|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **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*

*По предмету: «Функциональное и логическое программирование»*

**Тема:** Списки в LISP. Использование стандартных функций.

Преподаватель: Толпинская Н.Б.,

Студент: Несчастный Н.Н.,

Группа: ИУ7-65Б

Москва, 2020 г.

**LISP - LIST Processing.**

Язык нечисловых вычислений - с его помощью можно считать, можно использовать функции вместо операторов (имена похожи на знаки: sin, + и тп). Язык нацелен на обработку символьных конструкций - нет четкого разделения на данные и программы; программа может строить себя и выполняться.

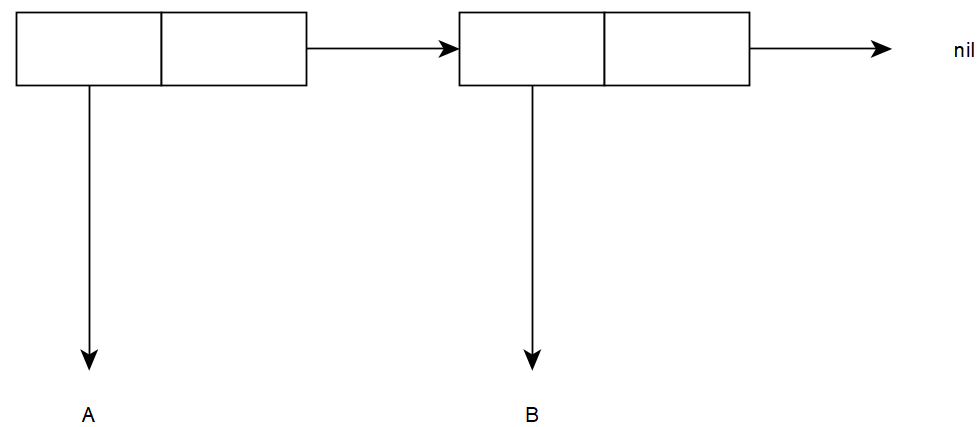
Программа в языке LISP - это список. Элементы списка трактуются так - первый элемент списка - имя функции, остальные элементы - аргументы.

Базис языка:

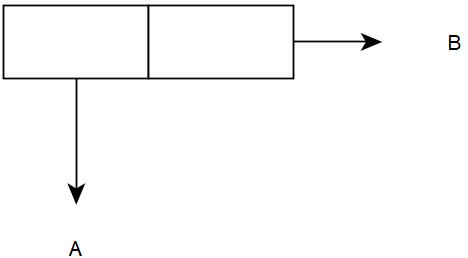
* Атом - символы (идентификаторы) и константы. Константы - числа (разных типов, но вообще типизация отсутствует) и логические (T,Nil). Всё что не Nil то True.
* Точечная пара - два элемента, синтаксически разделенные точкой и заключенные в скобки. (<атом>.<атом>), (<атом>.<точечнаяпара>), (<точечнаяпара>.<атом>), (<точечнаяпара>.<точечнаяпара>).
* S-выражение - либо атом, либо точечная пара.
* Список - специальный тип с-выражения. Список::=Nil | (<S-выражение>.<список>). Таким образом, список - рекурсивно определенная динамическая структура, которая может быть пустой.

**Пример**: (A.(B.(C.Nil))) допустима упрощенная запись (A B C). () - признак конструкции, которая в памяти представляется особым образом. (А) - список, А - атом.

В памяти любому списку соответствует конструкция из двух ячеек - списочная ячейка. Списковая ячейка содержит не данные, но два указателя - на голову и хвост. Хвост в свою очередь также список, которому снова соответствует списковая ячейка.



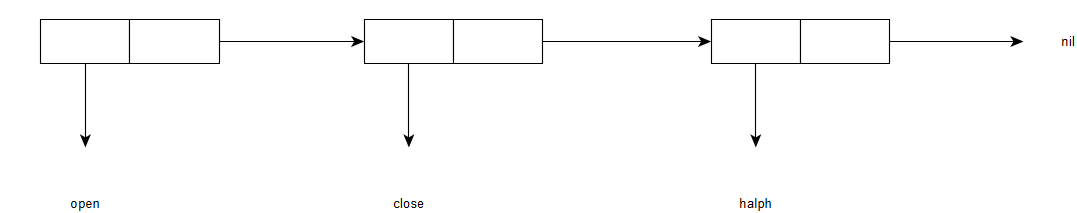
При этом точечная пара (A.B):



