



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»

Отчет по лабораторной работе №2 по дисциплине "Функциональное и логическое программирование"

Тема Определение функций пользователя

Студент Малышев И. А.

Группа ИУ7-61Б

Оценка (баллы) _____

Преподаватель: Толпинская Н. Б.

Москва — 2022 г.

Теоретические вопросы

1. Базис Lisp

Базис языка представлен:

- структурами и атомами;
- функциями;

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

Функции, входящие в базис языка: atom, eq, cons, car, cdr, cond, quote, lambda, eval, apply, funcall.

2. Классификация функций

- "чистая" функция – соответствует математической функции;
- специальные функции или формы – могут принимать несколько аргументов;
- псевдофункции – создают «эффект» – отображение на экране процесса обработки данных и т.п.;
- функции с вариантами значения;
- функционалы – принимают в качестве аргумента функцию или возвращают функцию;
- базисные функции (cons, car, cdr, atom, cond, eq, quote, eval, lambda, apply, funcall).

3. Способы создание функций

Определение функций пользователя в Lisp-е возможно двумя способами:

- с использованием Лямбда-нотации (функции без имени): (lambda (<аргументы>) (<тело>));
- с использованием макро определения DEFUN: (defun <имя> (<аргументы>) (<тело>)).

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.

5. Назначение и отличие в работе Cons и List

Функции *list*, *cons* являются функциями создания списков (*cons* – базовая, *list* – нет). *cons* создает списковую ячейку и устанавливает два указателя на аргументы. *list* принимает переменное число аргументов и возвращает список, элементы которого – переданные в функцию аргументы.

Отличие в работе *cons* и *list* заключается в том, что *cons* создаёт одну списковую ячейку и устанавливает два указателя на аргументы, а *list* создаёт списковые ячейки для каждого аргумента, где голова – сам элемент списка, хвост – остальная часть списка.

Практические задания

1. Составить диаграмму вычисления следующих выражений:

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

Диаграмма вычисления оформлена на листе бумаги, прилагающийся к отчету.

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

Листинг 1: Решение задания №2

```
1 (defun hypot (a b) (sqrt (+ (* a a) (* b b))))
```

Диаграмма вычисления оформлена на листе бумаги, прилагающийся к отчету.

3. Написать функцию, вычисляющую объем параллелепипеда по 3-м его сторонам, и составить диаграмму ее вычисления.

Листинг 2: Решение задания №3

```
1 (defun volume (a b c) (* a b c))
```

Диаграмма вычисления оформлена на листе бумаги, прилагающийся к отчету.

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

```
(list 'a c) -> the variable c is unbound; (list 'a 'c) -> (a c)  
(cons 'a 'b 'c) -> invalid number of arguments; (cons 'a 'b) -> (a . b)  
(cons 'a (b c)) -> undefined function b; (cons 'a '(b c)) -> (a b c)  
(list 'a (b c)) -> undefined function b; (list 'a '(b c)) -> (a (b c))
```

```
(cons 'a '(b c)) -> (a b c)
(list a '(b c)) -> the variable a is unbound; (list 'a '(b c)) -> (a (b c))
(caddr (1 2 3 4 5)) -> 3
(list (+ 1 '(length '(1 2 3)))) -> (LENGTH '(1 2 3)) is not of type NUMBER; (list (+ 1 (length '(1 2 3)))) -> 4
```

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

Листинг 3: Решение задания №5

```
1 (defun longer_than (l1 l2) (> (length l1) (length l2)))
```

6. Каковы результаты вычисления следующих выражений?

```
(cons 3 (list 5 6)) -> (3 5 6)
(cons 3 '(list 5 6)) -> (3 list 5 6)
(list 3 'from 9 'lives (- 9 3)) -> (3 from 9 lives 6)
(+ (length for 2 too)) (car '(21 22 23)) -> illegal function call; (+ (length '(for 2 too)) (car '(21 22 23))) -> 24
(cdr '(cons is short for ans)) -> (is short for ans)
(car (list one two)) -> the variable one is unbound; (car (list 'one 'two)) -> one
(car (list 'one 'two)) -> one
```

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

```
(mystery (one two)) -> undefined function one; (mystery '(one two)) -> (two one)
(mystery one 'two) -> invalid number of arguments; (mystery '(one two)) -> (two one)
(mystery (last one two)) -> undefined function last; (mystery '(last one two)) -> (one last)
(mystery free) -> unbound variable free; (mystery '(free)) -> (NIL free)
```

8. 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.Брэдбери "+451 по Фаренгейту" в системе по Цельсию?

Листинг 4: Решение задания №8

```
1 (defun f-to-c (temp) (* (/ 5 9) (- temp 32.0)))
```

Роман Р.Брэдбери "+451 по Фаренгейту" в системе по Цельсию назывался бы "+232.778 по Цельсию".

9. Что получится при вычисления каждого из выражений?

```
(list 'cons t NIL) -> (cons T NIL)
(eval (list 'cons t NIL)) -> (T)
(eval (eval (list 'cons t NIL))) -> The function T is undefined
```

Дополнительно

1. Написать функцию, вычисляющую катет по заданной гипотенузе и другому катету прямоугольного треугольника, и составить диаграмму ее вычисления.

Листинг 5: Решение задания

```
1 (defun get_cathetus (c a) (sqrt (- (* c c) (* a a))))
```

Диаграмма вычисления оформлена на листе бумаги, прилагающийся к отчету.

2. Написать функцию, вычисляющую площадь трапеции по ее основаниям и высоте, и составить диаграмму ее вычисления.

Листинг 6: Решение задания

```
1 (defun trapezium_area (a b h) (* (/ (+ a b) 2) h))
```

Диаграмма вычисления оформлена на листе бумаги, прилагающийся к отчету.