, инструкции цикла и нелокального выхода.