

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

Студент группы 8О-308Б Ивенкова Любовь, № по списку 6.

Контакты: lyubov.iven@mail.ru

Работа выполнена: 18.04.2022

Преподаватель: Иванов Дмитрий Анатольевич, доц. каф. 806

Отчет сдан:

Итоговая оценка:

Подпись преподавателя:

1. Тема работы

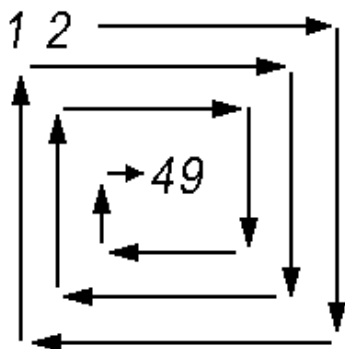
Последовательности, массивы и управляющие конструкции Коммон Лисп.

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 № 3.46)

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.



```
(defun spiral-matrix (n)
  ...)
```

```
(spiral-matrix 7) =>
#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))
```

4. Оборудование ПЭВМ студента

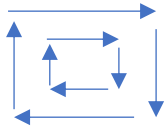
PC, процессор Intel Core i5-4460 @ 3,2 GHz (32 x 100), память 11,7 Gb, 64-разрядная система.

5. Программное обеспечение ЭВМ студента

OS Windows 10, программа LispWorks 7.1 Personal (64-bit).

6. Идея, метод, алгоритм

Разобьём матрицу на витки спирали, а каждый виток на 4 части – верхнюю, правую, нижнюю и левую:



Этот обход сохраняет требуемый порядок вывода чисел.

7. Сценарий выполнения работы

В начале создаём саму матрицу размера $n \times n$. Вычисляем количество витков, затем в цикле проходимся по каждому из них. Каждую часть витка также обходим циклом, расставляя числа.

Для задания матрицы и других локальных переменных используем `let`. В нём зададим: `curr` – текущее число, которое мы ставим в матрицу, изначально равно 1, увеличивается на единицу с каждым шагом; `lvl` – количество витков (уровней); `border` – переменная конца каждой части витка (так как в витке строки и столбцы пересекаются, то надо очень аккуратно указывать границы каждого, также эта переменная зависит от витка, на котором мы находимся).

Для удобства печати используем функцию `print-matrix`, приведённую в курсе.

8. Распечатка программы и её результаты

Программа

```
(defun spiral-matrix(n)
  (let ((matrix (make-array (list n n)))
        (lvl (+ (floor n 2) (mod n 2)))
        (curr 1)
        (border))
    (loop for i upfrom 0 to (- lvl 1)
      do
        (setq border (- n (+ i 1)))
        (loop for j upfrom i to border
          do
            (setf (aref matrix i j) curr)
            (setq curr (+ curr 1)))
        (loop for j upfrom (+ i 1) to border
          do
            (setf (aref matrix j border) curr)
            (setq curr (+ curr 1)))
        (loop for j downfrom (- border 1) to i
          do
            (setf (aref matrix border j) curr)
            (setq curr (+ curr 1)))
```

```

        (loop for j downfrom (- border 1) to (+ i 1)
              do
                (setf (aref matrix j i) curr)
                (setq curr (+ curr 1))))
      matrix))

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

```

Результаты

CL-USER 1 > (print-matrix (SPIRAL-MATRIX 3))

```

#2A((1 2 3)
     (8 9 4)
     (7 6 5))

```

CL-USER 2 > (print-matrix (SPIRAL-MATRIX 4))

```

#2A((1 2 3 4)
     (12 13 14 5)
     (11 16 15 6)
     (10 9 8 7))

```

CL-USER 3 > (print-matrix (SPIRAL-MATRIX 7))

```

#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

10. Замечания автора по существу работы

Программа проходит по всем виткам спирали, которых или $n/2$ штук, или $n/2 + 1$. К тому же внутри каждого витка она выполняет четыре прохода по его частям, длиной не превышающим n . В итоге получаем сложность $O(n^2)$.

11. Выводы

В процессе выполнения работы я познакомилась с двумерными массивами и функциями для работы с ними в языке Common Lisp. Также изучила работу цикла loop. Для удобства печати двумерного массива использовала функцию print-matrix из курса.