



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»

Лабораторная работа №1

Студент: Луговой Д.М.

Группа: ИУ7-61Б

Преподаватель: Толпинская Н.Б.

Москва, 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.

1. Базовые элементы языка

Функциональное программирование ориентировано на символьную обработку данных.

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

Данные и программы в Lisp представлены в виде символьных выражений - S-выражений. Любое S-выражение является атомом или структурой.

Атомы в Lisp:

- Символы - последовательность из букв и цифр, начинающаяся с буквы, включая и другие литеры, не занятые в синтаксисе.
- Специальные символы - логические константы Т и Nil.
- Самовычисляемые атомы - натуральные, дробные и вещественные числа и строки.

Более сложные данные выстраиваются из структур данных - одинаково устроенных блоков памяти. В Lisp это бинарные узлы, содержащие пары объектов произвольного вида. Любая структура в Lisp заключается в круглые скобки.

Точечная пара представляет собой бинарный узел, левая и правая части которого равноправны и могут хранить в себе атомы и точечные пары, разделенные точкой (Примеры: (A.B), (A.(C.D)))

2. Определение списка, его синтаксис

Список – это структура данных. Может быть пустой и непустой. Список может быть пустым. В Lisp возможны два типа представления пустого списка: пара пустых скобок и специальный символ Nil. Синтаксически списком является последовательность из нуля или более элементов, заключенных в скобки и разделенных пробелами. (Пример: (A B C), (A), (A (B)))

3. Функция QUOTE и символ '

И программа, и данные в Lisp представляются в списочной форме. Для явного выделения данных используется функция QUOTE и ' - ее сокращенное обозначение. Эта функция защищает выражение от вычисления.

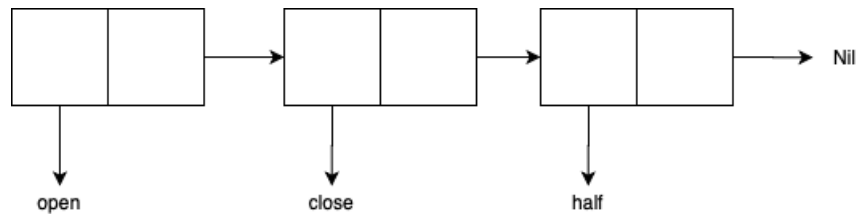
4. Представление списка в памяти

В памяти список представлен списочной ячейкой, содержащей указатели на голову списка и на его хвост. Голова списка представляет из себя S-выражение, хвост является списком.

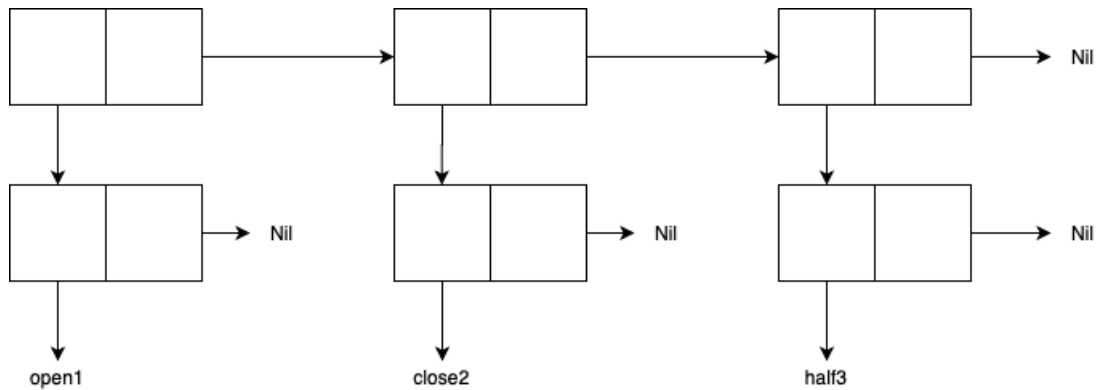
Задание №1

Представить списки в виде списочных ячеек:

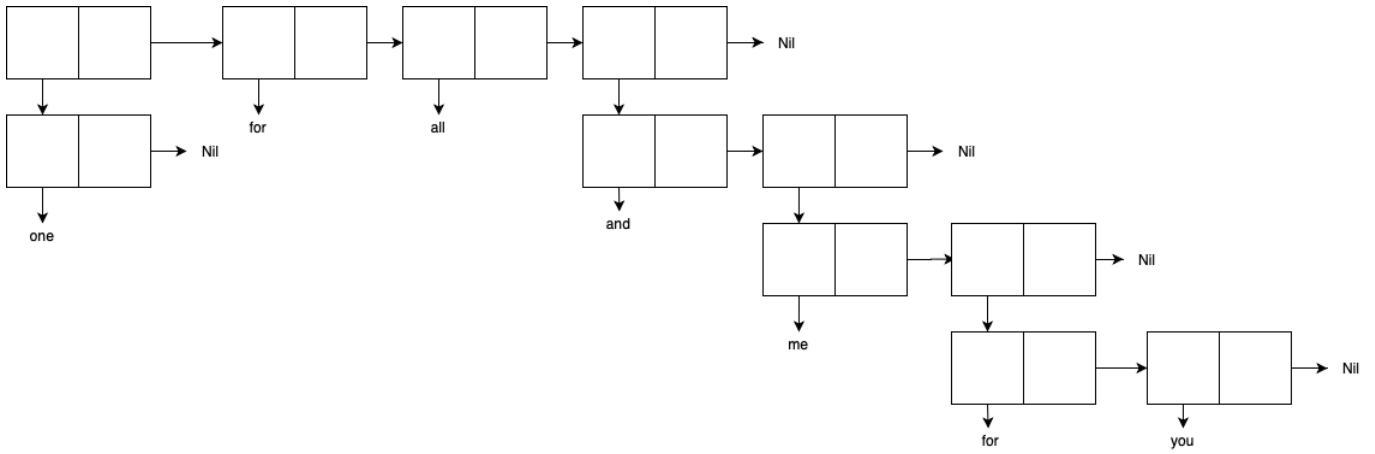
- '(open close half)



