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

Студент группы 8О-308 Коростелев Дмитрий, № по списку 11.

Контакты: dmitry.k48@yandex.ru
Работа выполнена: 09.04.2021
Преподаватель: Иванов Дмитрий Анатольевич, доц. каф. 806
Отчет сдан:
Итоговая оценка:
Подпись преподавателя:

1. Тема работы

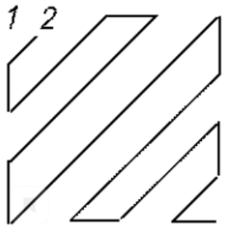
Последовательности, массивы и управляющие конструкции Коммон Лисп

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 № 3.44/3)

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, ... n^2, расположенные по схеме, показанной на рисунке.



4. Оборудование студента

Ноутбук Asus VivoBook Pro 15, процессор Intel® Core™ i7-7700HQ CPU 2.80GHz 2.81GHz, память 8ГБ, 64-разрядная система.

5. Программное обеспечение

ОС Windows 10, программа LispWorks Personal Edition 7.1.2

6. Идея, метод, алгоритм

Идея алгоритма заключается в последовательном обходе ячеек массива как показано на рисунке, т.е. как видно из рисунка, обход начинается с левого верхнего угла и далее повторяются два действия: 1) подниматься или спускаться по текущей диагонали вправо вверх или вниз до того момент, пока это будет возможно 2) сделать переход на ячейку вправо или на ячейку вниз 3) повторять первые два пункта до того момента, пока возможен переход на новую ячейку (пока не будут помечены все ячейки матрицы)

7. Сценарий выполнения работы

8. Распечатка программы и её результаты

Программа

```
(defun matrix-tl-bl (n)

     (let ((matrix (make-array  (list N N))))
     (do ((i 0 (+ i 1))
           (c 0 (+ c (if (< i n) (+ 1 i) (- (* 2 n) i 1))))
          )
          ((>= i (- (* 2 n) 1)))
          (if (< i n)
               (do ((j 0 (+ j 1)))
                    ((> j i))
                    (if (= (rem i 2) 0)
                         (setf (aref matrix (- i j) j) (+ c j 1))
                         (setf (aref matrix j (- i j)) (+ c j 1))
                    )
               )
               (do ((j 0 (+ j 1)))
                    ((>= j (- (- (* 2 n) i) 1)))
                    (if (= (rem i 2) 0)
                         (setf (aref matrix (- (- n 1) j) (+ (+ (-
i n) j) 1)) (+ c j 1))
                         (setf (aref matrix (+ (+ (- i n) j) 1) (-
(- n 1) j)) (+ c j 1))
                    )
               )
          )
```

```
      )
      matrix
      )
)
```

Результаты

```
CL-USER 1 > (matrix-tl-bl 4)

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

CL-USER 2 > (matrix-tl-bl 3)

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

CL-USER 3 > (matrix-tl-bl 2)

#2A((1 2) (3 4))

CL-USER 4 > (matrix-tl-bl 1)

#2A((1))

CL-USER 5 > (matrix-tl-bl 5)

#2A((1 2 6 7 15) (3 5 8 14 16) (4 9 13 17 22) (10 12 18 21 23) (11
19 20 24 25))
```

9. Дневник отладки

№	Дата, время	Событие	Действие по исправлению	Примечание
1	09.04.2021	Плохой стиль кода, много глобальных переменных	Полностью переписал программу в одной функции и серьезно упростил код	
2	19.05.2021	Ненужная глобальная переменная	Убрал переменную, добавил let-блок	

10. Замечания автора по существу работы

11. Выводы

Написал и протестировал функцию на языке Common Lisp, которая заполняет двумерную матрицу заданной размерности «диагональной змейкой», научился создавать массивы и освоил общие функции работы с ними, научился работать с циклами.