



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 №1

*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.

Базис Lisp образуют: атомы, структуры, базовые функции, базовые функционалы.

Вся информация в Lisp представляется в виде символьных выражений.

Символьное выражение это атом или точечная пара.

Атомами являются: символы – набор литер, специальные символы – {T, Nil}, самоопределимые атомы – числа, строки.

Более сложные данные в Lisp выстраиваются с помощью **бинарных узлов**, содержащих пару указателей. Каждый бинарный узел соответствует минимальному блоку памяти, выделяемому системой при организации и обработке структур данных.

Точечная пара – структура данных, состоящая из двух символьных выражений, разделенных точкой.

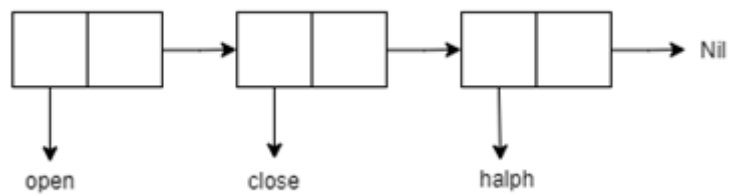
Список – это структура данных. Может быть пустой и непустой. Если непустой, то состоит из двух элементов: первый - любой формы, а второй - список.

В памяти список представляется бинарным узлом, состоящим из двух указателей: *car* – указатель на первый элемент, *cdr* – указатель на оставшуюся часть.

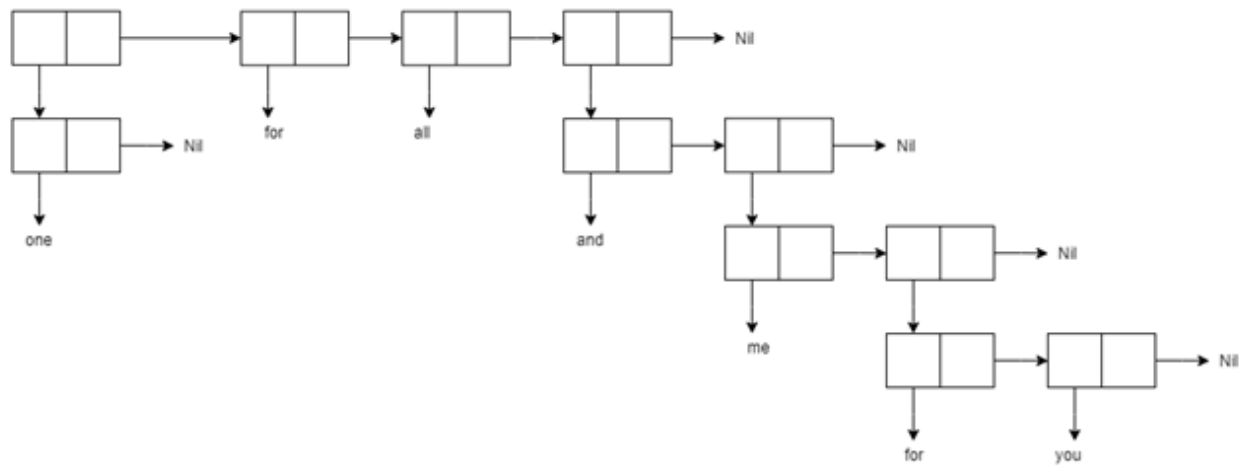
Синтаксически любая структура (точечная пара или список) в языке Lisp заключается в круглые скобки. Точечная пара – (A.B). Пустой список можно задать пустыми скобками () или специальным символом *nil*. Непустой список можно задать через точечную пару (A.(B.())) (в этом случае происходит дублирование разделителей) или как последовательность атомов, разделенных пробелами (A B C).

