

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

Студент группы 8О-307 МАИ *Бердикин Тимофей*, №2 по списку

Контакты: timofey.1234@mail.ru

Работа выполнена: 08.04.2020

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 №17)

Доказать, что любую целочисленную денежную сумму, большую 7 рублей, можно выплатить без сдачи трёшками и пятёрками.

Запрограммируйте на языке Коммон Лисп функцию с одним параметром - натуральным числом n - и возвращающую с помощью `values` натуральные числа a и b , такие что $3a + 5b = n$.

4. Оборудование ПЭВМ студента

Ноутбук Asus ROG Strix, Intel® Core™ i7-7700HQ CPU @ 2.80GHz × 8, память: 11,6Gb, разрядность системы: 64.

5. Программное обеспечение ЭВМ студента

OS Linux Ubuntu 20.10, LispWorks.

6. Идея, метод, алгоритм

Для начала доказательство.

Почему бы не считать тройками? Допустим, что числа можно представить как $3n$, $3n + 1$ или $3n + 2$. С числами вида $3n$ всё просто - они тут же набираются n трёшками. А что делать с числами вида $3n + 1$? Путём простых подсчётов я определил, что их можно представить как $3(n - 3) + 10$. Они набираются $(n - 3)$ трёшками и двумя пятёрками. А числа вида $3n + 2$ уже можно представить как $3(n - 1) + 5$. Очевидно, что они набираются $(n - 1)$ трёшками и одной пятёркой. Что и требовалось доказать. Из вышеописанного следует и разложение.

Оно, к слову, не единственно верное. Существуют и другие варианты, но я выбрал именно этот. Пока число не делилось нацело на 3 или на 5, я отнимал 5.

7. Сценарий выполнения работы

8. Распечатка программы и её результаты

8.1. Исходный код

```
(defun exchange (n a b)
  (cond ((= (mod n 3) 0) (values (+ a (/ n 3)) b))
        ((= (mod n 5) 0) (values a (+ b (/ n 5)))))
  (T (exchange (- n 5) a (+ b 1)))))
```

```
(defun change (n)
  (exchange n 0 0))
```

8.2. Результаты работы

```
CLUSER 3 : 1 > ( change 50)
0
10
CLUSER 4 : 1 > (change 90)
30
0
CLUSER 5 : 1 > ( change 65)
0
13
CLUSER 6 : 1 > ( change 12)
4
0
CLUSER 7 : 1 > ( change 13)
1
2
CLUSER 8 : 1 > ( change 77)
```

9. Дневник отладки

Дата	Событие	Действие по исправлению	Примечание
09.04	Undefined operator VAL	val -> values	Ошибка в имени функции

10. Замечания автора по существу работы

Данная работа показалась мне достаточно тяжёлой из-за изучения нового языка программирования с неимперативной парадигмой.

11. Выводы

В процессе выполнения работы я познакомился с новым для себя языком под названием *Common Lisp* и даже написал программу на этом языке.

Было довольно тяжело знакомиться с *LispWorks* и работать в этой среде, она показалась мне очень неудобным инструментом, так что придётся искать альтернативные варианты.

Но любой опыт - это, прежде всего, опыт, и я рад, что у меня всё же получилось запрограммировать лабораторную работу.