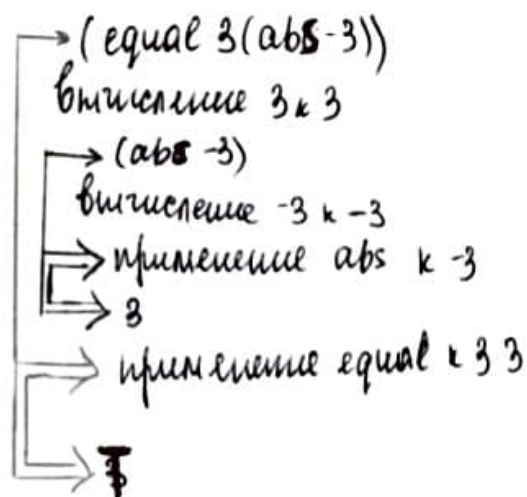


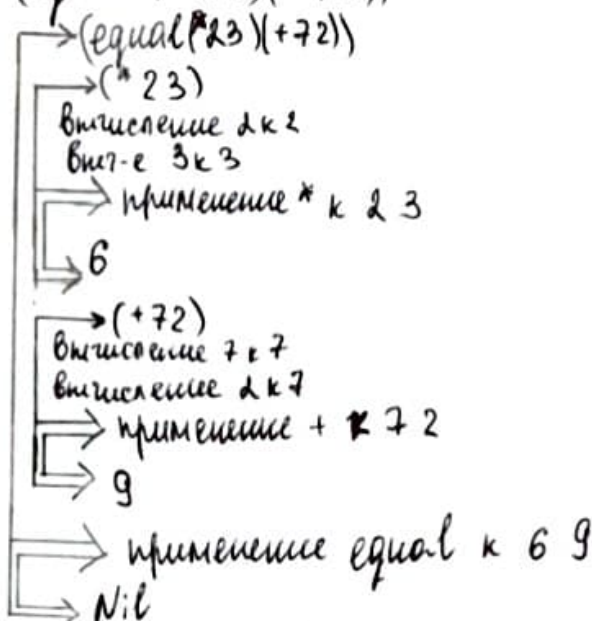
Цель работы: приобрести навыки работы в системе Common Lisp
 Задачи работы: изучить работу интерпретатора Lisp, алгоритм работы ф-ции eval, структуру и порядок обработки программы в Lisp.

Задание 1.

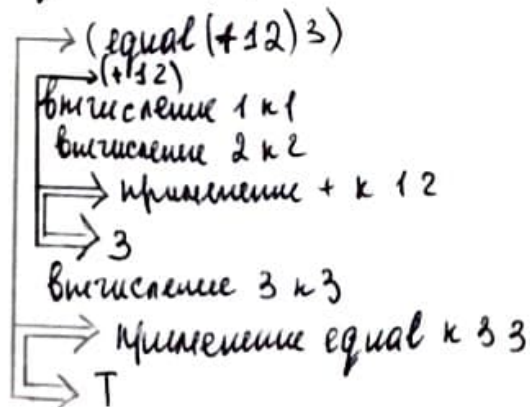
Составить диаграмму вычисления следующего выражения:
 (equal 3 (abs -3))



(equal (* 2 3) (+ 7 2))



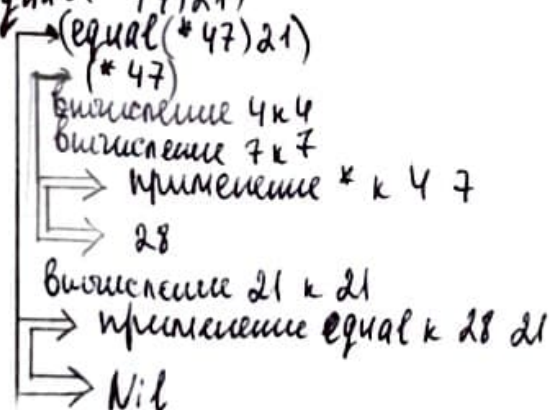
(equal (+ 1 2) 3)



