



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 (ИУ7)

**ОТЧЕТ**

по лабораторной работе № 3

Название: Функция eval

Дисциплина: Функциональное и логическое программирование

Студент

ИУ7-63Б

(Группа)

\_\_\_\_\_  
(Подпись, дата)

В.П. Федоров

(И.О. Фамилия)

Преподаватель

\_\_\_\_\_  
(Подпись, дата)

Н.Б. Толпинская

(И.О. Фамилия)

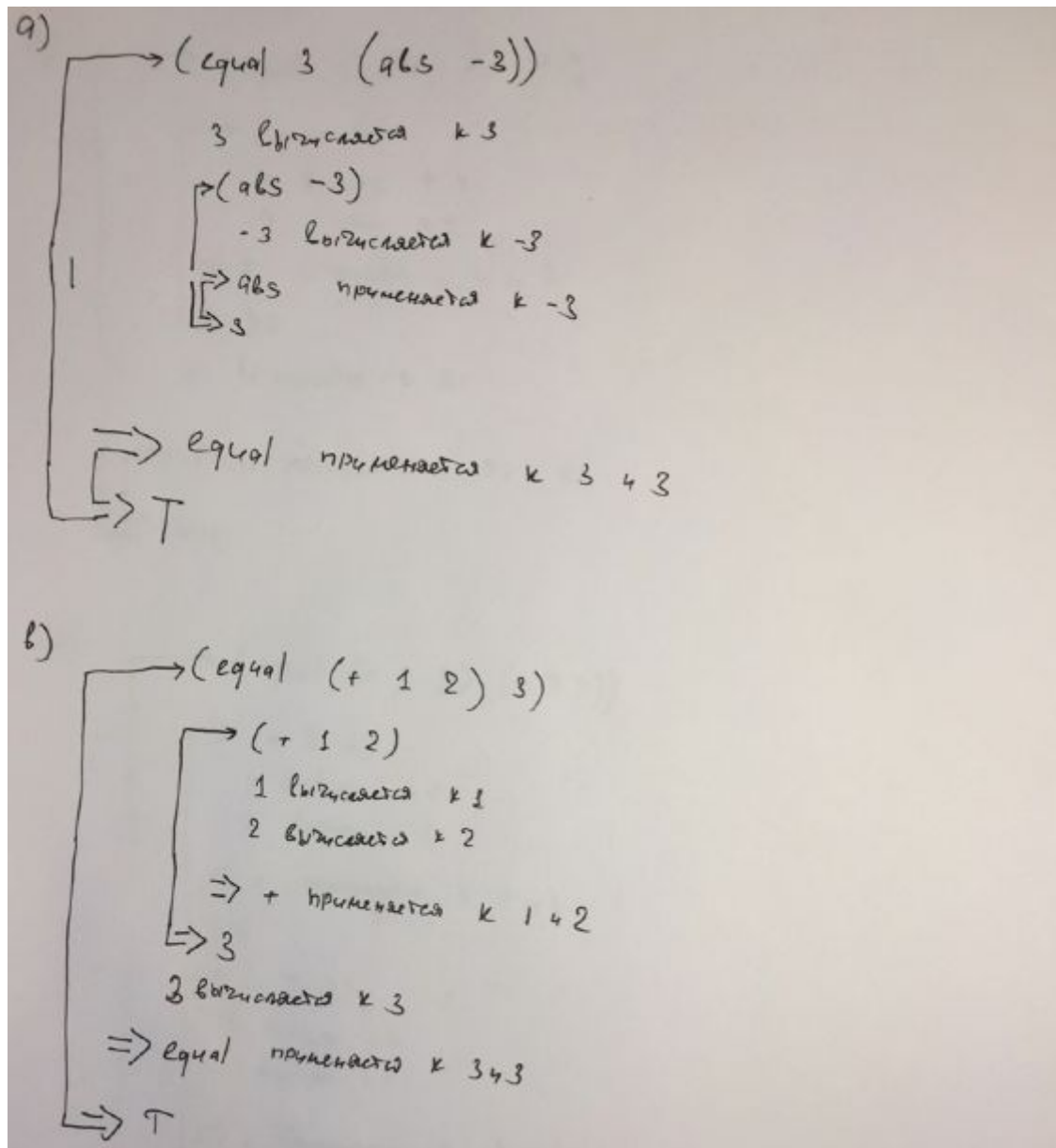
Москва, 2021

