

*Государственное 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»

Отчет по лабораторной работе №5

по курсу ФиЛП

Студент

(Подпись, дата) Якубаускайте М. А.

Преподаватель

(Подпись, дата) Толпинская Н.Б.

Москва 2019

Теоритическая часть

Атом. Представление в памяти

Атом - символ, идентификатор - синтаксически - набор литер (букв латинского алфавита, цифр), начинающихся с букв; специальные символы T, Nil - используются для обозначения логических констант; самоопределяемые атомы - натуральные, дробные числа, вещественные числа, строки - последовательность символов, заключенная в двойные апострофы. У каждого символьного атома есть окружение и время жизни, символьные атомы не дублируются и в памяти представлены в виде 5 указателей:

- а) имя - name. В таблице символьных атомов хранится его внешнее имя, строка, представляющая его при вводе, выводе;
- б) значение - value. В таблице символьных атомов хранится его значение, как аргумента функции, фактическое значение, которое с ним связано, при его использовании в роли формального параметра некоторой функции;
- в) функция - function. В таблице символьных атомов хранится его функциональное значение, определяющее выражение функции при его использовании в качестве имени этой функции;
- г) свойство - properties. В таблице символьных атомов хранится список свойств атома за исключением внешнего имени, остальные значения могут отсутствовать;
- д) пакет - package - логические пакеты - таблицы, отображающие имена в символы.

Локальное и глобальное определение значения атома.

Lisp - интерпритатор в ходе своей работы поддерживает специальную таблицу символьных атомов - таблицу символов, в которой хранится информация обо всех атомах, встретившихся в тексте интерпритируемой программы или используемых в качестве обрабатываемых данных. Для присваивания рабочей переменной значения применяется форма SET. Чтобы присвоить переменной pi значение 22/7:

| | |
|---|-----------------------|
| 1 | (SET (QUOTE PI) 22/7) |
| 2 | (SETQ PI 22/7) |

В общем случае атом имеет несколько разных значений. Они независимы друг от друга, один и тот же символьный атом может быть использован как имя функции и как имя формального параметра. Нужная его интерпретация обеспечивается, исходя из контекста его применения. В таблице символьных атомов для каждого атома хранятся не сами значения, а указатели на внутренние представления этих значений. Внутренним представлением символьного атома является указатель на соответствующий элемент таблицы атомов.

Глобальные атомы определяются в глобальной области видимости. Локальные атомы определяются в области видимости функции.

EVAL - функция, которая принудительно вычисляет значения, обеспечивая дополнительный вызов интерпретатора LISP, вызов может производиться внутри вычисляемого S-выражения, позволяет снять блокировку QUOTE.

QUOTE - специальная одноаргументная функция, которая блокирует вычисления и возвращает в качестве значения этот аргумент.

Практическая часть

```
1
2 (SETF a 2)
3 (SETF B 3)
4 (SETF C 5)
5
6 A => 2
7 B => 3
8 C => 5
9
10 'A => A
11 '(+ A C) => (+ A C)
12 (A) => EVAL: undefined function A
13 (EVAL 'A) => 2
14
15 (DEFUN A() 'B)
16 (DEFUN B() 4)
17
18 A => 2
19 B => 3
20 C => 5
21 'A => A
22 '(+ A C) => (+ A C)
23 (A) => B
24 (+ A A) => 4
25 (+ (B) B) => 7
26 (B B B) => EVAL/APPLY: too many arguments given to B
27
28 (SETF A B)
29
30 A => 3
31 B => 3
32 C => 5
33 'A => A
34 '(+ A C) => (+ A C)
35 (A) => B
```

```

36 (EVAL 'A) => 3
37
38 (SETF C B)
39
40 A => 3
41 B => 3
42 C => 3
43 'A => A
44 '(+ A C) => (+ A C)
45 (A) => B
46 (EVAL 'A) => 3
47
48 (DEFUN A(X Y) (+ X Y))
49
50 A => 3
51 B => 3
52 C => 3
53 'A => A
54 '(+ A C) => (+ A C)
55 (A) => EVAL/APPLY: Too few arguments (0 instead of at least 2) given to A
56 (EVAL 'A) => 3
57 (A A A) => 6
58 (A B A) => 6
59
60 (DEFUN B(X Y) (SETF B (+ B 1)) X Y B))A => 3B => 3(B B B) => 36

```

Функция, которая вычисляет катет по гипотенузе и другому катету.

```
(DEFUN CATET(C1 G) (SQRT (-(*G G)(*C1 C1))))
```