



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»

---

Отчёт к лабораторной работе №1  
по курсу: «Функциональное и логическое  
программирование»  
Тема: **Списки в Lisp. Использование стандартных  
функций**

Студент группы ИУ7-62Б

А.П. Сорокин

(И.О. Фамилия)

Преподаватель

Н.Б. Толпинская

(И.О. Фамилия)

2020 г.

## СОДЕРЖАНИЕ

<b>1</b>	<b>Цели и задачи</b>	<b>2</b>
<b>2</b>	<b>Теоретическая часть</b>	<b>3</b>
<b>3</b>	<b>Практическая часть</b>	<b>4</b>
3.1	Задание 1 . . . . .	4
3.2	Задание 2 . . . . .	7

## 1 Цели и задачи

**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.

## 2 Теоретическая часть

**Список** – динамическая структура данных, которая может быть пустая или непустая, состоящая из головы и хвоста, являющимся списком. Описание списка в Lisp представлено в листинге 1.

Листинг 1 – Описание списка в Lisp

```
Список ::= < пустой список > | < непустой список >, где  
< пустой список > ::= ( ) | Nil,  
< непустой список > ::= (< первый элемент > . < хвост >),  
< первый элемент > ::= < выражениеS- >,  
< хвост > ::= < список >.
```

---

### Базовые принципы Lisp:

- 1) Все данные представляются в форме символьных выражений, называемых S-выражениями.
- 2) Описание способов обработки S-выражений представляется программами, рассматриваемыми как символьные данные. Определения и вызовы этих функций имеют вид S-выражений, то есть формально они могут обрабатываться как обычные данные, получаться в процессе вычислений и преобразовываться как значения.
- 3) Система функционального программирования допускает, что программа может интерпретировать и/или компилировать программы, представленные в виде S-выражений. Это сближает методы функционального программирования с методами низкоуровневого программирования и отличается от традиционной методики применения языков высокого уровня.

### 3 Практическая часть

#### 3.1 Задание 1

Представить следующие списки в виде списочные ячеек:

- '(open close halph)
- '((open1) (close2) (halph3))
- '((one) for all (and(me(for you))))
- '((TOOL) (call))
- '((TOOL1) ((call2)) ((sell)))
- '(((TOOL) (call)) ((sell)))

Решения представлены на рисунках 1 - 6.

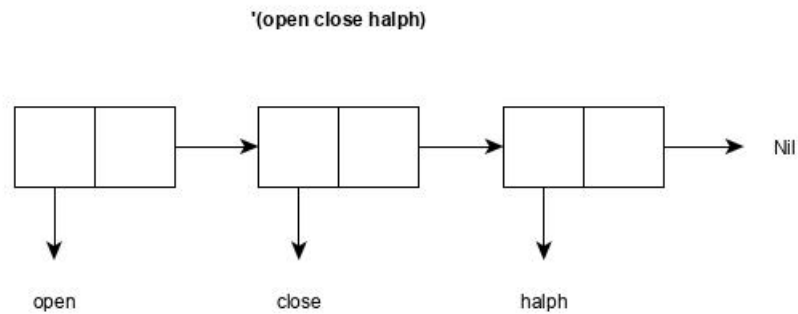


Рисунок 1 – '(open close halph)

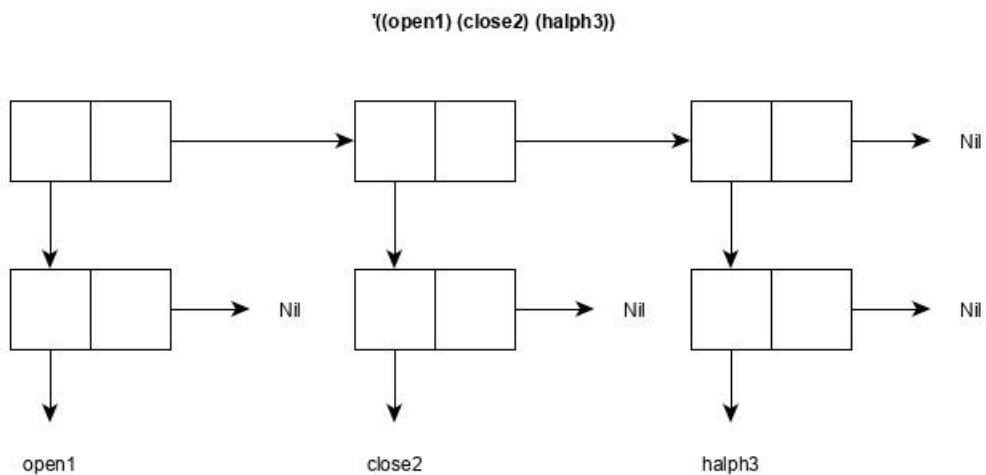


Рисунок 2 – '((open1) (close2) (halph3))

