

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

Студент группы 8О-307 МАИ *Ефимов Александр*, №7 по списку  
Контакты: `aleks.efimov2011@yandex.ru`  
Работа выполнена: 27.02.2021

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 Lisp-систему, определять переменные и функции, работать с условными операторами, работать с числами, используя схему линейной и древовидной рекурсии.

## 3. Задание (вариант №1.45)

С помощью формулы

$$\frac{1}{3} \left( \frac{x}{y^2} + 2y \right)$$

запрограммируйте на языке Common Lisp функцию для вычисления кубического корня. Причем у является приближением к кубическому корню из  $x$ .

Использовать функции `good-enough-p`, `improve` и `cube`.

## 4. Оборудование студента

Процессор Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz, память: 7.6Gi, разрядность системы: 64.

## 5. Программное обеспечение

ОС Arch Linux, утилита CLisp.

## 6. Идея, метод, алгоритм

Формула, предоставленная в задании, выводится из метода Ньютона, а именно: необходимо найти некоторый  $x$  – кубический корень числа  $a$  такой, что  $x^3 = a$ . Если это обозначить за функцию:

$$f(x) = x^3 - a$$

то с помощью метода Ньютона можно итеративно найти кубический корень:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} = x_k - \frac{x_k^3 - a}{3x_k^2} = \frac{3x_k^3 - x_k^3 + a}{3x_k^2} = \frac{1}{3} \left( 2x_k + \frac{a}{x_k^2} \right)$$

Каждое новое значение  $x_{k+1}$  может иметь тип *float*, поэтому его нужно сравнивать с оригиналом в пределах машинного (или заданного разработчиком) эпсилона.

## 7. Распечатка программы и её результаты

### 7.1. Исходный код

---

```
1  ;;;; Calculates a cubic root of value
2
3  (defconstant +eps+ short-float-epsilon
4    "Defines an epsilon, which will be used to compare two float variables.")
5
6  (defun to-cube (x) (* x x x))
7
8  (defun eps-equal (a b)
9    "Checks if two values are close enough to be considered equal
10   (float-point comparison)"
11   (<= (abs (- a b)) +eps+))
12
13  (defun calculate-next (previous-x original-x)
14    "Calculates next value in Newton method"
15    (/
16      (+
17        (/ original-x (* previous-x previous-x))
18        (* 2 previous-x))
19      3))
20
21  (defun cuberoot-iteration (previous-x original-x)
22    "Checks if cube of current value is close enough to
23    be considered cube root of original"
24    (if (eps-equal (to-cube previous-x) original-x)
25        previous-x
```

```
26      (cuberoot-iteration
27        (calculate-next previous-x original-x)
28        original-x)))
29
30 (defun cuberoot (x)
31   "Calculates the cube root of the value 'x'"
32   (cuberoot-iteration (float x) (float x)))
```

---

## 7.2. Результаты работы

---

```
[1]> (load "cuberoot.lisp")
;; Loading file cuberoot.lisp ...
;; Loaded file cuberoot.lisp
#P"/home/rookstar/Documents/git/func-prog/lab1/cuberoot.lisp"
[2]> (cuberoot 0.001)
0.100085415
[3]> (cuberoot 1)
1.0
[4]> (cuberoot 1000)
10.0
[5]> (cuberoot 8)
2.0
[6]> (cuberoot 27)
3.0
[7]> (cuberoot 9)
2.0800838
[8]> (to-cube 2.0800838)
9.0
```

---

## 8. Замечания автора по существу работы

Сложность работы искусственно увеличивается ввиду её плохой описанности. Следует хотя бы упомянуть используемый для решения метод Ньютона для упрощения.

## 9. Выводы

При итеративном делении нужно быть аккуратным не передавать переменные с типом *rational*, иначе можно быстро получить ошибку переполнения.