

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

Студент группы 8О-306 Плешков Алексей, № по списку 22.

Контакты: pleshkov911@yandex.ru

Работа выполнена: 12.06.2022

Преподаватель: Иванов Дмитрий Анатольевич, доц. каф. 806

Отчет сдан:

Итоговая оценка:

Подпись преподавателя:

1. Тема работы

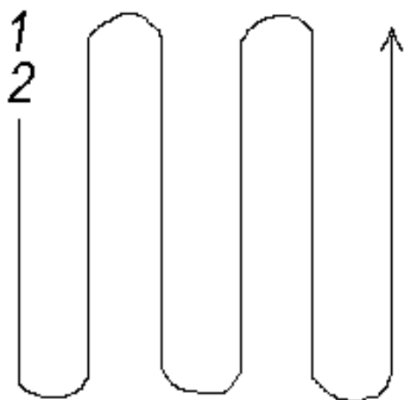
Последовательности, массивы и управляющие конструкции Common Lisp.

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

Ноутбук HP, процессор Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz 1.80 GHz, память 8ГБ, 64-разрядная система.

5. Программное обеспечение

OS Windows 10, программа LispWorks Personal Edition 6.1.1

6. Идея, метод, алгоритм

Заполняем матрицу по столбцу, чётные столбцы заполняются сверху вниз, нечётные – снизу вверх. При каждом заполнении увеличиваем счётчик на единицу для выполнения условия (элементами матрицы являются числа от 1 до квадрата исходного числа n). На каждом шаге цикла заполняются два столбца – чётный и нечётный.

7. Сценарий выполнения работы

8. Распечатка программы и её результаты

Программа

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

Результаты

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

```
#2A((1))
```

```
CL-USER 2 > (print (matrix-tl-bl 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))
```

```
CL-USER 3 > (print (matrix-tl-bl 8))
```

```
#2A((1 16 17 32 33 48 49 64)
     (2 15 18 31 34 47 50 63)
     (3 14 19 30 35 46 51 62)
     (4 13 20 29 36 45 52 61)
     (5 12 21 28 37 44 53 60)
     (6 11 22 27 38 43 54 59)
     (7 10 23 26 39 42 55 58)
     (8 9 24 25 40 41 56 57))
```

9. Дневник отладки

№	Дата, время	Событие	Действие по исправлению	Примечание
1				

10. Замечания автора по существу работы

У меня уже имелся опыт с подобным обходом матрицы, но было интересно реализовать это на языке Lisp.

11. Выводы

Я познакомился с массивами в языке Lisp, узнал, как выполнять различные операции над ними, как использовать циклы. Массивы часто используются, в них удобно хранить данные, данная лабораторная работа помогла мне с ними ознакомиться. Также в языке Lisp существует множество библиотек и функций для работы с ними. Некоторые я рассмотрел, пока выполнял работу.