



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 №3

Название: _____ Работа интерпретатора Lisp

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 Базис Lisp

Базис языка — минимальный набор конструкций языка и структур данных, позволяющих решить любую задачу.

К базису языка Lisp относятся:

- атомы;
- бинарные узлы;
- встроенные функции — atom, eq, car, cdr, cons;
- специальные функции и функционалы — cond, quote, eval, lambda.

1.2 Классификация функций

Функции можно классифицировать с точки зрения организации.

1. Чистые функции — «чистые математические» функции (базис).
2. Рекурсивные функции — основной принцип организации повторных вычислений.
3. Специальные функции или формы — могут иметь переменное количество аргументов или вариативную их обработку.
4. Псевдофункции — создание каких-либо эффектов на экране.
5. Функции с вариантами значений.
6. Функционалы (функции высших порядков) — в качестве аргументов используют функции или возвращают их в качестве результата.

7. Базисные функции — минимальный набор функций, позволяющих решить любую задачу.

Также базисные и функции ядра можно классифицировать с точки зрения действий.

1. Селекторы — переходят по соответствующему указателю списковой ячейки.
2. Конструкторы — создают структуры данных.
3. Предикаты — позволяют классифицировать или сравнивать структуры.

1.3 Способы создания функций

1. С помощью `lambda`. После ключевого слова указывается лямбда-список и тело функции.

```
1 (lambda (x y) (+ x y))
```

Для применения используются лямбда-выражения.

```
1 ((lambda (x y) (+ x y)) 1 2)
```

2. С помощью `defun`. Используется для неоднократного применения функции (в том числе рекурсивного вызова).

```
1 (defun sum (x y) (+ x y))  
2 (sum 1 2)
```

1.4 Работа функций `cond`, `if`, `and/or`

В языке Lisp для организации управления вычислительным процессом используются управляющие структуры. Внешне они схожи на вызов функции, отличие в использовании аргументов.

COND

```
1 (cond (test1 value1)
2       (test2 value2)
3       ...
4       (testn valuen))
```

cond последовательно (слева направо) вычисляет значения `testi` ($i=1..n$), пока не получит значение, отличное от `Nil`. Как только оно встречено, результатом будет являться `valuei`. Если все `testi` равны `Nil`, возвращает `Nil`.

IF

```
1 (if test t_body f_body)
```

if вычисляет значение `test`; если оно отлично от `Nil`, выполняется `t_body`, иначе – `f_body`.

AND/OR

```
1 (and arg1 arg2 ... argn)
2 (or arg1 arg2 ... argn)
```

and последовательно вычисляет значения `argi` ($i=1..n$), пока не встретится `Nil`. Если такое значение встретилось, то функция возвращает `Nil`, иначе – последнее вычисленное значение.

or последовательно вычисляет значение `argi` ($i=1..n$), пока не встретится значение, отличное от `Nil`. Если такое встретилос, то функция возвращает последнее вычисленное значение, иначе – `Nil`.