

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

Студент группы М8О-307 МАИ *Безлуцкая Елизавета*, №2 по списку
Контакты: lizabezlutskaya@gmail.com
Работа выполнена: 17.03.2019

Преподаватель: Иванов Дмитрий Анатольевич, доц. каф. 806
Отчет сдан:
Итоговая оценка:
Подпись преподавателя:

Тема работы

Примитивные функции и особые операторы Коммон Лисп.

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.

Задание (вариант №40)

Запрограммируйте на языке Коммон Лисп функционал `product`. Функционал должен вычислять произведение чисел от `a` до `b` и принимать в качестве входных параметров одноместную функцию `f`, а также `a` и `b`.

Оборудование студента

Процессор Intel Core i5-3230M 4 @ 2.6GHz, память: 350Gb, разрядность системы: 64.

Программное обеспечение

Ubuntu 16.04 LTS, clisp compiler

Идея, метод, алгоритм

Функция `product` рекурсивна и работает следующим образом:

- если левая граница больше, чем правая, функция вернет 1

- если левая граница меньше, чем правая, то перемножаем левую границу и результат рекурсивного вызова функции `product` с параметрами `a + 1` и `b`, как только встретим ситуацию, когда `a` равно `b`, то
- вернуть `f` с параметром `b`.

Сценарий выполнения работы

Распечатка программы и её результаты

Исходный код

```
(defun product (f a b)
  (cond ((= a b) (funcall f b))
        ((> a b) 1)
        ((< a b) (* (funcall f a) (product f (1+ a) b))))))
```

Результаты работы

```
(print (product #'1+ -1 4))
0
(print (product #'1+ 1 4))
120
(print (product #'1+ 4 4))
5
(print (product #'1+ 6 4))
1
(print (product #'1+ -5 2))
0
(print (product #'1+ -7 -3))
-720
(print (product #'abs -7 -3))
2520
```

Дневник отладки

| Дата | Событие | Действие по исправлению | Примечание |
|----------|--|---|------------|
| 17.03.19 | В ситуации, когда левая граница больше правой, функция возвращала 1 вместо 0 | Конструкция <code>if</code> заменена на <code>cond</code> с целью добавления условия «левая граница равна правой», при выполнении которого возвращается правая граница, обработанная функцией <code>f</code> . А при выполнении условия «левая граница больше, чем правая», функция возвращает 1. | |
| 18.03.19 | Произведение нулевого количества сомножителей трактуется как 1. | При попадании в ветвь с условием <code>a</code> больше <code>b</code> , выходим с результатом 1. | |

Замечания автора по существу работы

Подозреваю, я не первая, кому было так трудно разобраться в огромном количестве скобок. Если бы не данная особенность языка, программировать на языке Lisp мне было бы приятнее.

Выводы

Большую часть времени я потратила на то, чтобы изучить синтаксис языка, вспомнить необходимые мне конструкции и отключиться от языка C++. Чтобы решить задачу, сначала я нарисовала дерево рекурсии и быстро написала решение, которое мне пришло в голову. Затем придумала тесты, которые потенциально могли сломать его, и привела код в финальный вид. Думаю, ничего сложного.