



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 Lisp. Использование стандартных функций.

Дисциплина: _____ Функциональное и логическое программирование

Студент	ИУ7-66Б	_____	А.Д. Ковель
	Группа	Подпись, дата	И. О. Фамилия

Преподаватель	_____	Н. Б. Толпинская
	Подпись, дата	И. О. Фамилия

Москва, 2023 г.

1 Теоретические вопросы

1. *Элементы языка: определение, синтаксис, представление в памяти.*

Лисп — функциональный язык программирования, данные и программы в котором представляются системами линейных списков символов. Программы в Лиспе понимают как применение функции к ее аргументам (вызов функции). Аргументом функции может быть любая форма Lisp.

2. *Особенности языка Lisp. Структура программы. Символ апостроф.*

Язык поддерживает функциональную методологию. Является бестиповым регистронезависимым языком.

Функциональная программа состоит из набора взаимосвязанных функций. Каждая функция определяется выражением, которое задает правило вычисления её значения в зависимости от значений ее аргументов и которое является композицией встроенных функций, а также других функций, описанных в программе.

Quote, или оператор цитирования — это специальный оператор, то есть, у него есть собственное правило вычисления: quote берет один аргумент и возвращает его текстовую запись.

3. *Базис языка Lisp. Ядро языка.*

Базис языка Lisp:

- а) атомы и структуры (представляющиеся бинарными узлами);
- б) несколько базовых функций и функционалов: встроенные — примитивные функции (atom, eq, cons, car, cdr); специальные функции и функционалы (quote, cond, lambda, eval, apply, funcall).

Ядро языка — набор частоиспользуемых функций, расширяющих базис. Например, функция list или defun.

2 Практические задания

1. Представить следующие списки в виде списочных ячеек:

- | | |
|--|-------------------------------|
| 1) '(open_close_half) | 2) '((open1)(close2)(half3)) |
| 3) '((one)_for_all_(and_(me_(for_you)))) | 4) '((TOOL)_call) |
| 5) '(((TOOL1)((call2))((shell)))) | 6) '(((TOOL)_call)_((shell))) |

Решение представлено на отдельном листе и приложено к отчету.

2. Используя только функции CAR и CDR написать выражения, возвращающие:

- | | | |
|------------|------------|--|
| 1) второй; | 2) третий; | 3) четвертый элементы
заданного списка. |
|------------|------------|--|

Решения:

- | |
|--------------------------------------|
| 1 (car (cdr '(a_b_c_d))) |
| 2 (car_(cdr_(cdr_(a b c d)))) |
| 3 (car (cdr (cdr (cdr '(a_b_c_d))))) |

3. Что будет в результате вычисления выражений?

- | | |
|--|-------------------------------|
| 1) (caadr '((blue_cube)_red_pyramid))) | 2) (cdar '((abc)(def)(ghi))) |
| 3) (cadr '((abc)(def)(ghi))) | 4) (caddr '((abc)(def)(ghi))) |

Решения:

- | | |
|----------|----------|
| 1) RED | 2) Nil |
| 3) (DEF) | 4) (GHI) |

4. Напишите результат выражений и объясните как он получен:

- | | | |
|-----------------------------------|-----------------------------|----------------------------------|
| 1) (list 'Fred_'and 'Wilma) | 2) (list 'Fred_(and Wilma)) | 3) (cons Nil Nil) |
| 4) (cons T Nil) | 5) (cons Nil T) | 6) (list Nil) |
| 7) (list Nil) | 8) (cons '(T)_Nil) | 9) (list '(one_two)_(free temp)) |
| 10) (cons 'Fred_(and Wilma)) | 11) (cons 'Fred_(Wilma)) | 12) (list Nil Nil) |
| 13) (list Nil T) | 14) (cons T (list Nil)) | 15) (list '(T)_Nil) |
| 16) (cons '(one_two)_(free temp)) | | |

Решения:

- | | | |
|-----------------------------|-----------------------|---------------------------|
| 1) (FRED AND WILMA) | 2) (FRED (AND WILMA)) | 3) (Nil) |
| 4) (T) | 5) (NIL.T) | 6) (NIL) |
| 7) (<code>list</code> Nil) | 8) ((T)) | 9) ((ONE TWO)(FREE TEMP)) |
| 10) (FRED AND WILMA) | 11) (FRED WILMA) | 12) (NIL NIL) |
| 13) (NIL T) | 14) (T NIL) | 15) ((T) Nil) |
| 16) ((ONE TWO)FREE TEMP) | | |

5. Написать лямбда-выражение и соответствующую функцию:

- 1) Написать функцию $f(ar1\ ar2\ ar3\ ar4)$, возвращающую $((ar1\ ar2)\ (ar3\ ar4))$.
- 2) Написать функцию $f(ar1\ ar2)$, возвращающую $((ar1)(ar2))$.
- 3) Написать функцию $f(ar1)$, возвращающую $((ar1))$

Представить результаты в виде списочных ячеек.

Решения.

1. Функция:

```
1 (defun f(ar1 ar2 ar3 ar4)(cons (list ar1 ar2) (list (list ar3  
ar4))))
```

Лямбда-выражение:

```
1 (lambda (ar1 ar2 ar3 ar4)(cons (list ar1 ar2) (list (list ar3  
ar4))))
```

2. Функция:

```
1 (defun f(ar1 ar2)(list(cons ar1 nil)(cons ar2 nil)))
```

Лямбда-выражение:

```
1 (lambda (ar1 ar2)(list(cons ar1 nil)(cons ar2 nil)))
```

3. Функция:

```
1 (defun f(ar1)(list (cons ar1 nil)))
```

Лямбда-выражение:

```
1 (lambda (ar1)(list (cons ar1 nil)))
```

Представление в виде списочных ячеек выполнено на отдельном листе и приложено к отчету.