ФГБОУ ВО “Чувашский государственный университет им. И. Н. Ульянова”
Факультет: ИВТ
Кафедра: Вычислительной техники
Предмет: Функциональное и логическое программирования

Лабораторная работа №10

Вариант: 6

Рекурсия в Лиспе

Выполнил: студент группы ИВТ-41-20
Галкин Дмитрий
Проверил: доцент Обломов И.А.

Чебоксары 2022 г.

Тема: Основы рекурсивного программирования.
Основные термины, ключевые слова: рекурсия, рекурсия по аргументам, рекурсия по значению, простая рекурсия, параллельная рекурсия, взаимная рекурсия, рекурсия более высокого порядка.

Теория

Рекурсия – Основное средство функционального программирования.
Простая рекурсия – Если рекурсивный вызов встречается в некоторой ветви не более одного раза.

```
(defun copy-lst (lst)
  (cond ((null lst) nil)
        (t (cons (car lst) (copy-lst (cdr lst)))))) .
```

Параллельная рекурсия – Если рекурсия встречается одновременно в нескольких.

```
(defun transform(lst)
  (cond ((null lst) nil)
     ((atom lst) (cons lst nil))
     (t (append (transform (car lst))
                (transform (cdr lst)))))).
```

Взаимная рекурсия – Рекурсия является взаимной между одной и более функциями, если они вызывают друг друга.

```
(defun reverse(lst)
  (cond ((atom lst) lst) (t (permutation lst nil)))).
```

Рекурсия более высокого порядка – Более сложные вычисления можно осуществить с помощью рекурсии более высокого порядка. В качестве аргумента рекурсивного вызова может выступать рекурсивный вызов.

```
(defun ackermann(m n)
 (cond ((= m 0) (+ n 1))
       ((= n 0) (ackermann (- m 1) 1))
       (t (ackermann (- m 1)
          (ackermann m (- n 1)))))).
```

Индивидуальное задание

Определить функцию, единственным аргументом которой являлся бы список списков, объединяющую все списки в одноуровневый список.

```
(defun splitList(list1 list2)
  (cond
    ((null list1) list2)
    (t (cons (car list1) (splitList (cdr list1) list2)))
  )
)
```

```
(defun unionList(oldList)
  (cond
    ((null oldList) NIL)
    (t (splitList (car oldList) (unionList (cdr oldList))))
  )
)

(print (unionList '((a b c) (f z x) (g h j))))
```

Вывод: изучил теоретическую информацию по языку Лисп, на практике использовал знания, полученные для того, чтобы писать программы с выполнением рекурсий.