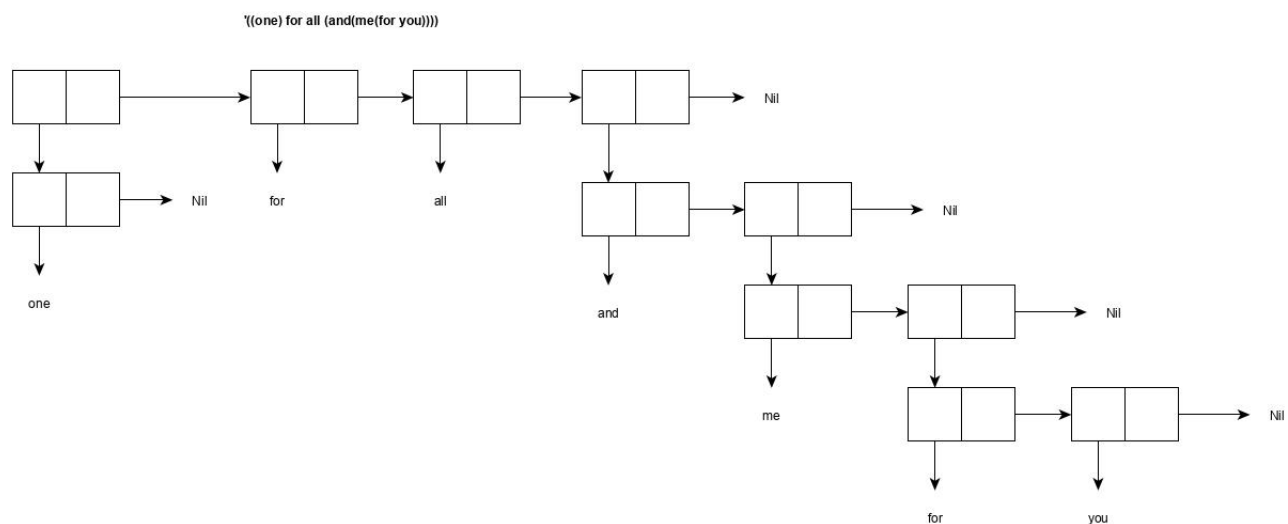


Рисунок 3 – '((one) for all (and(me(for you))))

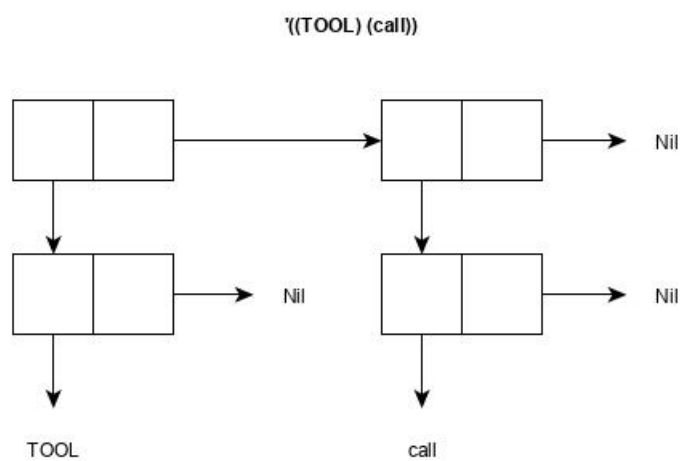


Рисунок 4 – ((TOOL) (call))

