Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

«Московский государственный технический университет имени Н.Э. Баумана»

(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

О т ч е т

по лабораторной работе № 1

Дисциплина: «Функциональное и логическое программирование»

Студент гр. ИУ7-65Б _____ А. П. Овчинникова
(Подпись, дата) (И.О. Фамилия)

Преподаватель _____ Н. Б. Толпинская
(Подпись, дата) (И.О. Фамилия)

Москва, 2020

Теоретическая часть.

Списки.

Список – это динамическая структура данных, которая может быть пустой или непустой. Если список не пуст, у него есть «голова» и «хвост», при этом «хвост» является списком.

Списки – одна из базовых структур в Лиспе. Список – это последовательность из нуля или более элементов, заключенных в скобки. Список может быть пустым. В Common Lisp возможны два типа представления пустого списка: пара пустых скобок и специальный символ nil (символ nil является самовычисляемым).

Базовые принципы Лиспа.

1. Лисп – интерактивный язык. Любая Лисп-система имеет интерактивный интерфейс, так называемый верхний уровень (toplevel).
2. В Лиспе используется префиксная нотация (первым располагается оператор).
3. Все выражения в Лиспе – либо атомы, либо списки.
4. Выражение вида *(функция список_аргументов)* – это вызов функции. Аргументом может быть самовычисляемый объект или другой вызов функции. Результат вызова функции вычисляется в два шага. Сначала вычисляются аргументы, слева направо. Затем аргументы применяются к функции, задаваемой оператором. Этот порядок называется правилом вычисления для Common Lisp. Существуют операторы, которые не следуют принятому в Common Lisp порядку вычислений (quote – защита от вычисления, if).
5. Лисп предоставляет все типы данных, которые есть в большинстве других языков (integer – целое число; строка – представляется как последовательность символов, окруженная двойными кавычками; символы – это имена переменных, существующие сами по себе; список).
6. Программы, написанные на Лиспе, представляются в виде списков.
7. Лисп – обработчик списков.

8. Функции, возвращающие логические значения «истина» (t) либо «ложь» (nil), называются предикатами.
9. Простейший условный оператор в Common Lisp – if. Обычно он принимает три аргумента: test-, then- и else-выражения. Сначала вычисляется тестовое test-выражение. Если оно истинно, вычисляется then-выражение («то») и возвращается его значение. В противном случае вычисляется else-выражение («иначе»).
10. Логические операторы and (и) и or (или) могут принимать любое количество аргументов, но вычисляют их до тех пор, пока не будет ясно, какое значение необходимо вернуть. Эти два оператора – макросы. Как и специальные операторы, макросы могут обходить обычный порядок вычисления.
11. Новые функции можно определить с помощью оператора defun. Он обычно принимает три или более аргументов: имя, список параметров и одно или более выражений, которые составляют тело функции.
12. Функциональное программирование – это понятие, которое означает написание программ, работающих через возвращаемые значения, а не изменения среды. Это основная парадигма в Лиспе. Большинство встроенных в Лисп функций не вызывают побочных эффектов.
13. В Лиспе функции – это самые обычные объекты, такие же как символы, строки или списки.

Определения.

Рекурсия – ссылка при описании элемента на этот же элемент.

Если B не является списком, то (A . B) – *точечная пара*. В точечной паре второй элемент может быть списком, а может быть не списком, то есть точечная пара – более общее понятие.

Базисные функции – чистые математические функции.

Чистые функции – функции, имеющие фиксированное количество аргументов.

Более сложные данные выстраиваются из унифицированных структур данных – одинаково устроенных блоков памяти. В Лиспе это *бинарные узлы*, содержащие пары объектов произвольного вида. Каждый бинарный узел соответствует минимальному блоку памяти, выделяемому системой программирования при организации и обработке структур данных. Выделение блока памяти и размещение в нем пары данных выполняет функция CONS, а извлечение левой и правой частей из блока выполняют функции CAR и CDR, соответственно.

Практическая часть.

Задание 1.

1. Представление списка *‘(open close halph)* в виде списочных ячеек изображено на рисунке 1.
2. Представление списка *‘((open1) (close2) (halph3))* в виде списочных ячеек изображено на рисунке 2.
3. Представление списка *‘((one) for all (and (me (for you))))* в виде списочных ячеек изображено на рисунке 3.
4. Представление списка *‘((TOOL)(call))* в виде списочных ячеек изображено на рисунке 4.
5. Представление списка *‘((TOOL1) ((call2)) ((sell)))* в виде списочных ячеек изображено на рисунке 5.
6. Представление списка *‘(((TOOL)(call)) ((sell)))* в виде списочных ячеек изображено на рисунке 6.

