

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

Студент группы 8О-306 МАИ *Наумов Дмитрий*, №15 по списку

Контакты: dandachok@gmail.com

Работа выполнена: 16.03.2020

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 №48)

Реализуйте на языке Коммон Лисп программу для умножения двух целых чисел за логарифмическое число шагов. Можно использовать функции сложения, вычитания, умножения и деления числа на 2, но нельзя умножать произвольные числа.

4. Оборудование студента

Процессор Intel Core i5-5200 4@2.2GHz, память: 4Gb, разрядность системы: 64.

5. Программное обеспечение

ОС Ubuntu 18.04.4 LTS, clisp 2.49.60

6. Идея, метод, алгоритм

Функция `fast*` рекурсивна и работает следующим образом:

- если один из аргументов ноль, то вернет ноль, иначе
- если второй аргумент равен 1 или -1, то вернет `a` и `-a` соответственно, иначе

- если второй аргумент меньше нуля, то вернет значение функции(как если бы вторым аргументом был положительным) с противоположным знаком, иначе
- если **b** нечетное, то вернет сумму **a** и результат умножения **a** на **b-1**, иначе
- вернет результат умножения **a** на **b/2** умноженный на 2

7. Распечатка программы и её результаты

7.1. Исходный код

```
(defun fast* (a b)
  (if (or (= a 0) (= b 0))
      0
      (if (= b 1)
          a
          (if (= b (- 1))
              (- a)
              (if (< b 0)
                  (- (fast* a (- b)))
                  (if (= (mod b 2) 1)
                      (+ a (fast* a (- b 1)))
                      (* 2 (fast* a (/ b 2))))))))))
```

7.2. Результаты работы

```
eldar@eldar-Aspire-E5-573G: ~/fp/dima/task1$ clisp
[1]> (load "./solution")
;; Загружается файл /home/eldar/fp/dima/task1/solution.fas...
;; Загружен файл /home/eldar/fp/dima/task1/solution.fas
#P"/home/eldar/fp/dima/task1/solution.fas"
[2]> (fast* 1 0)
0
[3]> (fast* 0 1)
0
[4]> (fast* 1 1)
1
[5]> (fast* 2 -1)
-2
[6]> (fast* 2 5)
10
[7]> (fast* 2 4)
8
[8]> (fast* -2 4)
-8
```

```

[9]> (fast* -2 -4)
8
[10]> (fast* 2 -4)
-8
[11]> (fast* 123 345)
42435
[12]> (exit)

```

8. Дневник отладки

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

9. Замечания автора по существу работы

Данное задание схоже с быстрым возведением в степень, которое мне приходилось реализовывать на 2 курсе, поэтому с алгоритмом проблем не возникло.

10. Выводы

Программа работает за логарифмическое время. Но смысла в этом мало, т.к. аппаратно умножение производится за константу, но реализованное аналогично возведение в степень будет быстрее, чем последовательное умножение.