

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

Студент группы М8О-307 МАИ *Днепров Иван*, №10 по списку
Контакты: *vanya.dneprov@gmail.com*
Работа выполнена: 26.03.2020

Преподаватель: Иванов Дмитрий Анатольевич, доц. каф. 806
Отчет сдан:
Итоговая оценка:
Подпись преподавателя:

1. Тема работы

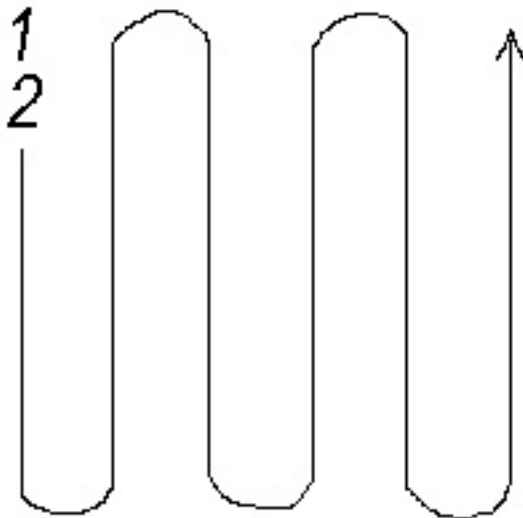
Последовательности, массивы и управляющие конструкции Common Lisp.

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 №3.42)

Запрограммировать на языке Коммон Лисп функцию, принимающую в качестве единственного аргумента целое число n - порядок матрицы. Функция должна создавать и возвращать двумерный массив, представляющий целочисленную квадратную матрицу порядка n , элементами которой являются числа $1, 2, \dots, n^2$, расположенные по схеме, показанной на рисунке.



4. Оборудование студента

MacBook (13-inch, Mid 2010), процессор 2,4 GHz Intel Core 2 Duo, память: 8Gb, разрядность системы: 64.

5. Программное обеспечение

Mac OS 10.13.6, компилятор clisp, текстовый редактор Sublime Text 3.

6. Идея, метод, алгоритм

Матрицу я заполняю не по сторокам. На каждом шаге я заполняю нечётный и чётный столбцы, причём, нечётные столбцы заполняю снизу вверх, а чётные – наоборот. При наполнении каждого столбца я увеличиваю счётчик на единицу и если в конце счётчик – нечётный, то последний столбец не заполняем, потом что его нет.

7. Сценарий выполнения работы

8. Распечатка программы и её результаты

8.1. Исходный код

```
(defun matr (n m val)
  (let ((matr nil) (line nil))
    (dotimes (i m t) (push val line))
    (dotimes (i n matr) (push line matr))))

(defun matrix-tl-br (n)
  (if < n 0
      ())
  (let ((matrix (make-array (list n n)))
        (cnt (ceiling n 2))
        (num 1))
    (dotimes (i cnt)
      (loop for j from 0 upto (1- n)
        do (setf (aref matrix j (* i 2)) num)
            (setf num (1+ num))
      )
    )
  )
```

```

; условие, чтобы корректно обрабатывались матрицы с нечетным размером
(when (or (evenp n) (< i (1- cnt)))
  (loop for j from (1- n) downto 0
    do (setf (aref matrix j (1+ (* i 2))) num)
      (setf num (1+ num))
  )
)
)
matrix)
)

```

8.2. Результаты работы

```

[2]> (matrix-tl-br 5)
#2A((1 10 11 20 21)
     (2 9 12 19 22)
     (3 8 13 18 23)
     (4 7 14 17 24)
     (5 6 15 16 25))
[3]> (matrix-tl-br 1)
#2A((1))
[4]> (matrix-tl-br 10)
#2A((1 20 21 40 41 60 61 80 81 100)
     (2 19 22 39 42 59 62 79 82 99)
     (3 18 23 38 43 58 63 78 83 98)
     (4 17 24 37 44 57 64 77 84 97)
     (5 16 25 36 45 56 65 76 85 96)
     (6 15 26 35 46 55 66 75 86 95)
     (7 14 27 34 47 54 67 74 87 94)
     (8 13 28 33 48 53 68 73 88 93)
     (9 12 29 32 49 52 69 72 89 92)
     (10 11 30 31 50 51 70 71 90 91))

```

9. Дневник отладки

Дата	Событие	Действие по исправлению	Примечание
------	---------	-------------------------	------------

10. Замечания автора по существу работы

Что-то подобное я писал на первом курсе, было интересно решить похожую задачу на Lisp.

11. Выводы

Я поработал с массивами в Lisp, странно, но до этого я думал, что в лиспе есть только списки. Я понял, как в работе с массивами применять циклы. Мне интересно было выполнять это задание, и хоть оно и оказалось не сложным, мне пришлось потратить некоторое время, чтобы разобраться с работой с массивами в Lisp, чтобы его выполнить.