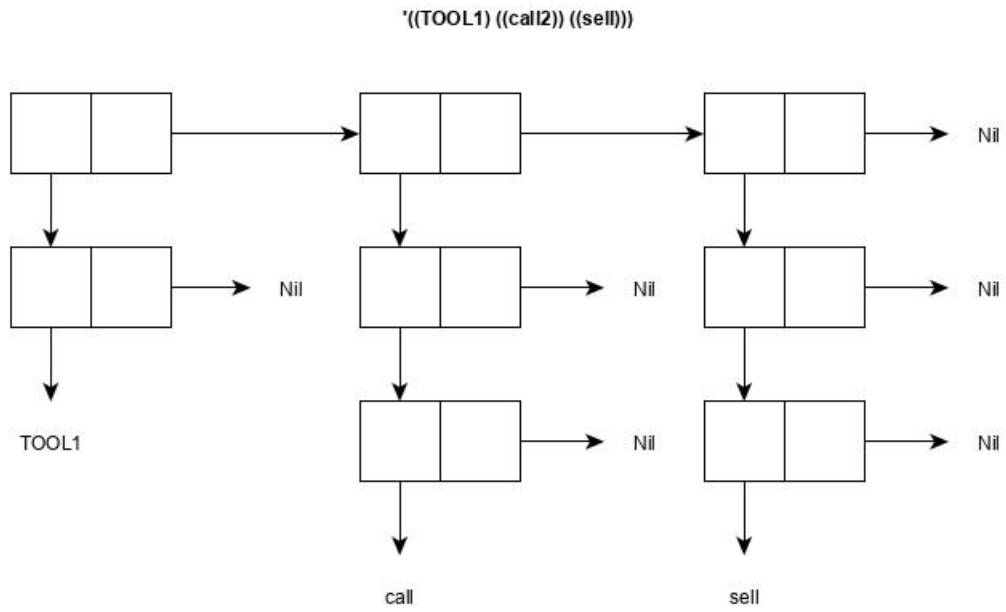


Рисунок 5 – '((TOOL1) ((call2)) ((sell)))

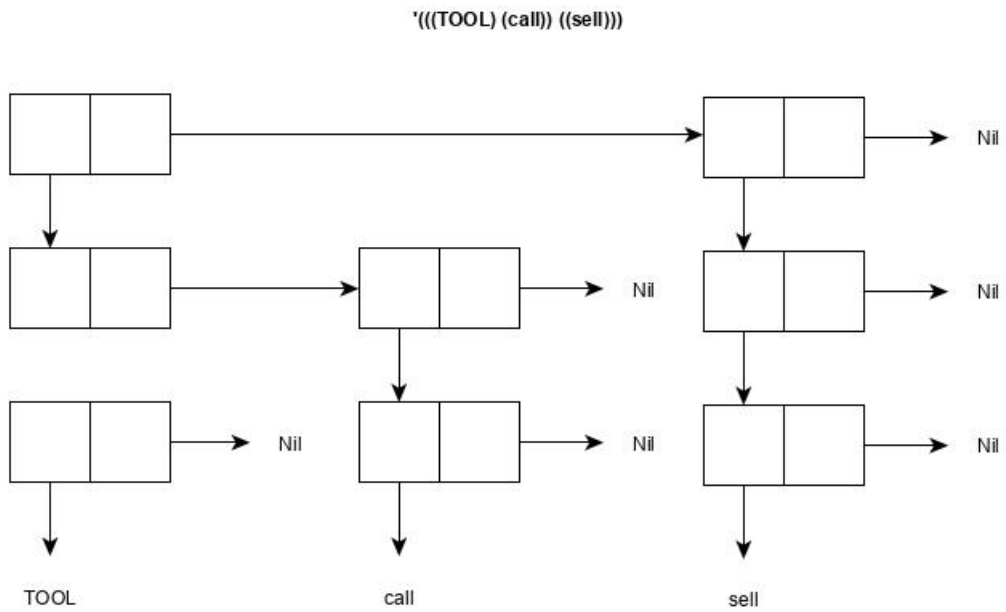


Рисунок 6 – '(((TOOL) (call)) ((sell)))

### 3.2 Задание 2

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

- 1) второй;
- 2) третий;
- 3) четвертый элементы заданного списка.

Решение представлено в листинге 2, а результат выполнения представленного кода - в листинге 3.

Листинг 2 – Код задания 2

```
(car (cdr '((open1) (close2) (halph3)))) ; 2nd element
(car (cdr (cdr '(open close halph)))) ; 3rd element
(car (cdr (cdr (cdr '((one) for all (and(me(for you)))))))) ; 4th
element
```

Листинг 3 – Результаты выполнения кода 2

```
(CLOSE2)
HALPH
(AND (ME (FOR YOU)))
```