

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

Студент группы 8О-308 МАИ *Марков Александр*, №15 по списку
Контакты: `markov.lifeacc@gmail.com`
Работа выполнена: 11.03.2021

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

3. Задание (вариант №1.27)

Функция f определяется правилом:

$$f(n) = \begin{cases} n & , \text{ если } n < 3 \\ f(n-1) + f(n-2) + f(n-3) & , \text{ если } n \geq 3 \end{cases}$$

Запрограммируйте на языке Коммон Лисп функцию, вычисляющую f с помощью линейно-рекурсивного процесса. Оцените требуемые время вычисления и оперативную память.

4. Оборудование студента

Процессор AMD Ryzen 5 4600H 3.00 GHz, память: 16Gb, разрядность системы: 64.

5. Программное обеспечение

ОС Windows 10, среда LispWorks Personal Edition 7.1.2

6. Идея, метод, алгоритм

Общая формула для $f(n)$ при $n > 3$: $f(n) = f(n-1) + f(n-2) + f(n-3)$. Далее $f(n-1)$ будет называться первым слагаемым для $f(n)$, $f(n-2)$ и $f(n-3)$ - вторым и третьим слагаемым для $f(n)$, соответственно.

Для вычисления функции f используется вспомогательная линейно-рекурсивная функция `f_help`, которая имеет один параметр `n` и возвращает 3 значения, равные $f(n-1)$, $f(n-2)$, $f(n-3)$.

Функция `f_help` работает следующим образом:

- Если $n = 3$, то $f(n-1) = f(2) = 2$, $f(n-2) = f(1) = 1$, $f(n-3) = f(0) = 0$. Возвращаем эти значения.
- Если $n > 3$, то чтобы найти все слагаемые для $f(n)$, нужно найти сначала слагаемые для $f(n-1)$. Первое слагаемое для $f(n-1)$ будет вторым для $f(n)$, второе слагаемое для $f(n-1)$ будет третьим для $f(n)$, а сумма трех слагаемых для $f(n-1)$ будет первым слагаемым для $f(n)$. Слагаемые для $f(n-1)$ находятся вызовом функции `f_help(n-1)`.

Функция `f` работает следующим образом:

- Если $n < 3$, то возвращаем `n`.
- Если $n \geq 3$, то находим слагаемые для $f(n)$ вызовом функции `f_help(n)`. Затем суммируем эти слагаемые и возвращаем результат.

7. Сценарий выполнения работы

8. Распечатка программы и её результаты

8.1. Исходный код

```
(defun f_help(n)
  (if (= n 3)
      (values 2 1 0)
      (multiple-value-bind (p1 p2 p3) (f_help (- n 1))
        (values (+ p1 p2 p3) p1 p2)
      )
  )
)

(defun f(n)
  (if (< n 3)
      n
  )
)
```

```

    (multiple-value-bind (p1 p2 p3) (f_help n)
      (+ p1 p2 p3)
    )
  )
)

```

8.2. Результаты работы

```

CL-USER 1 > (f 0)
0
CL-USER 2 > (f 1)
1
CL-USER 3 > (f 2)
2
CL-USER 4 > (f 3)
3
CL-USER 5 > (f 4)
6
CL-USER 6 > (f 5)
11
CL-USER 7 > (f 6)
20
CL-USER 8 > (f 7)
37
CL-USER 9 > (f 8)
68
CL-USER 10 > (f 9)
125
CL-USER 11 > (f 10)
230

```

9. Дневник отладки

Дата	Событие	Действие по исправлению	Примечание
12.03.2021	Была реализована программа, реализующая функцию с помощью итеративного процесса, вместо линейно-рекурсивного.	Программа была переписана с помощью линейно-рекурсивного процесса.	

10. Замечания автора по существу работы

Задание оказалось несложным, однако в качестве ознакомления с языком вызвало некие затруднения, которые были решены в процессе выполнения.

11. Выводы

В данной лабораторной работе я познакомился с языком Common Lisp и написал на нем программу, вычисляющую значение заданной функции. Программа работает за линейное время и память.