Задание 1: представить предложенные списки в виде списочных ячеек

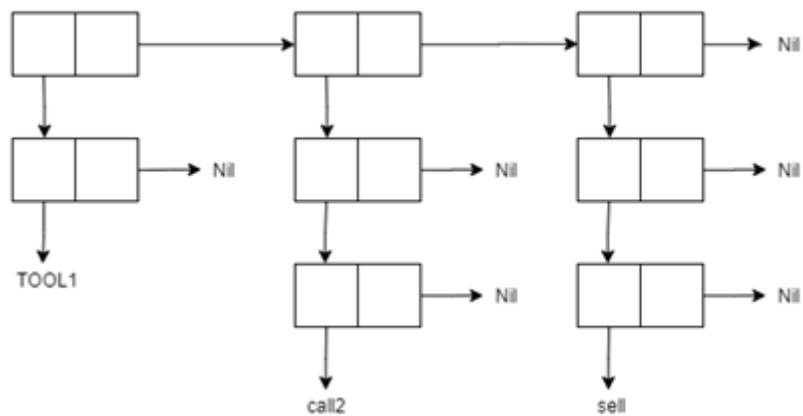
'(open close halph)



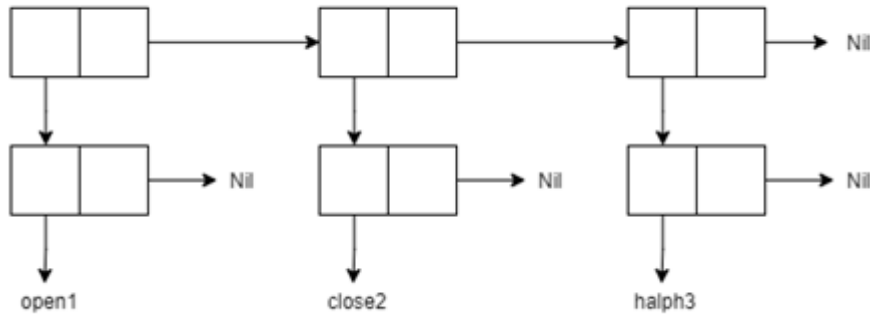
'((one) for all (and (me (for you))))



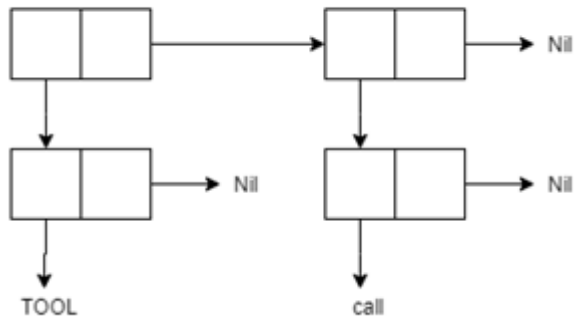
'((TOOL1) ((call2)) ((sell)))



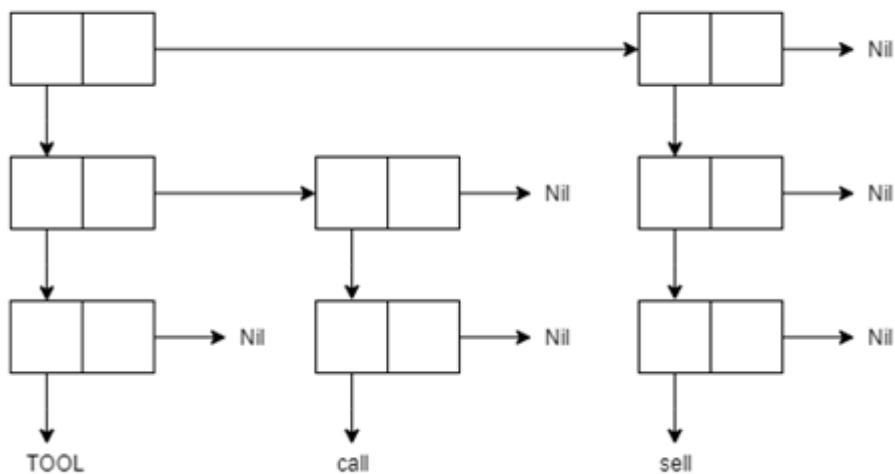
'((open1) (close2) (halph3))



'((TOOL)(call))



'(((TOOL)(call)) ((sell)))



Задание 2: используя функции CAR и CDR, написать выражения, возвращающие второй, третий, четвертый элементы заданного списка.

a) (car (cdr '(1 2 3 4 5))) ; вернет второй элемент

b) (car (cdr (cdr '(1 2 3 4 5)))) ; третий

c) (car (cdr (cdr (cdr '(1 2 3 4 5))))) ; четвертый