



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 Егорова П.А.

Группа ИУ7-64Б

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

Москва — 2023 г.

## Задание 1

Составить диаграмму вычисления следующих выражений:

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)

## Решение

Диаграммы оформлены на тетрадном листке бумаги и прилагаются к отчёту.

## Задание 2

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

## Решение

Листинг 1: Решение задания №2

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

Диаграмма оформлена на тетрадном листке бумаги и прилагается к отчёту.

## Задание 3

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

## Решение

Листинг 2: Решение задания №3

```
1 (list 'a c) → C IS UNBOUND; (list 'a 'c) → (A C)
2 (cons 'a (b c)); undefined function b, C IS UNBOUND;
3 (cons 'a '(b c)) → (A B C)
4 (caddr (1 2 3 4 5)) → 3
5 (cons 'a 'b 'c); INVALID NUMBER OF ARGUMENTS; (cons 'a '(b c)) → (a b c)
6 (list 'a (b c)); undefined function b, C IS UNBOUND; (list 'a '(b c)) → (A
  (B C))
```

```

7 (list a '(b c)); A IS UNBOUND; (list 'a '(b c)) → (A (B C))
8 (list (+ 1 '(length '(1 2 3)))) ; (LENGTH '(1 2 3)) is not of type NUMBER; (
    list (+ 1 (length '(1 2 3)))) → (4)

```

## Задание 4

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

## Решение

Листинг 3: Решение задания №4

```

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

```

## Задание 5

Каковы результаты вычисления следующих выражений?

## Решение

Листинг 4: Решение 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 'gives (- 9 3)) → (3 FROM 9 GIVES 6)
4 (+ (length for 2 too) (car '(21 22 23))) → FOR is unbound; (+ (length '(
    for 2 too)) (car '(21 22 23))) → 24
5 (cdr '(cons is short for ans)) → (IS SHORT FOR ANS)
6 (car (list one two)) → one, two are unbound;
7 (car (list 'one 'two)); → one

```

## Задание 6

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

## Решение

Листинг 5: Решение задания №6

```

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

```

## Задание 7

Написать функцию, которая переводит температуру в системе Фаренгейта в температуру по Цельсию (`defun f-to-c (temp)...`). Формулы:  $c = 5/9 * (f - 320)$ ;  $f = 9/5 * c + 32.0$ . Как бы назывался роман Р.Брэдли "451 по Фаренгейту" в системе по Цельсию?

### Решение

Листинг 6: Решение задания №7

```
1 (defun f-to-c (temp) (* 5/9 (- temp 320)))
```

Роман бы назывался "+73 по Цельсию" (с учётом округления числа  $655/9$ ).

## Задание 8

Что получится при вычисления каждого из выражений?

### Решение

Листинг 7: Решение задания №8

```
1 (list 'cons t NIL) -> (cons T NIL)
2 (eval (list 'cons t NIL)) -> (eval (cons T NIL)) -> (T)
3 (eval (eval (list 'cons t NIL))) -> (eval (T)) -> undefined function T
4 (apply #cons "(t NIL)) -> illegal complex number format: #C~S; (apply #'cons
   '(t NIL)) -> (T)
5 (eval NIL) -> NIL
6 (list 'eval NIL) -> (eval NIL)
7 (eval (list 'eval NIL)) -> NIL
```

## Контрольные вопросы

**Вопрос 1.** Базис языка Lisp.

**Ответ.** Базис языка образуют атомы, структуры (точечные пары и списки), базовые функции, базовые функционалы (функции, аргументами и значением которых являются функции).

**Вопрос 2.** Классификация функций языка Lisp.

**Ответ.**

- чистые (с фиксированным количеством аргументов) математические функции;
- рекурсивные функции;
- специальные функции – формы (принимают произвольное количество аргументов);
- псевдофункции (создающие «эффект», например, на экране);

- функционалы.

**Вопрос 3.** Способы создания функций.

**Ответ.** Функцию можно определить с помощью **defun** или **lambda**. (**defun** имя\_функции (список\_аргументов) тело\_функции).

**Вопрос 4.** Функции **car**, **cdr**, **eq**, **eql**, **equal**, **equalp**.

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

*car* принимает точечную пару или список в качестве аргумента и возвращает первый элемент (если список пустой, то *Nil*).

*cdr* принимает точечную пару или список в качестве аргумента и возвращает все элементы, кроме первого или *Nil*.

*eq* принимает на вход два аргумента и возвращает Т тогда и только тогда, когда аргументы являются идентичными объектами.

*eql* принимает на вход два аргумента и возвращает Т, если это числа одинакового типа с одинаковыми значениями, одинаковые символы или они равны через *eq*.

*equal* принимает на вход два аргумента и возвращает Т, если одинаково их выводимое представление.

*equalp* принимает на вход два аргумента и возвращает Т, если они равны через *equal* или равны, игнорируя регистры для строк и тип переменной для чисел.

**Вопрос 5.** Назначение и отличие в работе **cons** и **list**.

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