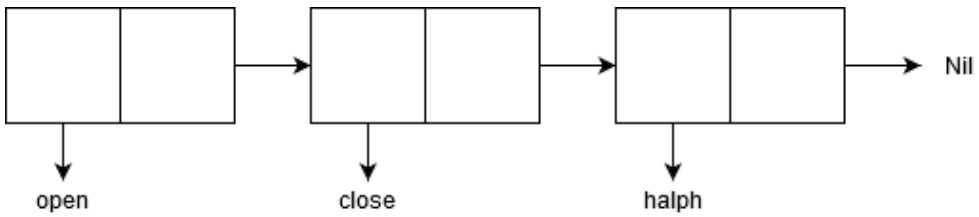

Рис. 1. *‘(open close halph)*

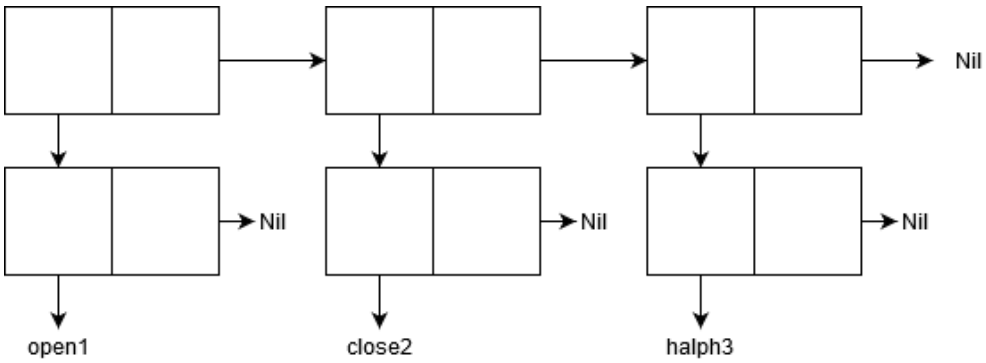

Рис. 2. *‘((open1) (close2) (halph3))*

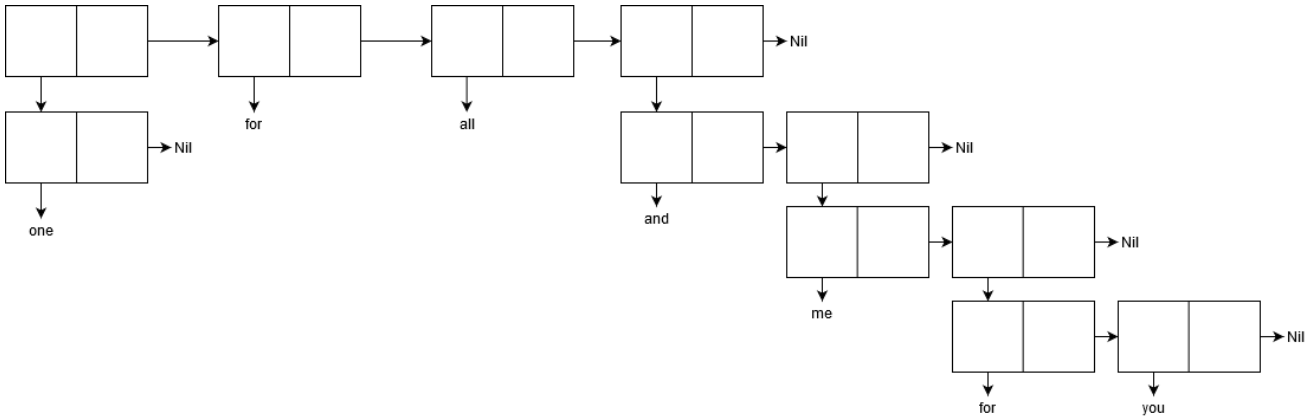

Рис. 3. *‘((one) for all (and (me (for you))))*

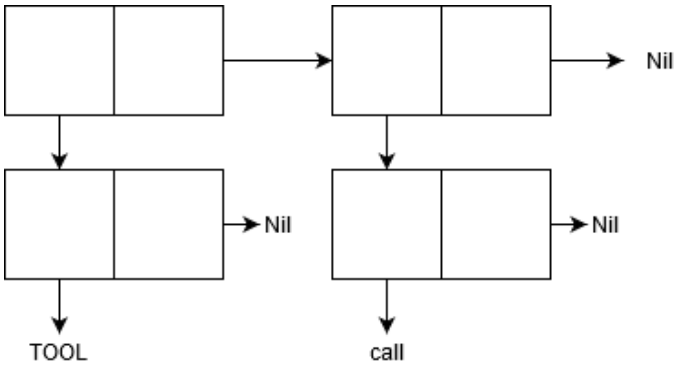

Рис. 4. *‘((TOOL)(call))*

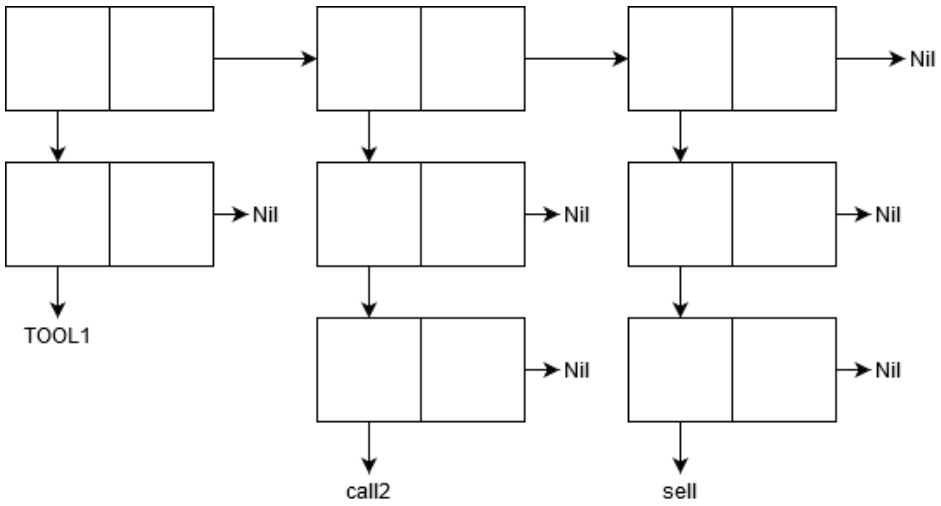

Рис. 5. *‘((TOOL1) ((call2)) ((sell)))*

