

# Отчет по лабораторной работе № 4 по курсу «Функциональное программирование»

Студент группы 80-306Б МАИ *Макаров Никита*, №11 по списку

Контакты: [quizbeat@gmail.com](mailto:quizbeat@gmail.com)

Работа выполнена: 16.05.2015

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

Запрограммировать на языке Common Lisp 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 ниже.

```
(defun matrix-1l-2l (n)
  ...)
```

```
(matrix-1l-2l 4) =>
#2A((1  3  4 10)
     (2  5  9 11)
     (6  8 12 15)
     (7 13 14 16))
```

## 4. Оборудование студента

Процессор Intel Core i5 2 @ 1.3GHz, память: 4Gb, разрядность системы: 64.

## 5. Программное обеспечение

ОС Mac OS X 10.9, среда Clozure CL 1.10

## 6. Идея, метод, алгоритм

Можно заметить, что числа в матрице следуют по диагоналям и направление движения меняется на противоположное от диагонали к диагонали. Также можно заметить, что направление движения зависит от длины текущей диагонали: если длина является четным числом, то движение идет вверх, иначе – вниз. Таким образом, можно построить такую матрицу в два этапа: на первом заполнить побочную диагональ и все, что выше, а на втором этапе оставшуюся часть. Заполнение в два этапа удобно из-за разных условий перехода на следующую диагональ.

## 7. Сценарий выполнения работы

## 8. Распечатка программы и её результаты

### 8.1. Исходный код

```
(defvar matrix (make-array '(1 1))) ; matrix

(defun mprint (m)
  (loop for i below (car (array-dimensions m)) do
    (loop for j below (cadr (array-dimensions m)) do
      (let ((cell (aref m i j)))
        (format t "~a " cell)))
      (format t "~%"))))

(defun matrix-1l-2l (n)
  (PROG
    ((i 0) (j 0) (value 1) (last (- n 1))) ; local vars
    (setf matrix (make-array (list n n)))
    (loop for diag-length from 1 to n do
      (if (evenp diag-length)
        (let ((end (- diag-length 1)))
          (loop for l from 0 to end do
            (setf (aref matrix i j) value)
            (incf value)
            (if (/= l end)
              (do () ((decf i) (incf j)))
            )
          )
        )
        (if (= j last)
          (incf i)
          ; else
          (incf j)
        )
      )
    )
  ; else
```

```

    (let ((end (- diag-length 1)))
      (loop for l from 0 to end do
        (setf (aref matrix i j) value)
        (incf value)
        (if (/= l end)
            (do () ((incf i)(decf j)))
          )
      )
    )
    (if (= i last)
        (incf j)
        ; else
        (incf i)
    )
  )
)
)
(loop for diag-length from last downto 1 do
  (if (evenp diag-length)
      (let ((end (- diag-length 1)))
        (loop for l from 0 to end do
          (setf (aref matrix i j) value)
          (incf value)
          (if (/= l end)
              (do () ((decf i) (incf j)))
            )
        )
      )
      (incf i)
    )
    ; else
    (let ((end (- diag-length 1)))
      (loop for l from 0 to end do
        (setf (aref matrix i j) value)
        (incf value)
        (if (/= l end)
            (do () ((incf i) (decf j)))
          )
      )
    )
    (incf j)
  )
)
)
)
matrix
)

(defun t (x) (mprint (matrix-1l-2l x))) ; fast testing

```

## 8.2. Результаты работы

```
> (mprint (matrix-1l-2l 4))
1 3 4 10
2 5 9 11
6 8 12 15
7 13 14 16
NIL
> (mprint (matrix-1l-2l 5))
1 3 4 10 11
2 5 9 12 19
6 8 13 18 20
7 14 17 21 24
15 16 22 23 25
NIL
> (mprint (matrix-1l-2l 6))
1 3 4 10 11 21
2 5 9 12 20 22
6 8 13 19 23 30
7 14 18 24 29 31
15 17 25 28 32 35
16 26 27 33 34 36
NIL
?
```

## 9. Дневник отладки

## 10. Замечания, выводы

Выполняя эту работу, я получил базовые навыки работы с массивами. В ходе изучения возникли некоторые трудности, не сразу получилось создать массив произвольного размера  $n$ . Еще для меня показался несколько громоздким синтаксис обновления значения массива, но эта проблема решается макросами. Для просмотра матрицы лучше воспользоваться дополнительной функцией печати, которая выводит двумерный массив в удобочитаемом формате.