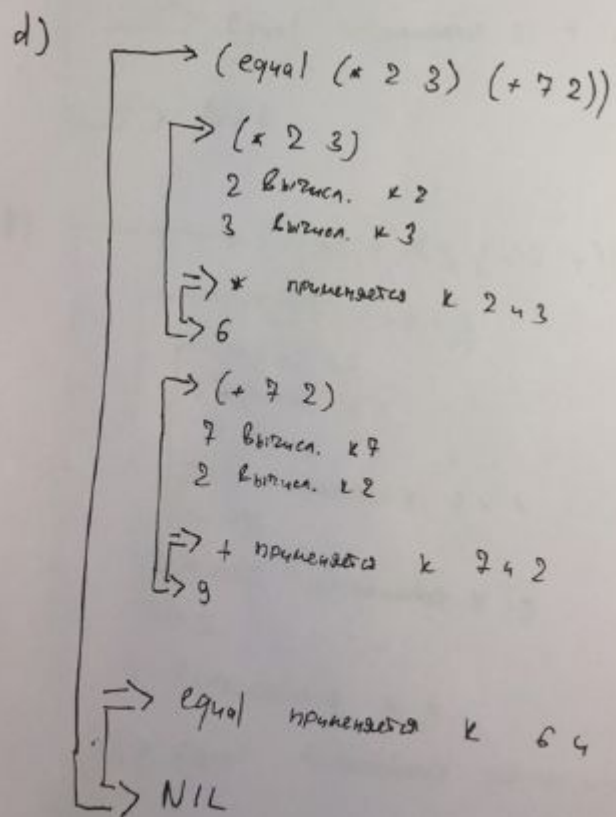
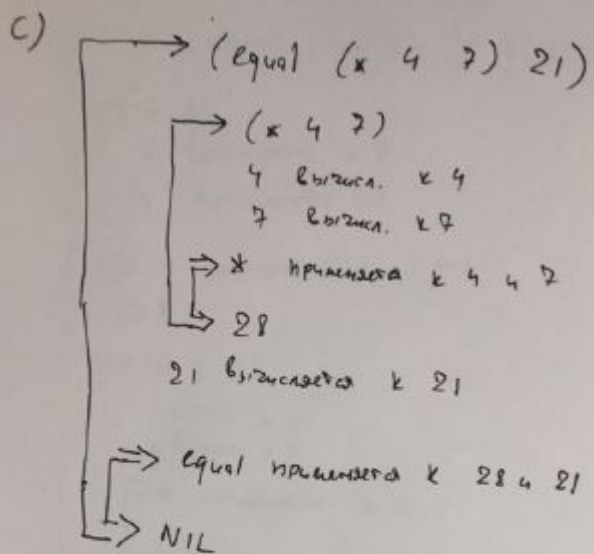
### Цели работы:

- изучить правила работы функции eval на структурированных списках-аргументах, построить диаграммы работы функции eval (выполнения функций);
- проанализировать работу функции eval при обработке стандартных функций и функций, определенных пользователем.

### Задача 1.

Диаграммы состояний.





