



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 №2

Название: \_\_\_\_\_ Определение функций пользователя

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 Функции `Car` и `Cdr`

Функции `Car` и `Cdr` являются базовыми функциями работы со списками. В качестве аргументов им передается точечная пара или список.

`Car` возвращает голову (первый элемент) списка, `Cdr` — хвост (все элементы, кроме первого).

Если входной аргумент является пустым списком, обе функции возвращают Nil. Если в списке 1 элемент, Car вернет голову, а Cdr – Nil.

## 1.5 Назначение и отличие в работе Cons и List

Cons — функция от двух аргументов. Создает списковую ячейку и составляет 2 указателя – на голову и на хвост – на входные аргументы.

List — функция от произвольного числа аргументов, при этом все они вычисляются. Строит новый список, первым элементом которого является значение первого аргумента, хвостом – значение второго аргумента.

```
1 (cons '(A) '(B)) ;; ((A) B)
2 (list '(A) '(B)) ;; ((A) (B))
```