



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»

Лабораторная работа №2 по дисциплине "Функциональное и логическое программирование"

Тема Определение функций пользователя

Студент Светличная А.А.

Группа ИУ7-53Б

Преподаватель Строганов Ю.В., Толпинская Н.Б.

Москва — 2023 г.

Оглавление

1	Теоритические вопросы	3
1.1	Базис Lisp	3
1.2	Классификация функций	3
1.3	Способы создания функций	4
1.4	Функции Car и Cdr	4
1.5	Функции eq, eql, equal, equalp	5
2	Практические задания	6
2.1	Задание №1	6
2.2	Задание №2	6
2.3	Задание №3	6
2.4	Задание №4	7
2.5	Задание №5	7
2.6	Задание №6	8
2.7	Задание №7	8
2.8	Задание №8	9

1 Теоритические вопросы

1.1 Базис Lisp

Базис языка — минимальный набор конструкций языка и структур данных, с помощью которых можно решить любую задачу.

Базис языка Lisp содержит:

- атомы и структуры;
- базовые функции и функционалы:
 - встроенные — примитивные функции (atom, eq, cons, car, cdr);
 - специальные функции и функционалы (quote, cond, lambda, eval, apply, funcall).

1.2 Классификация функций

Функции в Lisp классифицируют следующим образом:

- чистые математические функции (имеют фиксированное количество аргументов и один результат);
- специальные функции — формы (прнимают произвольное число аргументов или по разному обрабатывают аргументы);
- функции высших порядков — функционалы (используются для создания синтаксически управляемых программ).

По базисные функции разделяются следующим образом:

- конструкторы — создают значение (cons, list);
- селекторы — получают доступ по адресу (car, cdr);
- предикаты — возвращают Nil, T.

1.3 Способы создания функций

Функцией называется правило, по которому каждому значению одного или нескольких аргументов ставится в соответствие конкретное значение результата.

- В Lisp можно определить функцию без имени с помощью **λ-выражений**.
Lambda-определение безымянной функции:

```
(lambda <lambda-список> <форма>)
```

Lambda-вызов функции:

```
(<lambda-выражение> <формальные параметры>)
```

- Также в Lisp можно определить функцию с именем с помощью **defun**.
В таких функциях defun связывает символьный атом с Lambda-определением

```
(defun f <lambda-выражение>)
```

Упрощенное определение:

```
(defun f(arg1, ..., argN) <формы>)
```

1.4 Функции Car и Cdr

Функции **car**, **cdr** являются базовыми функциями доступа к данным.

- **car** принимает точечную пару или список в качестве аргумента и возвращает первый элемент или Nil, соответственно.
- **cdr** принимает точечную пару или список в качестве аргумента и возвращает все элементы кроме первого или Nil, соответственно.

1.5 Функции `eq`, `eq1`, `equal`, `equalp`

Функции `eq`, `eq1`, `equal`, `equalp` являются функциями сравнения.

- `eq` сравнивает символьные атомы.
- `eq1` сравнивает символьные атомы, числа одного типа.
- `equal` сравнивает символьные атомы, числа одного типа, списки.
- `equalp` сравнивает символьные атомы, числа разных типов, списки.

2 Практические задания

2.1 Задание №1

Составить диаграмму вычисления следующих выражений.

Листинг 2.1 – Условие задания №1

```
1 (equal 3 (abs - 3))  
2 (equal (+ 1 2) 3)  
3 (equal (* 4 7) 21)  
4 (equal (* 2 3) (+ 7 2))  
5 (equal (- 7 3) (* 3 2))  
6 (equal (abs (- 2 4)) 3))
```

2.2 Задание №2

Написать функцию, вычисляющую гипотенузу прямоугольного треугольника по заданным катетам и составить диаграмму её вычисления.

