

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

Студентка группы 8О-308 Понагайбо Анастасия, № по списку 13.

Контакты: ponagaibo@mail.ru

Работа выполнена: 10.04.2018

Преподаватель: Иванов Дмитрий Анатольевич, доц. каф. 806

Отчет сдан:

Итоговая оценка:

Подпись преподавателя:

1. Тема работы

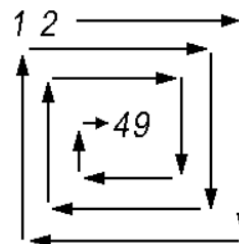
Последовательности, массивы и управляющие конструкции Коммон Лисп.

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 № 2.42)

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 спирали.



4. Оборудование студента

Ноутбук ASUS EeeBook, процессор Intel® Atom™ CPU Z3735F @ 1,33 GHz, память 2ГБ, 32-разрядная система.

5. Программное обеспечение

ОС Windows 8.1, программа CLisp в emacs.

6. Идея, метод, алгоритм

Необходимо в цикле обходить строки и столбцы матрицы и заполнять их. Сначала заполняются элементы верхней строки слева направо, после этого элементы правого столбца сверху вниз, потом элементы нижней строки справа налево и элементы левого столбца снизу вверх, не включая самый верхний элемент. Эта процедура повторяется для внутренней матрицы (без внешних строк и столбцов).

7. Сценарий выполнения работы

8. Распечатка программы и её результаты

Программа

```
(defun fill-array (n)
```

```

(let ((arr (make-array (list n n)
                       :initial-element n))
      (to (ceiling n 2))
      (m 1))
  (dotimes (k to)
    (loop for j upfrom k to (- n k 1)
          do (setf (aref arr k j) m)
              (setf m (+ 1 m)))
    (loop for i upfrom (+ k 1) to (- n k 1)
          do (setf (aref arr i (- n k 1)) m)
              (setf m (+ 1 m)))
    (loop for j downfrom (- n k 2) to k
          do (setf (aref arr (- n k 1) j) m)
              (setf m (+ 1 m)))
    (loop for i downfrom (- n k 2) to (+ k 1)
          do (setf (aref arr i k) m)
              (setf m (+ 1 m)))
  )
  arr)
)

(defun print-matrix (matrix &optional (chars 3) stream)
  (let ((*print-right-margin* (+ 6 (* (1+ chars)
                                       (array-dimension matrix 1)))))
    (pprint matrix stream)
    (values)))

(defun spiral-matrix (n)
  (print-matrix (fill-array n)))

(spiral-matrix 1)
(spiral-matrix 2)
(spiral-matrix 7)

```

Результаты

```

#2A((1))
#2A((1 2)
     (4 3))
#2A((1 2 3 4 5 6 7)
     (24 25 26 27 28 29 8)
     (23 40 41 42 43 30 9)
     (22 39 48 49 44 31 10)
     (21 38 47 46 45 32 11)
     (20 37 36 35 34 33 12)
     (19 18 17 16 15 14 13))

```

9. Дневник отладки

№	Дата, время	Событие	Действие по исправлению	Примечание
1	10.04.18	Неиспользуемая инициализация массива	Было убрано :initial-element n	

10. Замечания автора по существу работы

Для выполнения лабораторной работы потребовалось в определенном порядке циклически обходить матрицу и заполнять ее числами от 0 до n^2 . Сначала вычислялось число to , означающее количество применений заполнения ($n/2$ раз, где n – размерность матрицы). После этого числами от 0 до n^2 заполняется верхняя строка с k -го по $(n-k-1)$ -е место, где $0 < k < to$ означает, какой по счету «контур» матрицы, начиная с внешнего, сейчас заполняется. Потом заполняется правый столбец с $(k+1)$ -го по $(n-k-1)$ -е место, нижняя строка с $(n-k-2)$ -го

по k-е место и левый столбец с (n-k-2)-го по (k+1)-е место. Это обеспечивает заполнение одного «контура», после этого цикл повторяется для внутренних «контуров».

11. Выводы

При выполнении данной лабораторной работы я научилась создавать массивы для представления матриц, освоила инструкции цикла, а также написала функцию, которая создает матрицу размером n на n (где n – введенное пользователем число) и по спирали заполняет ее числами от 0 до n^2 . В программе использовались функции `make-array`, `ceiling`, конструкции `(loop for ... upfrom ... to ... do ...)`, `(loop for ... downfrom ... to ... do)`, `dotimes`. Программа работает правильно и прошла все тесты.