

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

Студент группы 8О-307 МАИ *Бронников Максим*, №4 по списку

Контакты: `max120199@gmail.com`

Работа выполнена: 15.03.2020

Преподаватель: Иванов Дмитрий Анатольевич, доц. каф. 806

Отчет сдан:

Итоговая оценка:

Подпись преподавателя:

1. Тема работы

Примитивные функции и особые операторы в языке 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 №6)

Запрограммируйте на языке Коммон Лисп функцию с тремя параметрами - действительными числами. Функция должна возвращать сумму квадратов двух больших из них.

4. Оборудование ПЭВМ студента

Процессор Intel© Celeron© CPU @ 2.16GHz x 2, память: 4096Mb, разрядность системы: 64.

5. Программное обеспечение ЭВМ студента

OS Linux Mint 19.3 Cinnamon, среда SLIME 2.24 с реализацией языка SBCL 1.4.5.debian

6. Идея, метод, алгоритм

Функция `square-sum-two-of-three` принимает в качестве аргументов 3 числа и работает следующим образом:

1. Находит сумму квадратов 3-ех аргументов при помощи вспомогательной функции `sqr`, возвращающей квадрат числа(произведение его на самого себя).
2. Находит минимальный элемент из 3-ёх аргументов с помощью функции `minimum`.
3. Вычитает из суммы, найденной на шаге (1) квадрат числа с шага (2). Функция возвращает полученный результат.

7. Сценарий выполнения работы

8. Распечатка программы и её результаты

8.1. Исходный код

```
;;; returns a^2
(defun sqr(a) (* a a))

;;; i redefine function min there
(defun minimum(a b c)
  (if (> a b)
      (if (> b c) c b)
      (if (> a c) c a)))

;;; result func
(defun square-sum-two-of-three(a b c)
  (+ (sqr a) (sqr b) (sqr c) (- (sqr (minimum a b c)))))
```

8.2. Результаты работы

```
(base) max@max-Lenovo-B50-30:~/FuncProg/lab1$ ls
main.lisp
(base) max@max-Lenovo-B50-30:~/FuncProg/lab1$ sbcl
This is SBCL 1.4.5.debian, an implementation of ANSI Common Lisp.
More information about SBCL is available at
  <http://www.sbcl.org/>.
```

SBCL is free software, provided as is, with absolutely no warranty.

It is mostly in the public domain; some portions are provided under

BSD-style licenses. See the CREDITS and COPYING files in the distribution for more information.

```
* (compile-file "./main.lisp")
```

```

; compiling file "/home/max/FuncProg/lab1/main.lisp" (written 15
  MAR 2020 09:09:33 PM):
; compiling (DEFUN SQR ...)
; compiling (DEFUN MINIMUM ...)
; compiling (DEFUN SQUARE-SUM-TWO-OF-THREE ...)

; /home/max/FuncProg/lab1/./main.fasl written
; compilation finished in 0:00:00.016
#P"/home/max/FuncProg/lab1/main.fasl"
NIL
NIL
* (load "main.fasl")

T
* (square-sum-two-of-three 3 -2 4)

25
* (square-sum-two-of-three -5 -2 -109)

29
* (square-sum-two-of-three 17 23 17)

818
* (square-sum-two-of-three 0 0 0)

0
* (square-sum-two-of-three 0 1 0)

1
*
(base) max@max-Lenovo-B50-30:~/FuncProg/lab1$ ls
main.fasl  main.lisp
(base) max@max-Lenovo-B50-30:~/FuncProg/lab1$ exit

```

9. Дневник отладки

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

10. Замечания автора по существу работы

При выполнении работы трудно определить какие функции, определенные в моей реализации языка являются стандартными и определены в других реализациях. Из-за этого мне пришлось реализовать свой аналог функции `min`.

11. Выводы

В процессе выполнения работы я познакомился с новым для себя языком под названием *Common Lisp* и даже написал простейшую программу с использованием стандартных составляющих языка.

Программа работает за константные время и память, поскольку принимает фиксированное количество аргументов и количество действий, выполняемых программой ограничено.

При выполнении задачи мне не пришлось столкнуться с рекурсией в силу простоты реализации моей программы, однако я надеюсь что при выполнении будущих лабораторных у меня получится получить более трудное задание, чтобы глубже изучить особенности пройденного материала.