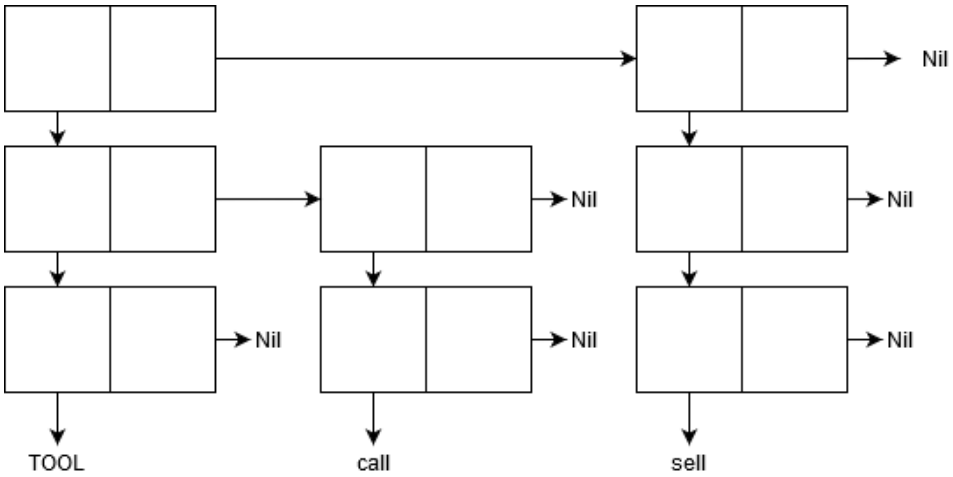

Рис. 5. *‘(((TOOL)(call)) ((sell)))*

Задание 2.

1. Функция, возвращающая второй элемент заданного списка:
   (car (cdr '(A B C)))
2. Функция, возвращающая третий элемент заданного списка:
   (car (cdr (cdr '(A B C D))))
3. Функция, возвращающая четвертый элемент заданного списка:
   (car (cdr (cdr (cdr '(A B C D E)))))