e)  $\rightarrow (eq491 (-7 3) (x 3 2))$

$\rightarrow (-7 3)$

7 вычисляется к 2

3 вычисляется к 3

$\rightarrow -$  применяется к 7 4 3  
 $\rightarrow 4$

$\rightarrow (x 3 2)$

3 вычисл. к 5

2 вычисл. к 2

$\rightarrow *$  применяется к 3 4 2  
 $\rightarrow 6$

$\rightarrow eq491$  применяется к 4 и 6

$\rightarrow NIL$

f)  $\rightarrow (eq491 (abs (-2 4)) 3))$

$\rightarrow (abs (-2 4))$

$\rightarrow (-2 4)$

2 вычисл. к 2

4 вычисл. к 4

$\rightarrow -$  применяется к 2 и 4  
 $\rightarrow -2$

$\rightarrow abs$  применяется к -2  
 $\rightarrow 2$

3 вычисляется к 3

$\rightarrow eq491$  ~~вычисляется~~ применяется к 2 4 3

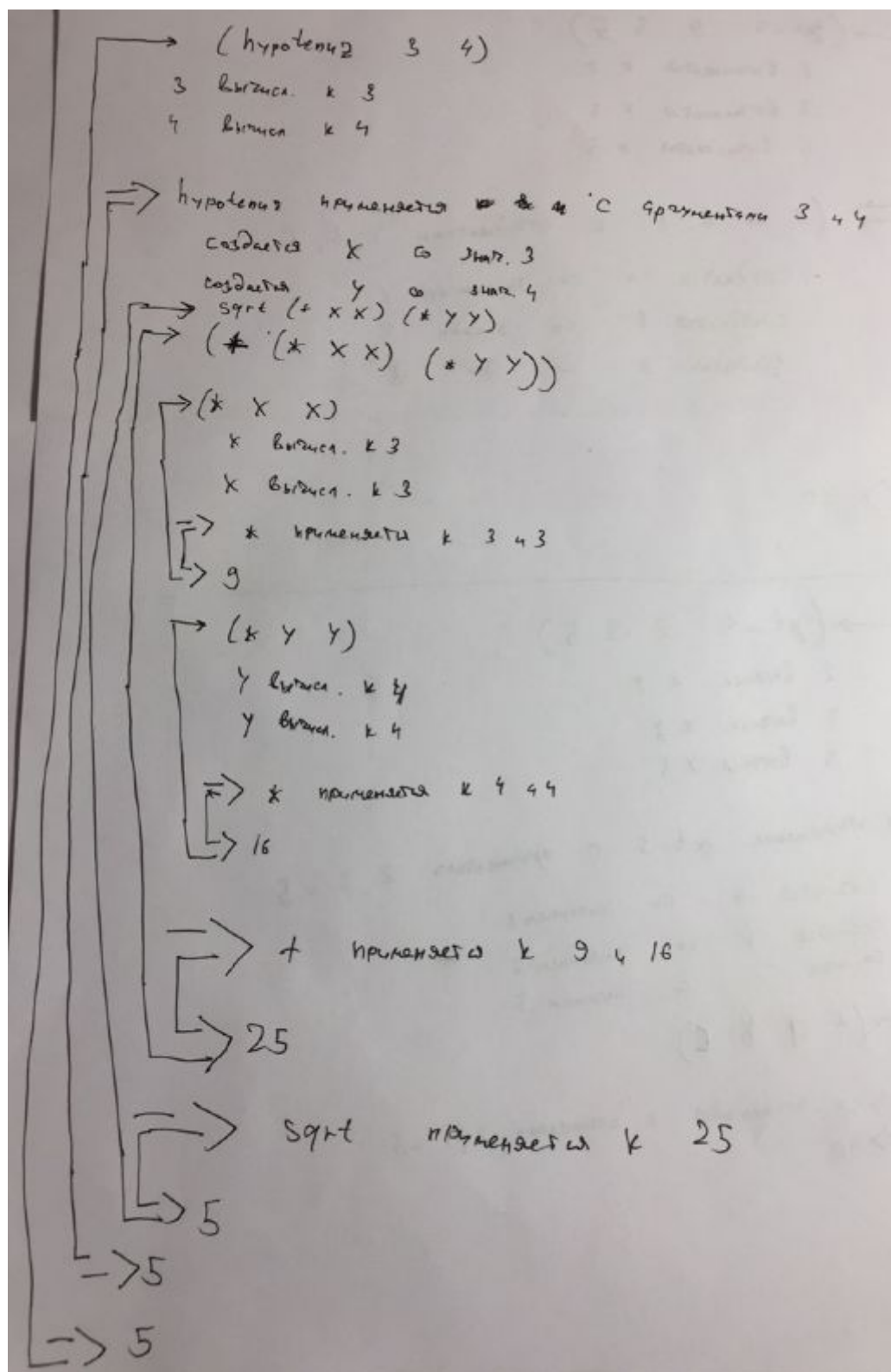
$\rightarrow NIL$

## Задача 2.

Написать функцию, вычисляющую длину гипотенузы прямоугольного треугольника по двум его сторонам

