Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

ОТЧЕТ

к лабораторной работе №2

По курсу: «Функциональное и логическое программирование»

Тема: ***«Списки в Lisp. Использование стандартных функций»***

Студентка ИУ7-65Б
Оберган Т.М

Преподаватель
Толпинская Н.Б

2020 г.

Цель работы: приобрести навыки использования списков и стандартных функций Lisp.

Задачи работы: изучить способ использования списков для фиксации информации, внутреннее представление одноуровневых и структурированных списков, методы их обработки с использованием базовых функций Lisp.

Функция в Лиспе есть однозначное отображение множества исходных данных на множество её значений. У функции может быть произвольно много аргументов, от нуля до любого конечного числа, но обязательно должно быть хотя бы одно значение

Классификация функций:

- Базовые функции – принимают фиксированное количество аргументов
- Формы – принимают не фиксированное количество аргументов или обрабатывают аргументы по разному
- Функционалы (высших порядков) – используют другие функции в качестве аргументов или вырабатывают в качестве результатов.

CAR и CDR являются базовыми функциями доступа к данным. CAR принимает точечную пару или пустой список в качестве аргумента и возвращает первый элемент или nil, соответственно. CDR принимает точечную пару или пустой список и возвращает список состоящий из всех элементов, кроме первого. Если в списке меньше двух элементов, то возвращается Nil.

LIST и CONS являются функциями создания списков (cons – базовая, list – нет). Функция cons создает списочную ячейку и устанавливает два указателя на аргументы. Функция list принимает переменное число аргументов и возвращает список, элементы которого – переданные в функцию аргументы.

Например список '(open close halph) из задания 1 можно представить как: (cons 'open (cons 'close (cons 'halph nil))) или (list 'open 'close 'halph).

Задание 3: найти результат вычисления выражений

Запись (caadr X) эквивалентна (car (car (cdr X))).

a) (CAADR ' ((blue cube) (red pyramid))) -> red

b) (CDAR '((abc) (def) (ghi))) -> nil

(CDAR '((a b c) (d e f) (g h i))) -> (b c)

c) (CADR ' ((abc) (def) (ghi))) -> (def)

d) (CADDR ' ((abc) (def) (ghi))) -> (ghi)

Задание 4: найти результат вычисления выражений

(list 'Fred 'and 'Wilma) -> (Fred and Wilma)
(list 'Fred '(and Wilma)) -> (Fred (and Wilma))
(cons Nil Nil) -> (Nil)
(cons T Nil) -> (T)
(cons Nil T) -> (Nil . T)
(cons t t) -> (T . T)
(cons t (list t)) -> (T T)

(list Nil) -> (Nil)
(cons '(T) Nil) -> ((T))

(list '(one two) '(free temp)) -> ((one two) (free temp))
(cons 'Fred '(and Wilma)) -> (fred and Wilma)
(cons 'Fred '(Wilma)) -> (Fred Wilma)

(list Nil Nil) -> (Nil Nil)
(list T Nil) -> (T Nil)
(list Nil T) -> (Nil T)

(cons T (list Nil)) -> (T Nil)
(list '(T) Nil) -> ((T) Nil)
(cons '(one two) '(free temp)) -> ((one two) free temp)

Задание 5: написать функцию, представить результаты в виде списочных ячеек.

```
(defun f1 (ar1 ar2 ar3 ar4)
        (list (list ar1 ar2) (list ar3 ar4)))
(f1 1 2 3 4) ; ((1 2) (3 4))
```

```
(defun f2 (ar1 ar2)
        (list (list ar1) (list ar2)))
(f2 1 2) ; ((1) (2))

(defun f3 (ar1)
        (list (list (list (ar1))))
(f3 1) ; (((1)))
```