Листинг 2.2 – Выполнение задания №2

```
1 (defun f(a b)(sqrt (+ (* a a) (* b b))))  
2 (f 3 4)
```

2.3 Задание №3

Каковы результаты вычисления следующих выражений? (объяснить возможную ошибку и варианты ее устранения)

Листинг 2.3 – Условие задания №3

```
1 (list 'a c)           ;; variable C has no value
2 (cons 'a (b c))       ;; undefined function B
3 (cons 'a '(b c))      ;; (A B C)
4 (caddy (1 2 3 4 5))   ;; undefined function CADDY
5 (cons 'a 'b 'c)       ;; too many arguments given to CONS
6 (list 'a (b c))       ;; undefined function B
7 (list a '(b c))       ;; variable A has no value
8 (list (+ 1 '(length '(1 2 3)))) ;; (LENGTH '(1 2 3)) is not a
                                   number
```

Листинг 2.4 – Выполнение задания №3

```
1 (list 'a 'c)          ;; (A C)
2 (cons 'a '(b c))      ;; (A B C)
3 (cons 'a '(b c))      ;; (A B C)
4 (caddr '(1 2 3 4 5)) ;; 3
5 (cons 'a 'b)          ;; (A . B)
6 (list 'a '(b c))      ;; (A (B C))
7 (list 'a '(b c))      ;; (A (B C))
8 (list (+ 1 (length '(1 2 3)))) ;; (4)
```

2.4 Задание №4

Написать функцию `longer_then` от двух списков-аргументов, которая возвращает Т, если первый аргумент имеет большую длину.

Листинг 2.5 – Выполнение задания №4

```
1 (defun longer_then(list1 list2)(> (length list1) (length list2)))
```

2.5 Задание №5

Каковы результаты вычисления следующих выражений?

Листинг 2.6 – Выполнения задания №5

```
1 (cons 3 (list 5 6))           ;; (3 5 6)
2 (cons 3 '(list 5 6))          ;; (3 LIST 5 6)
3 (list 3 'from 9 'lives (- 9 3)) ;; (3 FROM 9 LIVES 6)
4 (+ (length for 2 too)) (car '(21 22 23)) ;; variable FOR has no
                                         value
5 (+ (length '(for 2 too)) (car '(21 22 23))) ;; 24
6 (cdr '(cons is short for ans)) ;; (IS SHORT FOR ANS)
7 (car (list one two))          ;; variable ONE has no value
8 (car (list 'one 'two))         ;; ONE
```

2.6 Задание №6

Дана функция (defun mystery (x) (list (second x) (first x))).
Какие результаты вычисления следующих выражений?

Листинг 2.7 – Условие задания №6

```
1 (mystery (one two))           ;; undefined function ONE
2 (mystery one 'two)            ;; variable ONE has no value
3 (mystery (last one two))      ;; variable ONE has no value
4 (mystery free)                ;; variable FREE has no value
```

Листинг 2.8 – Выполнения задания №6

```
1 (mystery '(one two))          ;; (TWO ONE)
2 (mystery '(one 'two))         ;; ('TWO ONE)
3 (mystery (last '(one two)))   ;; (NIL TWO)
4 (mystery '(free))             ;; (NIL FREE)
```

2.7 Задание №7

Написать функцию, которая переводит температуру в системе Фаренгейта температуру по Цельсию (defun f-to-c (temp)...).

Формулы: $c = 5/9 * (f - 32.0)$; $f = 9/5 * c + 32.0$.

Как бы назывался роман Р. Брэдли "451 по Фаренгейту" в системе по Цельсию?

Листинг 2.9 – Выполнение задания №7

```
1 (defun f_to_c(temp)(* (/ 5 9) (- temp 32.0)))  
2 (f_to_c 451) ;; 232.77779
```

2.8 Задание №8

Что получится при вычисления каждого из выражений?

Листинг 2.10 – Выполнение задания №8

```
1 (list 'cons t NIL)                ;; (CONS T NIL)  
2 (eval (list 'cons t NIL))         ;; (T)  
3 (eval (eval (list 'cons t NIL)))  ;; undefined function T  
4 (apply #'cons '(t NIL))           ;; (T)  
5 (eval NIL)                        ;; NIL  
6 (list 'eval NIL)                  ;; (EVAL NIL)  
7 (eval (list 'eval NIL))           ;; NIL
```