



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»

НАПРАВЛЕНИЕ ПОДГОТОВКИ _____ «09.03.04 Программная инженерия»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1

Название: _____ Списки в Lisre. Использование стандартных функций

Дисциплина: _____ Функциональное и логическое программирование

Студент	<u>ИУ7-64Б</u>	_____	<u>С. Д. Параскун</u>
	Группа	Подпись, дата	И. О. Фамилия
Преподаватель		_____	<u>Н. Б. Толпинская</u>
		Подпись, дата	И. О. Фамилия
Преподаватель		_____	<u>Ю. В. Строганов</u>
		Подпись, дата	И. О. Фамилия

Москва, 2022 г.

1. Теоретические вопросы

1.1 Элементы языка: определение, синтаксис, представление в памяти

Элементы языка — атомы и точечные пары (структуры, которые строятся с помощью унифицированных структур - блоков памяти - бинарных узлов). Атомы бывают:

- **символы** (идентификаторы) — синтаксически представляют собой набор литер (последовательность букв и цифр, начинающаяся с буквы; могут быть связанные и несвязанные);
- **специальные символы** — используются для обозначения «логических» констант (T, Nil);
- **самоопределимые атомы** — числа, строки - последовательность символов в кавычках ("abc").

1.2 Синтаксис элементов языка

Точечная пара ::= (<атом> . <атом>) | (<точечная пара> . <атом>)
| (<атом> . <точечная пара>) | (<точечная пара> . <точечная пара>)

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

<пустой список> ::= () | Nil,

<непустой список> ::= (<S-выражение> . <список>),

Список — частный случай S-выражения.

Синтаксически любая структура (точечная пара или список) заключается в круглые скобки: (A . B) — точечная пара. (A) — список из одного элемента. Непустой список — (A . (B . (C . (D . Nil)))) или (A B C D) Пустой список — Nil или ().

Элементы списка могут быть списками, например — ((A (B C) (D (E)))). Таким образом, синтаксически наличие скобок является признаком структуры — списка или точечной пары.

Любая непустая структура Lisp в памяти представляется списковой ячейкой, хранящий два указателя: на голову (первый элемент) и хвост (все остальное).

1.3 Особенности языка Lisp. Структура программы. Символ апостроф

Можно выделить следующие особенности языка Lisp.

1. Использование символьной обработки.
2. Программа может быть представлена в виде данных, что позволяет ей самой изменять себя.
3. Отсутствие типизации (типов данных).
4. Блочное выделение памяти, этим занимается сам Lisp.
5. Программы представляются в виде списков.

Программа на языке Lisp записывается в виде последовательности атомов и списков, т.е. программа и данные имеют одинаковый синтаксис. Другими словами программа представляет собой последовательность форм, а форма или вычислимое выражение — это атом или список, который можно вычислить и получить значение. Как специальная функция **quote**. Данная функция блокирует вычисления своего единственного аргумента, то есть он воспринимается как константа. При выполнении функции аргумент обрабатывается по общей схеме.

1.4 Базис языка Lisp. Ядро языка

- **car** — возвращает первый элемент списка, являющегося значением ее единственного аргумента;

- `cdr` – возвращает хвост списка, являющегося значением его единственного аргумента;
- `cons` – строит новый список, первым элементом которого является первый аргумент, хвостом – второй;
- `atom` – возвращает значение `T`, если аргумент является атомом, иначе возвращает `Nil`;
- `eq` – проверяет совпадение двух своих аргументов-атомов, возвращает значение `T` если значение одного из его аргументов – атом и одновременно значения аргументов эквивалентны;
- `quote` – в качестве значения выдает аргумент, не вычисляя его значения;
- `eval` – выполняет двойное вычисление своего аргумента, чаще всего применяется для снятия блокировки функцией `quote`;
- `cond` – условное выражение, записывается как $(\text{cond } (p_1 e_1) .. (p_n e_n))$, где $(p_i e_i)$ – ветви условного выражения, выражения-формы p_i – условия ветвей.
- `caar`, `cadr`, `cdar`, `cddr`, `caaar`, ..., `cdddar`, `cddddr` – функции от одного аргумента, являющиеся суперпозицией функций `car` и `cdr`;
- `list` – составляет список из значений своих аргументов (в отличие от `cons` является симметричной относительно своих аргументов);
- `+`, `-`, `*`, `/` – простые арифметические функции;
- `=`, `/=`, `<`, `>`, `<=`, `>=` – арифметические предикаты;
- `evenp` – проверка числа на четность;
- `null` – проверка атома на равенство `Nil`;
- `listp` – проверка на список;
- `numberp` и `symbolp` – проверка на числовой или символьный атом соответственно;

- not, and, or – логические отрицание, конъюнкция, дизъюнкция.