

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

Студент группы 8О-306 Гаврилов Максим, № по списку 7.

Контакты: cobraj@yandex.ru

Работа выполнена: 22.03.2022

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**

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 № 1.39)**

Функционал sum порождает линейно-рекурсивный процесс. Его можно переписать так, чтобы суммирование выполнялось итеративно. Покажите, как сделать это, заполнив пропущенные выражения в следующем определении:

```
(defun sum (f a b)
  (labels ((iter (a result)
    (if ??
      ??
      (iter ?? ?))))
    (iter ?? ?)))
```

## **4. Оборудование студента**

процессор AMD FX(tm)-6300 Six-Core Processor 3.50 GHz, память 16ГБ, 64-разрядная система.

## **5. Программное обеспечение**

ОС Windows 10, программа portacle, версия slime 2.24

## **6. Идея, метод, алгоритм**

Итеративный процесс в рекурсии

## **7. Сценарий выполнения работы**

1. Изучить приведенные в примерах операторы.
2. На основе шаблона, приведенного в задании, построить функционал, осуществляющий итеративный поиск суммы значений функции.
3. Провести тестирование работы построенного функционала, сравнивая результат его работы с результатом работы приведенного в задании функционала sum.

## 8. Распечатка программы и её результаты

### Программа

```
(defun sum (f a b)
  (labels ((iter (a result)
    (if (> a b)
      result
      (iter (+ a 1) (+ result (funcall f a))))))
    (iter a 0)))
```

### Результаты

```
CL-USER> (defun sum (f a b)
  (labels ((iter (a result)
    (if (> a b)
      result
      (iter (+ a 1) (+ result (funcall f a))))))
    (iter a 0)))
SUM
CL-USER> (sum #'(lambda (x) (* x x)) 1 5)
55
CL-USER>
;сравним результат работы нашего функционала
;и стандартно линейно-рекурсивного функционала
;sum, что приведен в задании
(defun std-sum (f a b)
  (if (> a b)
    0
    (+ (funcall f a) (sum f (1+ a) b))))
STD-SUM
CL-USER> (std-sum #'(lambda (x) (* x x)) 1 5)
55
CL-USER> (sum #'(lambda (x) (+ x (* 2 x x))) 1 5)
125
CL-USER> (std-sum #'(lambda (x) (+ x (* 2 x x))) 1 5)
125
CL-USER> (sum #'(lambda (x) (* x x)) -5 0)
55
CL-USER> (std-sum #'(lambda (x) (* x x)) -5 0)
55
CL-USER> (sum #'(lambda (x) x) 0 100)
5050
CL-USER> (std-sum #'(lambda (x) x) 0 100)
5050
CL-USER>
```

## 9. Дневник отладки

№	Дата, время	Событие	Действие по исправлению	Примечание
1				

## **10. Замечания автора по существу работы**

Так как в задании приведен пример линейно-рекурсивного функционала `sum` (по ссылке), при отладке я сравнивал результат, полученный моей программой, с тем, что получился при вычислении приведенного в пример функционала на тех же входных данных. Их равенство означало успешное прохождение теста.

## **11. Выводы**

В данной лабораторной работе я познакомился с языком Коммон Лисп и написал на нем программу с использованием собственной функции, которая находит сумму значений функции на пространстве аргументов от  $a$  до  $b$ . Программа работает правильно и прошла все тесты.