

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

Студент группы 8О-307 МАИ *Гамов Павел*, №4 по списку  
Контакты: `pagamov@gmail.com`  
Работа выполнена: 25.03.2021

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

## 1. Тема работы

Примитивные функции и особые операторы Коммон Лисп

## 2. Цель работы

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 №1.43/4)

Даны целые числа  $h$  ( $0 \leq h \leq 12$ ) и  $m$  ( $0 \leq m \leq 60$ ), указывающие момент времени:  $h$  часов,  $m$  минут. Запрограммируйте на языке Коммон Лисп функцию с двумя параметрами  $h$  и  $m$ , вычисляющую наименьшее время, которое должно пройти до того момента, когда часовая и минутная стрелки на циферблате совпадут. Время следует вычислять как целое число минут, округлённое в меньшую сторону.

Примеры

`(parallel-hands-minutes 0 0) => 65`

`(parallel-hands-minutes 0 15) => 50`

## 4. Оборудование студента

macOS Catalina 10.15.7 Intel Core i5 2.3 GHz 8 GB RAM

## 5. Программное обеспечение

macOS, среда `vim` + `sbcl`

## 6. Идея, метод, алгоритм

Так как это мой первый опыт с данным языком, зная общие концепции я сделал прототип на Python, далее упростил его, подогнав под синтаксис. Обнаружил краевые условия, например когда оба значения равны 0, фактически они уже встретились, так что в главной функции `parallel-hands-minutes` я расписал два сценария, где вызываю дополнительную функцию `sub-f` где присутствует аккумулятор  $z$ , в котором хранится переменная для ответа.

## 7. Сценарий выполнения работы

Сделал прототип на Python. Нашел краевые условия. Создал `if` условие, откуда вызываю `sub-f` функцию. В `sub-f` функции оператором `cond` вызываю далее хвостовую рекурсивную функцию.

## 8. Распечатка программы и её результаты

### 8.1. Исходный код

```
#!/usr/local/bin/sbcl --script
(defun sub-f (x y z)
  (cond
    ((> x 11) (sub-f 0 y z))
    ((> y 59) (sub-f (+ x 1) 0 z))
```

```
((and (= x (/ y 5)) (= 0 (rem y 5))) z)
(t (sub-f x (+ y 1) (+ z 1))))))
```

```
(defun parallel-hands-minutes (x y)
  (if (= (rem (+ y 1) 60) 0)
      (sub-f (+ x 1) (+ y 1) 1)
      (sub-f x (+ y 1) 1)))
```

```
(print (parallel-hands-minutes 0 0))
(print (parallel-hands-minutes 0 15))
```

## 8.2. Результаты работы

```
65
50
```

## 9. Дневник отладки

## 10. Замечания автора по существу работы

sbcl предоставляет возможность отладки программы, надо лучше освоить все команды. Пока нет полного понимания языка, можно прототипировать на бумаге или другом языке.

## 11. Выводы

Я познакомился с lisp, sbcl, простыми операторами, с тем как определять и вызывать функции. Создал код использующий хвостовые рекурсии, попутно разбирая другие возможные варианты решения задачи.