

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

Студент группы 8О-308 Петросян Виктор, № по списку 15.

Контакты: viko20000@mail.ru

Работа выполнена: 05.05.2020

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 работать с многомерными массивами, изучить способы навигации и изменения значений в его ячейках.

3. Задание (вариант № 3.29)

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

Функция должна возвращать новую матрицу B того же размера, полученную транспонированием A , т.е. по правилу:

- каждую строку с номером i сделать столбцом с номером i ,
- каждую столбец с номером i сделать строкой с номером i ,
 $i = 1, \dots, n$, где n - порядок матрицы A .

Исходный массив A должен оставаться неизменным.

4. Оборудование студента

Ноутбук Samsung NP300E5c-S0URU 15.6, процессор Intel® Core™ i5-2410M CPU 2.30GHz, память 6ГБ, 64-разрядная система.

5. Программное обеспечение

Linux 4.15.0-96-generic #97-Ubuntu SMP, использовал sbcl(для запуска .lisp файлов) + VSCode(для редактирования кода)

6. Идея, метод, алгоритм

Освоив несколько функций для работы с массивами, а именно создание массива make-array, изменение значения по индексу при помощи комбинации setf + aref я просто реализовал всем известный из Линейной алгебры алгоритм транспонирования матрицы. Функцию copy-array нашел в интернете. Не смог найти встроенный аналог в стандартной библиотеке.

7. Сценарий выполнения работы

8. Распечатка программы и её результаты

Программа

```
(defun copy-array (array)

  (let ((dims (array-dimensions array)))

    (adjust-array (make-array dims :displaced-to array) dims)))

(defun transpose (matrix)
(let* ((n (array-dimension matrix 0))
      (result (copy-array matrix))
      (i 0) (j 0))
  (loop while (< i n)
    do
      (setq j 0)
      (loop while (< j n)
        do
          (setf (aref result i j) (aref matrix j i))
          (setq j (+ j 1)))
        (setq i (+ i 1)))
    (return-from transpose result)))

(defun execute (nums)
  (print nums)
  (print (transpose nums))
  (print nums)
  (print "-----"))

(defvar first-test (make-array '(3 3) :initial-contents '((1 2 3)(4 5 6)(7 8 9))))
(defvar second-test (make-array '(4 4) :initial-contents '((100 5 6 9)(1 2 3 8)(5 10 12 1)(1 5 13 10))))
(defvar third-test (make-array '(1 1) :initial-contents '((31415))))
(execute first-test)
(execute second-test)
(execute third-test)
```

Результаты

```
#2A((1 2 3) (4 5 6) (7 8 9))
#2A((1 4 7) (2 5 8) (3 6 9))
#2A((1 2 3) (4 5 6) (7 8 9))
"-----"
#2A((100 5 6 9) (1 2 3 8) (5 10 12 1) (1 5 13 10))
#2A((100 1 5 1) (5 2 10 5) (6 3 12 13) (9 8 1 10))
#2A((100 5 6 9) (1 2 3 8) (5 10 12 1) (1 5 13 10))
"-----"
#2A((31415))
#2A((31415))
#2A((31415))
"-----"
```

9. Дневник отладки

№	Дата, время	Событие	Действие по исправлению	Примечание
1				

10. Замечания автора по существу работы

11. Выводы

В данной лабораторной работе я познакомился с двумерными массивами, изучил простейшие способы их объявления, а также навигации и изменения значения в конкретной ячейке. Написал программу, которая принимает в качестве параметра двумерный массив, представляющий действительную квадратную матрицу и возвращает новую матрицу того же размера что и исходная, полученную транспонированием. Программа работает правильно и прошла все тесты.