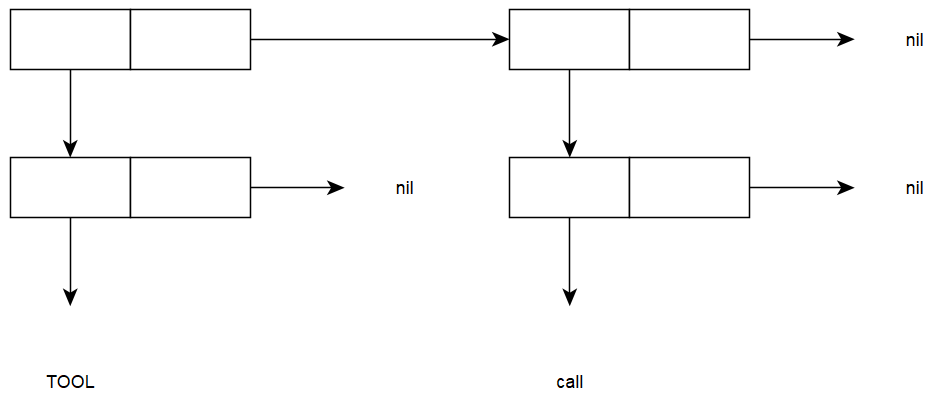
Почему базовой структурой выбран список? Это наиболее общая динамическая структура, на основе которой можно организовать стек, очередь и тд. Список позволяет обрабатывать себя как удобно программе.