Листинг 2.1: Функция, вычисляющая гипотенузу, на языке LISP

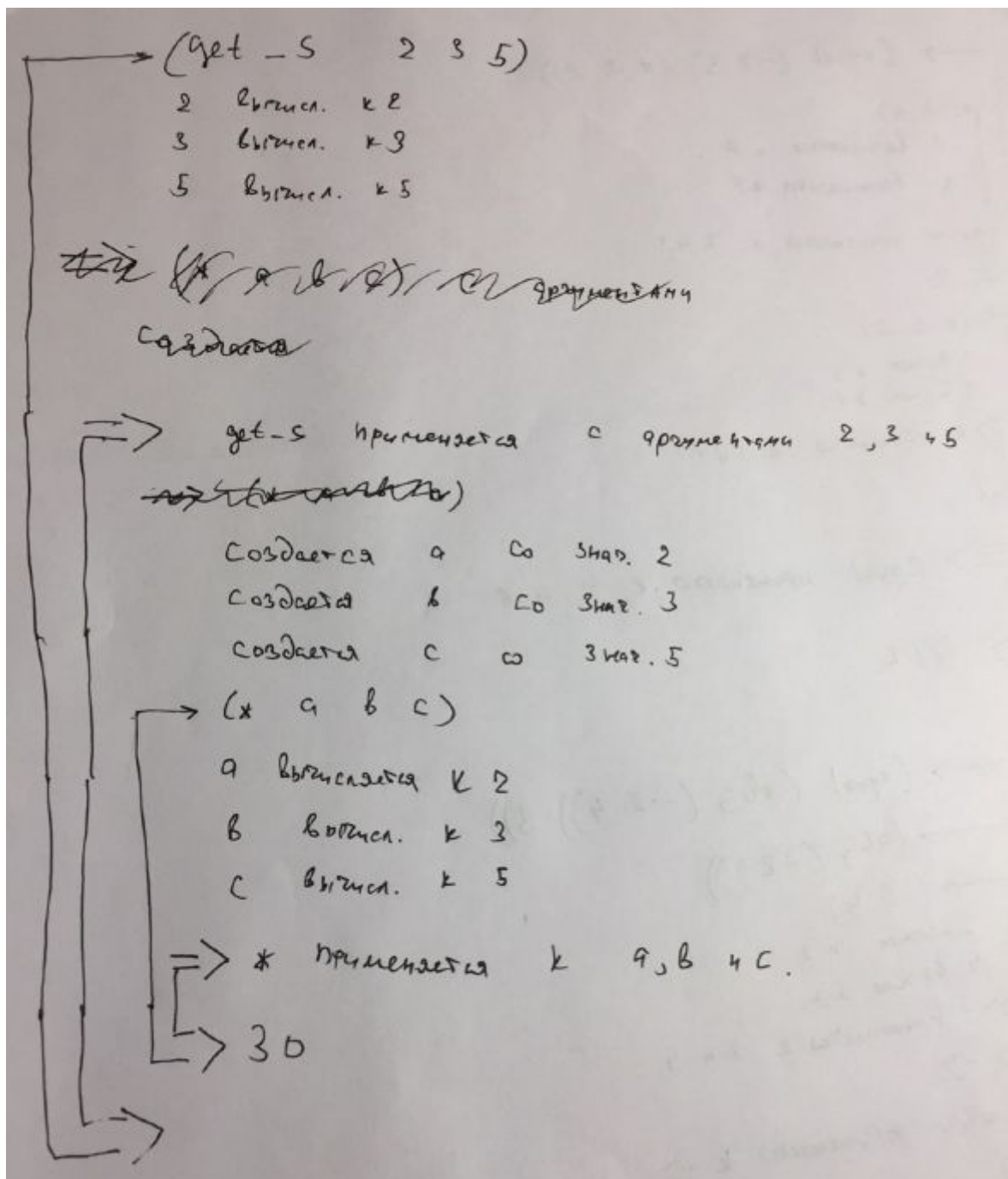
```
(defun hypotenuz(x y)
  (sqrt (+ (* x x) (* y y)))
)
```



