



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, искусственный интеллект и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Отчет по лабораторной работе №1
по курсу «Функциональное и логическое
программирование»

«Списки в Lisre. Использование стандартных функций»

Студент группы **ИУ7-64Б**

(Подпись, дата)

Д.С. Чепиго

(И.О. Фамилия)

Преподаватели

(Подпись, дата)

Н.Б. Толпинская

(И.О. Фамилия)

(Подпись, дата)

Ю.В. Строганов

(И.О. Фамилия)

2023 г.

Содержание

1	Теоретические вопросы	3
1.1	Элементы языка: определение, синтаксис, представление в памяти.	3
1.2	Особенности языка Lisp. Структура программы. Символ апостроф.	4
1.3	Базис языка Lisp. Ядро языка.	4
2	Практические задания	5
2.1	Задание №1	5
2.2	Задание №2	5
2.3	Задание №3	5
2.4	Задание №4	6
2.5	Задание №5	6

1 Теоретические вопросы

1.1 Элементы языка: определение, синтаксис, представление в памяти.

Вся информация (данные и программы) в Lisp представляется в виде символьных выражений – S-выражений.

По определению **S-выражение** ::= <атом>|<точечная пара>

Атомы бывают:

- символы (идентификаторы) – синтаксически – набор литер (букв и цифр), начинающихся с буквы;
- специальные символы – {T, Nil} (используются для обозначения логических констант);
- самоопределимые атомы – натуральные числа, дробные числа (например 2/3), вещественные числа, строки – последовательность символов, заключенных в двойные апострофы (например "abc").

Более сложные данные – списки и точечные пары (структуры) строятся из унифицированных структур – блоков памяти – бинарных узлов.

Точечные пары ::= (<атом>.<атом>) | (<атом>.<точечная пара>) | (<точечная пара>.<атом>) | (<точечная пара>.<точечная пара>);

Список ::= <пустой список> | <непустой список>, где

<пустой список> ::= () | Nil,

<непустой список> ::= (<первый элемент> . <хвост>),

<первый элемент> ::= <S-выражение>,

<хвост> ::= <список>.

Синтаксически: любая структура (точечная пара или список) заключается в круглые скобки (A . B) – точечная пара, (A) – список из одного элемента, пустой список изображается как Nil или ();

1.2 Особенности языка Lisp. Структура программы. Символ апостроф.

Особенности языка Lisp:

- бестиповый язык;
- символьная обработка данных;
- любая программа может интерпретироваться как функция с одним или несколькими аргументами;
- автоматизированное динамическое распределение памяти, которая выделяется блоками;
- программа может быть представлена как данные, то есть программа может изменять саму себя.

Символ апостроф – сокращенное обозначение функции `quote`, блокирующей вычисление своего аргумента.

1.3 Базис языка Lisp. Ядро языка.

Базис языка – минимальный набор конструкций и структур данных, с помощью которого можно написать любую программу. Базис Lisp образуют:

- атомы;
- структуры;
- базовые функции;
- базовые функционалы.

2 Практические задания

2.1 Задание №1

Сдано на отдельном листе.

2.2 Задание №2

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

1. второй;

```
1 (car (cdr '(l i s t)))
```

2. третий;

```
1 (car (cdr (cdr '(l i s t))))
```

3. четвертый элемент заданного списка.

```
1 (car (cdr (cdr (cdr '(l i s t)))))
```

2.3 Задание №3

Что будет в результате вычисления выражений?

1. (CAADR '((blue cube) (red pyramid)))
red
2. (CDAR '((abc) (def) (ghi)))
nil
3. (CADR '((abc) (def) (ghi)))
(def)
4. (CADDR '((abc) (def) (ghi)))
(ghi)

2.4 Задание №4

Напишите результат вычисления выражений и объясните как он получен:

```
1 (list 'Fred 'and 'Wilma)           ; (FRED AND WILMA)
2 (list 'Fred '(and Wilma))          ; (FRED (AND WILMA))
3 (cons Nil Nil)                     ; (NIL)
4 (cons T Nil)                       ; T
5 (cons Nil T)                       ; (NIL . T)
6 (list Nil)                         ; (NIL)
7 (cons '(T) Nil)                    ; ((T))
8 (list '(one two) '(free temp))     ; ((ONE TWO) (FREE TEMP))
9
10 (cons 'Fred '(and Wilma))          ; (FRED AND WILMA)
11 (cons 'Fred '(Wilma))              ; (FRED WILMA)
12 (list Nil Nil)                     ; (NIL NIL)
13 (list T Nil)                       ; (T NIL)
14 (list Nil T)                       ; (NIL T)
15 (cons T (list Nil))                ; (T NIL)
16 (list '(T) Nil)                    ; ((T) NIL)
17 (cons '(one two) '(free temp))     ; ((ONE TWO) FREE TEMP)
```

2.5 Задание №5

Написать лямбда-выражение и соответствующую функцию:

1. Написать функцию (f ar1 ar2 ar3 ar4), возвращающую список: ((ar1 ar2) (ar3 ar4)).

```
1 ((lambda (ar1 ar2 ar3 ar4)
2 (list (list ar1 ar2) (list ar3 ar4)))
3 1 2 3 4)
4
5 (defun f1 (ar1 ar2 ar3 ar4)
6 (list (list ar1 ar2) (list ar3 ar4)))
```

2. Написать функцию (f ar1 ar2), возвращающую ((ar1) (ar2)).

```
1 ((lambda (ar1 ar2)
2   (list (list ar1) (list ar2)))
3 1 2)
4
5 (defun f1 (ar1 ar2)
6   (list (list ar1) (list ar2)))
```

3. Написать функцию (f ar1), возвращающую (((ar1))).

```
1 ((lambda (ar1)
2   (list (list (list ar1)))))
3 1)
4
5 (defun f1 (ar1)
6   (list (list (list ar1))))
```

Результаты в виде списочных ячеек представлены на отдельном листе.