

Отчет по лабораторной работе №2 по курсу «Функциональное программирование»

Студентка группы 8О-308 Понагайбо Анастасия, № по списку 13.

Контакты: ponagaibo@mail.ru

Работа выполнена: 01.04.2018

Преподаватель: Иванов Дмитрий Анатольевич, доц. каф. 806

Отчет сдан:

Итоговая оценка:

Подпись преподавателя:

1. Тема работы

Простейшие функции работы со списками Коммон Лисп

2. Цель работы

Научиться конструировать списки, находить элемент в списке, использовать схему линейной и древовидной рекурсии для обхода и реконструкции плоских списков и деревьев.

3. Задание (вариант № 2.42)

Запрограммировать на языке Коммон Лисп функцию, принимающую в качестве аргумента список целых чисел ($X_1 \dots X_n$). Функция должна вернуть список ($Y_1 Z_1 Y_2 Z_2 \dots Y_k Z_k$), где Y_1, \dots, Y_l - чётные элементы исходного списка, взятые в порядке следования, Z_1, \dots, Z_m - нечетные элементы исходного списка, взятые в порядке следования, $k = \min(l, m)$, т.е. результирующий список должен содержать только первые k пар. Для проверки можно использовать встроенные в Коммон Лисп предикаты `evenp` и `oddp`.

4. Оборудование студента

Ноутбук ASUS EeeBook, процессор Intel® Atom™ CPU Z3735F @ 1,33 GHz, память 2ГБ, 32-разрядная система.

5. Программное обеспечение

ОС Windows 8.1, программа CLisp в emacs.

6. Идея, метод, алгоритм

Необходимо обойти заданный список и поместить его элементы в два новых списка — для четных элементов и для нечетных. После этого, пока в обоих списках есть элементы, нужно попарно помещать их в результирующий список: сначала четный элемент, затем нечетный.

7. Сценарий выполнения работы

8. Распечатка программы и её результаты

Программа

```
(defun parse(L E O)
  (if (evenp(first L))
      (setf E (append E (list (first L)))))
```

```

        (setf O (append O (list (first L)))))
    (if (> (length rest L)) 0)
        (parse (rest L) E O)
        (list E O))
    )

(defun merge1(E O R)
  (if (and (> (length E) 0) (> (length O) 0))
      (progn (setf R (append R (list (first E)) (list (first O)))))
    (merge1 (rest E) (rest O) R))
  R)
)

(defun even-odd(L)
  (if (= (length L) 0)
      nil)
  (merge1 (first (parse L () ())) (second (parse L () ())) () )
  )

(print (even-odd (list 1 2)))
(print (even-odd (list 1 10 4 13 7)))
(print (even-odd (list 1 3 5 7)))

```

Результаты

```

(2 1)
(10 1 4 13)
nil

```

9. Дневник отладки

№	Дата, время	Событие	Действие по исправлению	Примечание
1				

10. Замечания автора по существу работы

Для выполнения лабораторной работы потребовалось рекурсивно пройти данный на вход список и определить, является текущий (первый) элемент списка четным или нечетным. Если он четный, то необходимо добавить его к списку четных элементов, иначе добавить к списку нечетных. После этого рекурсивно обходятся полученные списки (пока оба не пусты) и первые их элементы добавляются к результирующему списку. После этого выводится полученный список.

11. Выводы

При выполнении данной лабораторной работы я научилась работать со списками и написала функцию, которая составляет список из пар четных и нечетных элементов исходного списка. В программе использовался предикат `evenp`, функции работы со списками `first`, `second`, `rest`, `length`, `append`, функции `setf`, `list`. Программа работает правильно и прошла все тесты.