



КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Преподаватель Толпинская Н. Б.

Москва — 2021 г.

# Оглавление

<b>1</b>	<b>Задания</b>	<b>2</b>
1.1	Используя только функции <code>CAR</code> и <code>CDR</code> , написать выражения, возвращающие: (№2) . . . . .	2
1.1.1	Второй элемент . . . . .	2
1.1.2	Третий элемент . . . . .	2
1.1.3	Четвертый элемент . . . . .	2
1.2	Что будет в результате вычисления выражений? (№3) . . . .	2
1.3	Что будет в результате вычисления выражений? (№4) . . . .	3
1.4	Написать функцию, возвращающую список... (№5) . . . . .	6
<b>2</b>	<b>Ответы на вопросы к лабораторной работе</b>	<b>7</b>
2.1	Классификация функций . . . . .	7
2.2	Базис . . . . .	7
2.3	Как выполняются <code>car/cdr</code> , какие результаты вернут (примеры) . . . . .	8
2.4	Отличие <code>list</code> от <code>cons</code> . . . . .	9

# 1 Задания

## 1.1 Используя только функции CAR и CDR, написать выражения, возвращающие: (№2)

### 1.1.1 Второй элемент

```
1 (car (cdr '(1 2 3 4)))
```

### 1.1.2 Третий элемент

```
1 (car (cdr (cdr '(1 2 3 4))))
```

### 1.1.3 Четвертый элемент

```
1 (car (cdr (cdr (cdr '(1 2 3 4)))))
```

## 1.2 Что будет в результате вычисления выражений? (№3)

```
1 (CAADR '((blue cube) (red pyramid)))
```

Результат: red.

```
1 (CDAR '((abc) (def) (ghi)))
```

Результат: Nil

```
1 (CADR '((abc) (def) (ghi)))
```

Результат: (def)

```
1 (CADDR '((abc) (def) (ghi)))
```

Результат: (ghi)

### 1.3 Что будет в результате вычисления выражений? (№4)

```
1 (list 'Fred 'and Wilma)
```

Результат: ошибка (несвязная переменная)

```
1 (list 'Fred '(and Wilma))
```

Результат: (Fred (and Wilma))

```
1 (cons Nil Nil)
```

Результат: (Nil)

```
1 (cons T Nil)
```

Результат: (T)

```
1 (cons Nil T)
```

Результат: (Nil . T)

```
1 (list Nil)
```

Результат: (Nil)

```
1 (list Nil)
```

Результат: (Nil)

```
1 (cons (T) Nil)
```

Результат: ошибка (функция T не определена)

```
1 (list '(one two) '(free temp))
```

Результат: ((one two) (free temp))

```
1 (cons 'Fred '(and Wilma))
```

Результат: (Fred and Wilma)

```
1 (cons 'Fred '(Wilma))
```

Результат: (Fred Wilma)

```
1 (list Nil Nil)
```

Результат: (Nil Nil)

```
1 (list T Nil)
```

Результат: (T Nil)

```
1 (list Nil T)
```

Результат: (Nil T)

```
1 (cons T (list Nil))
```

Результат: (T Nil)

```
1 (list (T) Nil)
```

Результат: ошибка (функция T не определена)

```
1 (cons '(one two) '(free temp))
```

Результат: ((one two) free temp)

## 1.4 Написать функцию, возвращающую список... (№5)

Функция (f ar1 ar2 ar3 ar4), возвращающая ((ar1 ar2) (ar3 ar4)):

С помощью функции list:

```
1 (defun fl1(ar1 ar2 ar3 ar4) (list (list ar1 ar2) (list ar3 ar4)))
```

С помощью функции cons:

```
1 (defun fc1(ar1 ar2 ar3 ar4) (cons  
2 (cons ar1 (cons ar2 Nil))  
3 (cons  
4 (cons ar3 (cons ar4 Nil))  
5 ) Nil)  
6 ))
```

Функция (f ar1 ar2), возвращающая ((ar1) (ar2)):

С помощью функции list:

```
1 (defun fl2(ar1 ar2) (list (list ar1) (list ar2)))
```

С помощью функции cons:

```
1 (defun fc2(ar1 ar2) (cons  
2 (cons ar1 Nil)  
3 (cons (cons ar2 Nil) Nil)  
4 ))
```

Функция (f ar1), возвращающая (ar1):

С помощью функции list:

```
1 (defun fl3(ar1) (list (list (list ar1))))
```

С помощью функции cons:

```
1 (defun fc3(ar1) (cons (cons (cons ar1 Nil) Nil) Nil))
```

Результаты в виде списочных ячеек приложены к отчету на отдельном листе.

## 2 Ответы на вопросы к лабораторной работе

### 2.1 Классификация функций

Функции в Lisp классифицируют следующим образом:

- чистые математические функции;
- рекурсивные функции;
- специальные функции — формы (сегодня 2 аргумента, завтра - 5);
- псевдофункции (создают эффект на внешнем устройстве);
- функции с вариативными значениями, из которых выбирается 1;
- функции высших порядков — функционал: используется для синтаксического управления программ (абстракция языка).

По назначению функции разделяются следующим образом:

1. конструкторы — создают значение (`cons`, например);
2. селекторы — получают доступ по адресу (`car`, `cdr`);
3. предикаты — возвращают `Nil`, `T`.

### 2.2 Базис

Базис состоит из:

1. структуры, атомы;
2. встроенные (примитивные) функции (`atom`, `eq`, `cons`, `car`, `cdr`);
3. специальные функции, управляющие обработкой структур, представляющих вычислимые выражения (`quote`, `cond`, `lambda`, `label`, `eval`).



Таким образом, функции, входящие в базис:

- `atom` — функция определения, является ли объект атомом (возвращает `T`, если да, иначе — `Nil`);
- `eq` — функция проверки атомов на равенство (возвращает `T`, если равны, иначе — `Nil`)
- `cons` — функция создания точечной пары;
- `car` — функция получения первого элемента точечной пары;
- `cdr` — функция получения второго элемента точечной пары;
- `cond` — функция, позволяющая организовать ветвление (чаще всего используется в частичных функциях и имеет вид: `(cond (p1 e1) (p2 e2) ... (pn en))`, где `pi` — предикат, а `ei` — форма, соответствующая предикату `ei`)
- `quote` — функция блокировки вычислений, аргумент не вычисляется;
- `eval` — функция-интерпретатор, противоположность функции `quote`, пытается вычислить аргумент;
- `lambda`;
- `label`.

## 2.3 Как выполняются `car/cdr`, какие результаты вернут (примеры)

`car` — функция получения первого элемента точечной пары.

Примеры:

S-выражение	Результат выполнения <code>car</code>
<code>(A . B)</code>	<code>A</code>
<code>((A . B) . C)</code>	<code>(A . B)</code>
<code>A</code>	ошибка

`cdr` — функция получения второго элемента точечной пары.

S-выражение	Результат выполнения <code>cdr</code>
<code>(A . B)</code>	<code>B</code>
<code>(A . (B . C))</code>	<code>(B . C)</code>
<code>A</code>	ошибка

## 2.4 Отличие `list` от `cons`

`cons` — функция конструирования точечной пары, на вход получает 2 значения и делает из них точечную пару.

`list` — функция конструирования списка. На вход получает произвольное количество элементов и делает из них список.

Вызовы `(list 1 2 3 4)` и `(cons 1 (cons 2 (cons 3 (cons 4 Nil))))` эквивалентны, то есть дают одинаковый результат.