**Задание 1: Представить следующие списки в виде списочных ячеек**

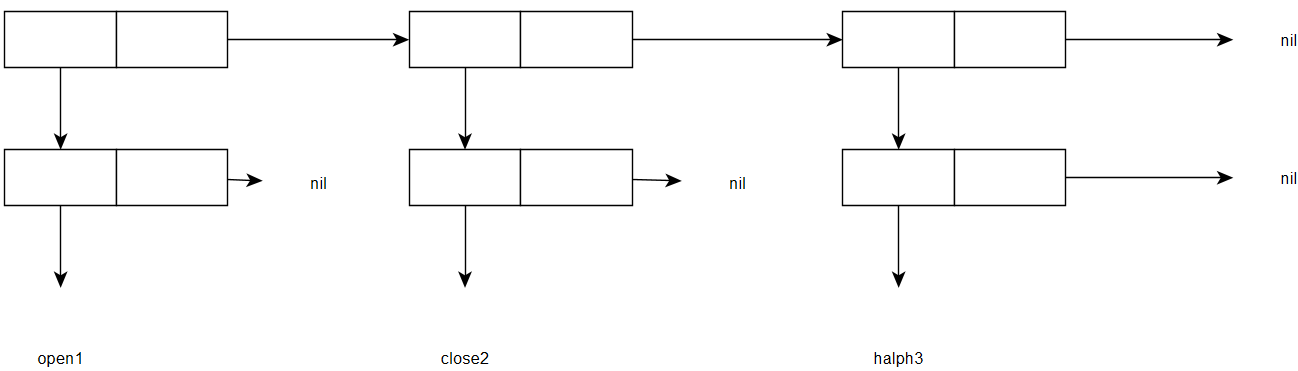
*'(open close halph)*



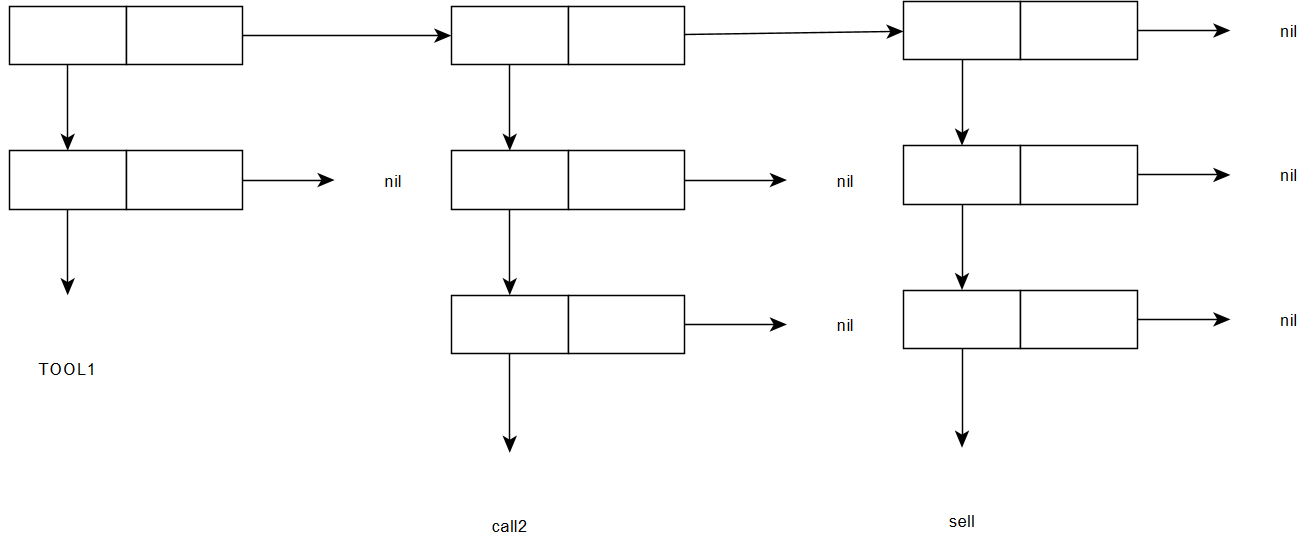
*'((TOOL) (call))*



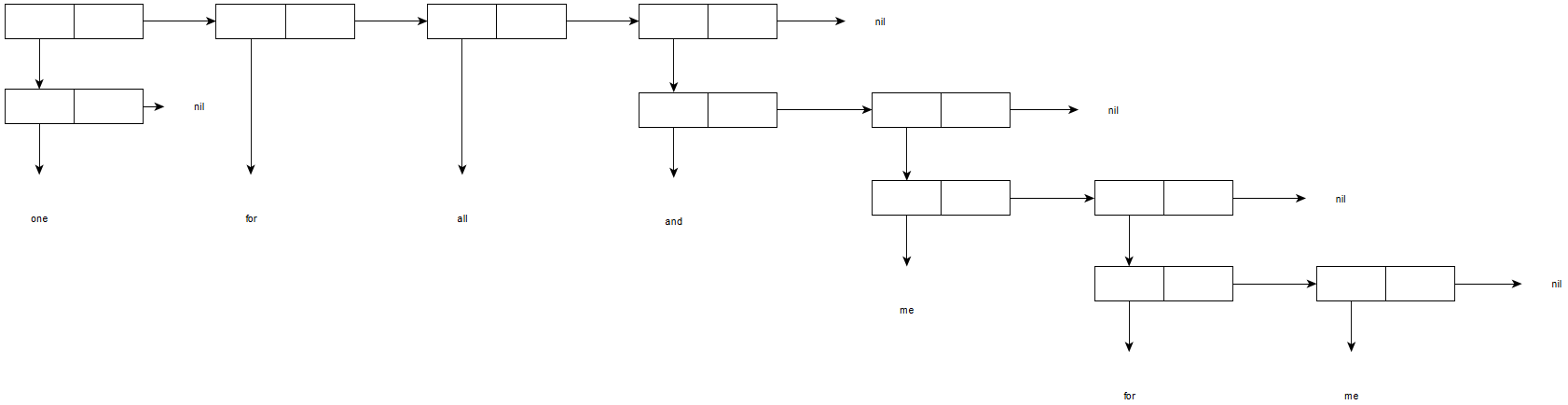
*'((open1) (close2) (halph3))*

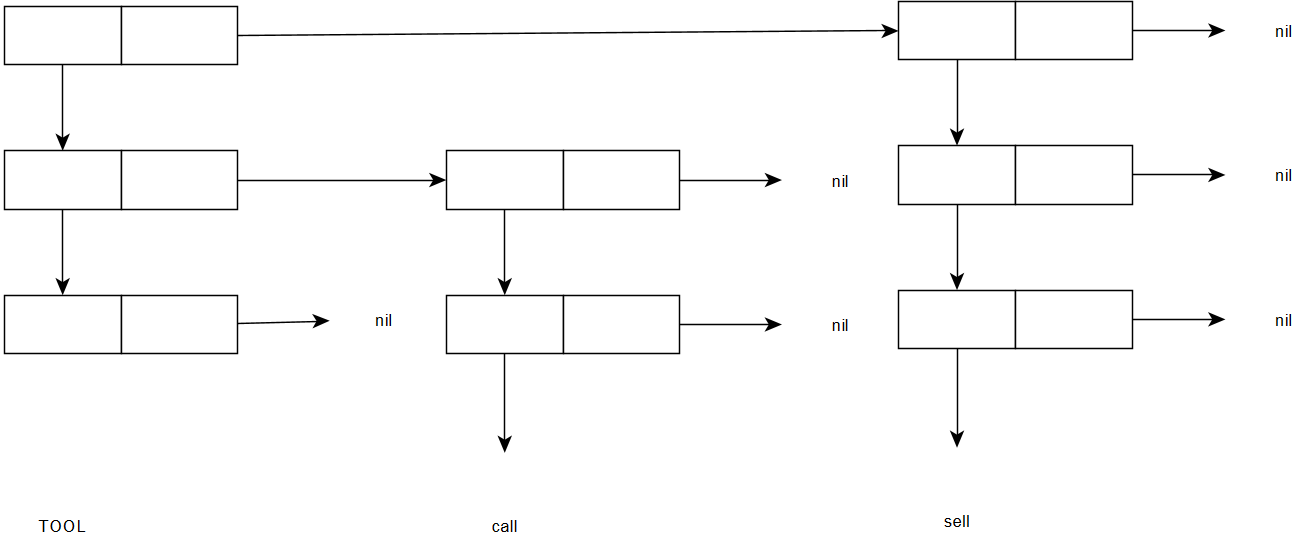


*'((TOOL1) ((call2)) ((sell)))*



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



'(((TOOL) (call)) ((sell))) **Задание 2: Используя только функции car и cdr, написать выражения, возвращающие**

1. *второй элемент заданного списка*
2. *третий элемент заданного списка*
3. *четвертый элемент заданного списка*

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

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

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