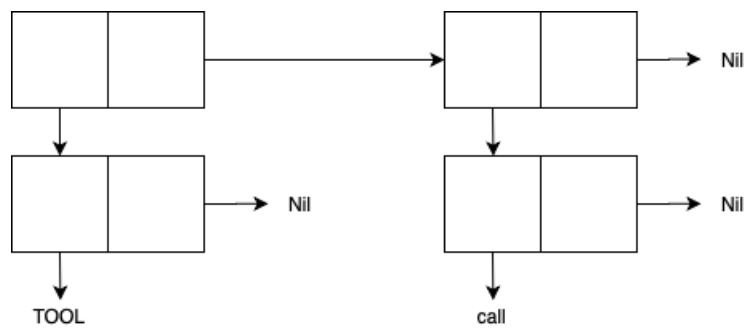
- '((open1)(close2)(half3))



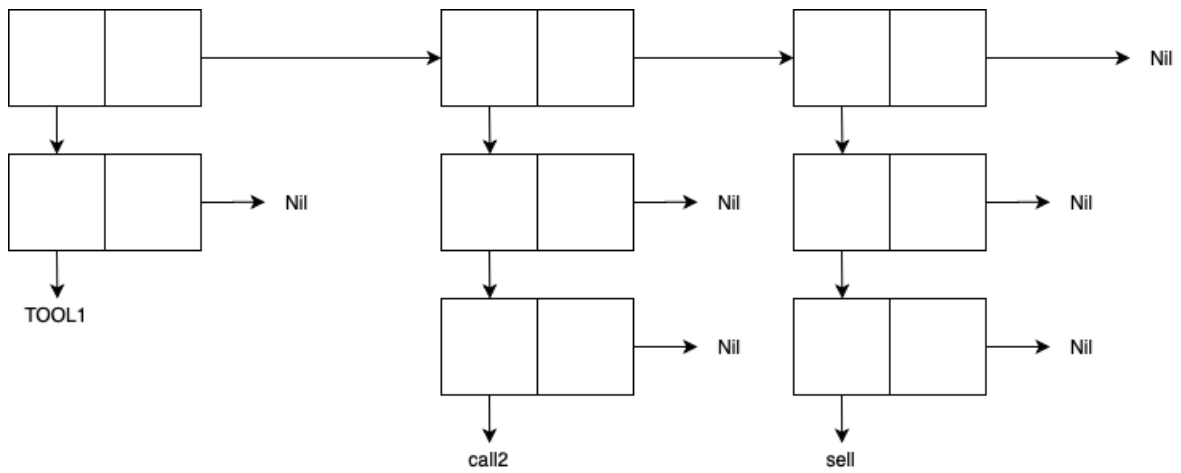
- '((one) for all (and(me(for you))))



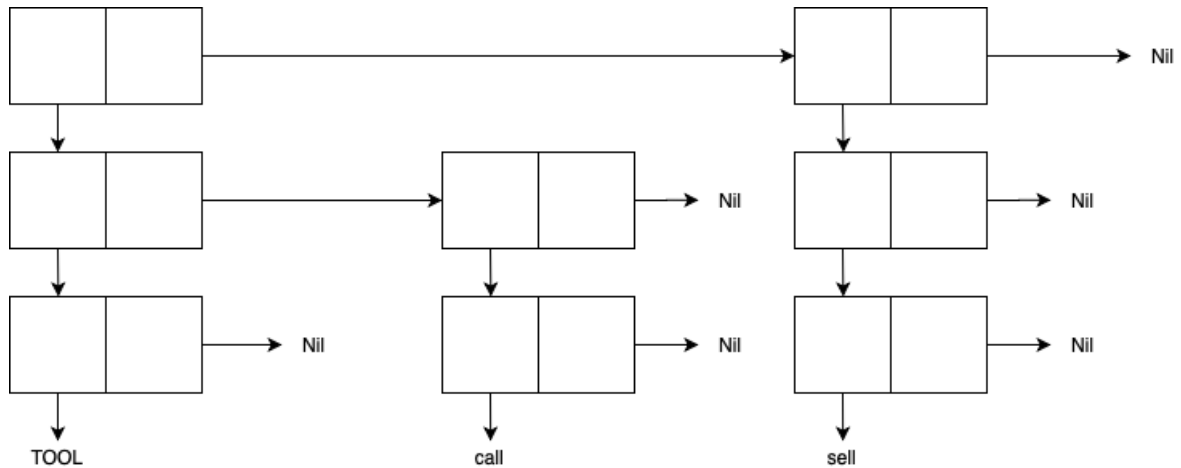
- '((TOOL)(call))



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



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



Задание №2

Используя только функции CAR и CDR, написать выражения, возвращающие:

1. второй элемент списка

`(car (cdr '(1 2 3 4)))`

Пример: `(car (cdr '((1 2) 3 (4 (5)))))` -> 3

2. третий элемент списка

`(car (cdr (cdr '(1 2 3 4))))`

Пример: `(car (cdr (cdr '(1 (2 (3)) ((4) 5 (6))) 7))))` -> ((4) 5 (6))

3. четвертый элемент списка

`(car (cdr (cdr (cdr '(1 2 3 4)))))`

Пример: `(car (cdr (cdr (cdr '(((1 2) ((3 4) 5)) 6 (7 8) ((10) 11)))))` -> ((10) 11)