



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»

Дисциплина: «Функциональное и логическое
программирование»

Лабораторная работа №1

Студент: Левушкин И. К.

Группа: ИУ7-62Б

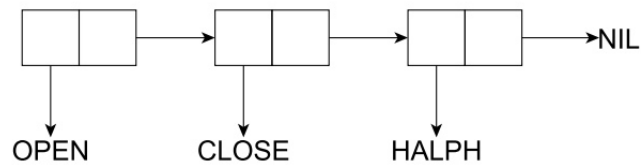
Преподаватели: Толпинская Н. Б.,

Строганов Ю. В.

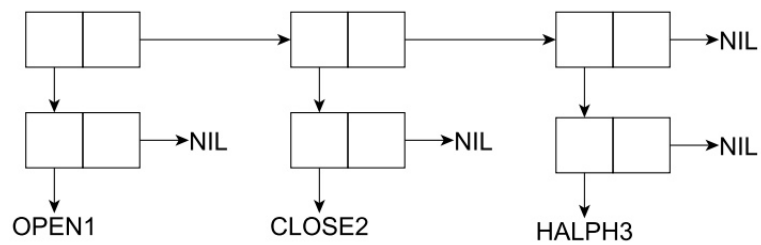
Москва, 2020 г.

1 Представить списки в виде списочных ячеек

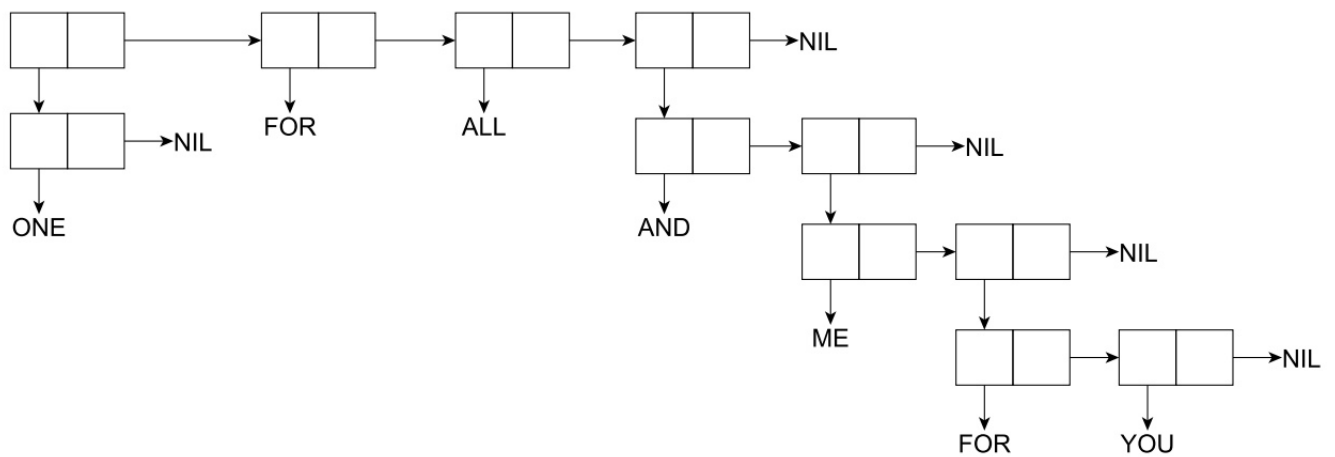
1.1 '(open close halph)



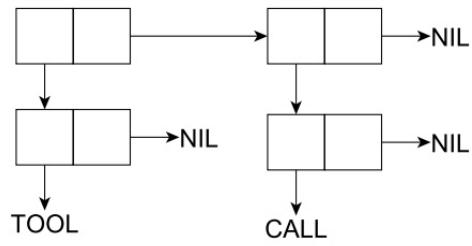
1.2 '((open1) (close2) (halph3))



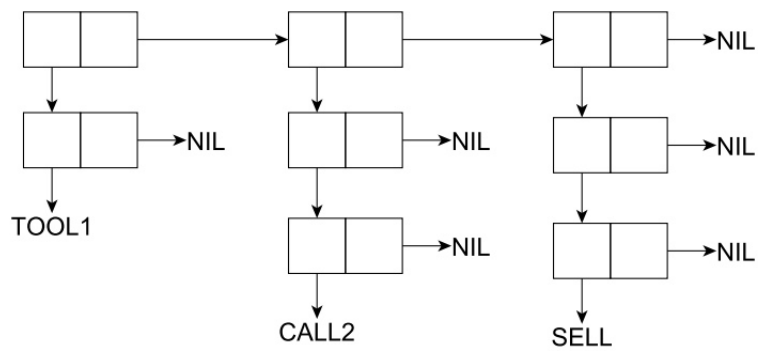
1.3 '((one) for all (and (me (for you))))



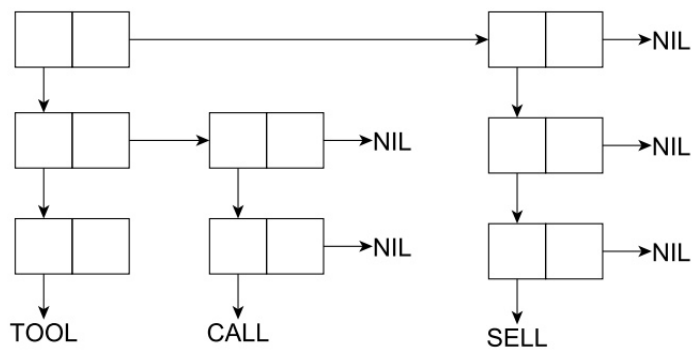
1.4 '((TOOL) (call))



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



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



2 Используя только функции CAR и CDR, написать выражения, возвращающие необходимый элемент списка:

2.1 второй

| Выражение | Результат |
|---|---------------|
| (cadr '(a b c d e)) | b |
| (cadr '((1 2) (3 4) (5 6))) | (3 4) |
| (cadr '((one) for all (and (me)))) | for |
| cadr '(abc ((def) 1 y 3) (4 5) u v (z)) | ((def) 1 y 3) |

2.2 третий

| Выражение | Результат |
|---|-----------|
| (caddr '(a b c d e)) | c |
| (cadr '((1 2) (3 4) (5 6))) | (5 6) |
| (cadr '((one) for all (and (me)))) | all |
| cadr '(abc ((def) 1 y 3) (4 5) u v (z)) | (4 5) |

2.3 четвертый

| Выражение | Результат |
|---|------------|
| (cadddr '(a b c d e)) | d |
| (cadddr '((1 2) (3 4) (5 6))) | Nil |
| (cadddr '((one) for all (and (me)))) | (and (me)) |
| cadddr '(abc ((def) 1 y 3) (4 5) u v (z)) | u |

Рекурсивное определение списка

Список является динамической структурой, может быть как пустым, так и непустым. Если он непустой, то должен иметь голову и хвост, который также представляется списком.

Базовые принципы Lisp

1. Условные высказывания
2. Функциональный тип данных

3. Рекурсия
4. Динамическое типизирование
5. Сборка мусора
6. Программы, составленные из выражений
7. Символьный тип
8. Нотации для кода, использующего деревья символов и констант
9. Постоянная целостность языка