

# Отчет по лабораторной работе № 1 по курсу «Функциональное программирование»

Студент группы М8О-307 МАИ *Довженко Анастасия*, №7 по списку  
Контакты: [tutkarma@gmail.com](mailto:tutkarma@gmail.com)  
Работа выполнена: 15.03.2019

Преподаватель: Иванов Дмитрий Анатольевич, доц. каф. 806  
Отчет сдан:  
Итоговая оценка:  
Подпись преподавателя:

## 1. Тема работы

Примитивные функции и особые операторы Common Lisp.

## 2. Цель работы

Научиться вводить S-выражения в Лисп-систему, определять переменные и функции, работать с условными операторами, работать с числами, используя схему линейной и древовидной рекурсии.

## 3. Задание (вариант №1.34)

Поле шахматной доски определяется парой натуральных чисел, каждое из которых не превосходит восьми:

- первое число - номер вертикали (при счете слева направо).
- второе - номер горизонтали (при счете снизу вверх).

Определите на языке Коммон Лисп функцию-предикат с четырьмя параметрами - натуральными числам  $k$ ,  $l$ ,  $m$ ,  $n$ , каждое из которых не превосходит восьми.

$k$ ,  $l$

Задают поле, на котором расположена фигура - ладья.

$m$ ,  $n$

Задают поле, куда она должен попасть.

Функция должна возвращать

$T$ ,

если ладья  $(k,l)$  может попасть на поле  $(m,n)$  за один ход;

$i, j$

два значения с помощью `values`, если ладья  $(k, l)$  может попасть на поле  $(m, n)$  за два хода через поле  $(i, j)$ .

## 4. Оборудование студента

Ноутбук Asus UX310U, процессор Intel Core i7-6500U CPU 2.50GHz x 4, память: 8Gb, разрядность системы: 64.

## 5. Программное обеспечение

ОС Ubuntu 16.04 LTS, компилятор clisp, текстовый редактор Sublime Text 3.

## 6. Идея, метод, алгоритм

Ладья может попасть из  $(k, l)$  в  $(m, n)$  за один ход, если она расположена на той же вертикали или/и горизонтале, что и конечное поле. Отсюда получаем условие `k == m or l == n`. Во всех остальных случаях ладья попадет в конечное поле через «транзитную» клетку, которую можно выбрать двумя способами: либо это будет клетка  $(m, l)$ , либо клетка  $(k, n)$ . В своем решении я выбрала второй вариант.

## 7. Сценарий выполнения работы

## 8. Распечатка программы и её результаты

### 8.1. Исходный код

```
(defun castle-moves (k l m n)
  (cond ((or (= k m)
              (= l n)) "T")
        (t (values k n))))
```

### 8.2. Результаты работы

```
(castle-moves 1 1 1 1)
"T"
(castle-moves 4 5 7 8)
4 ;
8
(castle-moves 7 7 2 4)
7 ;
4
(castle-moves 2 6 6 6)
```

"Т"

## 9. Дневник отладки

Дата	Событие	Действие по исправлению	Примечание
------	---------	-------------------------	------------

## 10. Замечания автора по существу работы

Работа показалась мне слишком простой с точки зрения программирования.

## 11. Выводы

При выполнении работы я вспомнила синтаксис языка. Мне кажется, эта работа была больше направлена на аналитическое решение (хоть и простое), нежели на программное. Основные сложности были связаны со средой разработки, потому что мне не хотелось использовать IDE, онлайн-компиляторы по непонятной мне причине не выводили все значения, возвращаемые функцией, и только из консоли это заработало более-менее приемливо.