### Задача 3.

Написать функция, вычисляющую объем параллелепипеда по трем его сторонам

```
(defun get_s(a b c)
  (* a b c)
)
```



#### Задача 4.

выражение	результат
(list 'a 'b c)	error: The variable c is unbound
(cons 'a (b c))	error: The variable c is unbound
(cons 'a '(b c))	(a b c)
(caddr (1 2 3 4 5))	error: illegal function call
(cons 'a 'b 'c)	error: invalid number of arguments: 3
(list 'a (b c))	error: The variable c is unbound
(list a '(b c))	error: The variable a is unbound
(list (+ 1 '(length '(1 2 3))))	error: The value (length '(1 2 3)) is not of type NUMBER

#### Задача 5.

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

```
(defun longer_then(listOne listTwo)
  (> (length listOne) (length listTwo))
)
```



**Задача 6.**

выражение	результат
(cons 3 (list 5 6))	(3 5 6)
(cons 3 '(list 5 6))	(3 LIST 5 6)
(list 3 'from 9 'gives (- 9 3))	(3 from 9 gives 6)
(+ (length '(1 foo 2 too))(car '(21 22 23)))	25
(cdr '(cons is short for and))	(is short for and)
(car (list one two))	error: The variable one in unbound
(car (list 'one 'two))	one

**Задача 6 (часть 2).**

Дана функция:

```
(defun mystery (x) (list (second x) (first x)))
```

выражение	результат
(mystery '(one two))	(two one)
(mystery 'free)	error: the value free is not of type LIST
(mystery (last 'one 'two))	error: the variable one is unbound

(mystery 'one 'two)	error: invalid number of arguments: 2
---------------------	---------------------------------------

## Контрольные вопросы.

### 1. Классификация функций в LISP

- по аргументам и поведению
  - Чистые функции - фиксированное количество аргументов, для определенного набора аргументов есть фиксированный результат;
  - Функции формы (специальные функции) - функции, которые принимают произвольное количество аргументов или по разному обрабатывают результат ;
  - Функции высшего порядка (Функционалы) - принимают или возвращают в качестве результата функцию.
  - псевдофункции - создают эффект на экране;
- по именованию:
  - именованные - можно определить через defun;
  - неименованные - определяются через lambda.

### 2. Базис языка LISP

**Базис** - минимальный набор возможностей, (средств) с помощью которых можно решить какую-то задачу.

**Базис Lisp** образуют атомы, структуры, базовые функции и функционалы.

### 3. Список: представление и интерпретация

Список - структура данных, состоящая из элементов, которыми могут быть атомы или другие списки. Список также может не содержать элементов вовсе, такой список называется пустым и обозначается как **Nil** или **()**.

Список является фундаментом языка LISP и может представлять как данные, так и код.

4. Как выполняются *car* и *cdr*? Какие результаты? Примеры.

Функции *car* и *cdr* служат для выделения головы и хвоста списка соответственно. Функции *car* и *cdr* можно применять только к списку и точечной паре. Попытка применить *car* и *cdr* к атому приведет к ошибке *ERRSTATE*.

Примеры:

(car '(1 2 3))	1
(cdr '(1 2 3))	2 3
(cdr (cdr '(1 2 3)))	3
(cdr (cdr (cdr '(1 2 3))))	Nil
(car '(a . b))	a
(cdr '(a . b))	b
(car 2)	ERRSTATE
(cdr 3)	ERRSTATE

5. Отличия в выполнении *list* и *cons*.

*list* можно применить для произвольного количества аргументов, *cons* работает только для двух. *cons* объединяет значение двух своих аргументов в точечную пару, *list* формирует список.