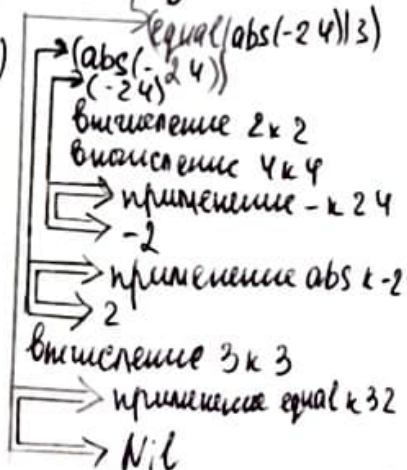
(equal (- 7 3) (* 3 2))



(equal (* 4 7) 21)



(equal (abs (- 2 4)) 3)



Задание 2

Написать ф-ю, вычисляющую гипотенузу прямоугольного треугольника по заданным катетам и составить диаграмму ее выполнения.

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

+

→ (hyp 3 4)

вычисление 3 к 3

вычисление 4 к 4

→ (sqrt(+(* x x)(* y y))) с аргументами 3, 4

создается x со значением 3

создается y со значением 4

→ (+(* x x)(* y y))

→ (* x x)

* вычисл к 3

* вычисл к 3

→ применение * к 3 3

→ 9

→ (* y y)

* вычисление y к 4

* вычисление y к 4

→ применение * к 4 4

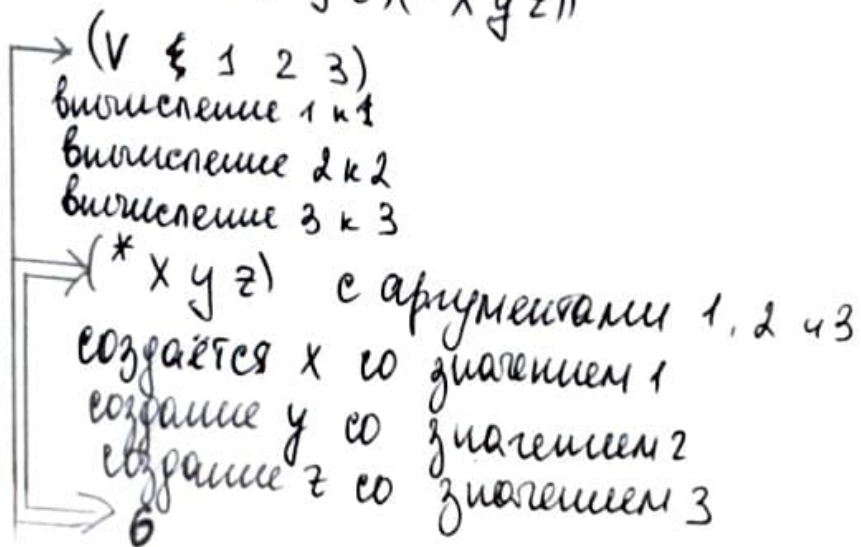
→ 16

→ применение + к 9 и 16

→ 25

→ 5.0

Написать ф-ю, вычисляющую объем параллелепипеда по 3-м его сторонам и составить диаграмму ее вычисления
(defun v(x y z)(* x y z))



Задача 4

Каков результат вычисления след-х выр-й?
 (list 'a 'b c)

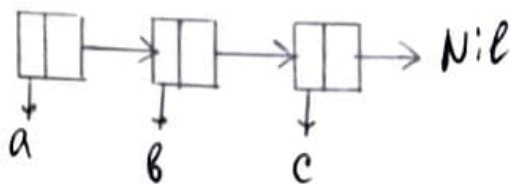
Результат: Error: The variable c is unbound.

Ошибка, т.к. при c нет апострофа, а значение не определено.
 (cons 'a (bc))

Результат: Undefined function: B. Undefined variable: c.

Вторым арг-м в cons передаём список (bc). Так как b стоит на первом месте и при нём нет апострофа, то lisp ~~не~~ попытается вычислить ф-ю с именем b. Но такой ф-ции нет. Т.е. аналогично, к c ~~не~~ попытается обратиться по значению, которое не определено. Если бы перед списком (bc) был апостроф, то lisp не пытался бы вычислить его.

(cons 'a '(b c))



Результат: (a b c)

Создана новая пара в одной ячейке которой a, а вторая ячейка указывает на двухэлементный список (b c). В памяти это представляется аналогично трех-элемент-ному списку.

(caddr(1 2 3 4 5))

Результат: Error: illegal function error.

(cons 'a 'b 'c)

Результат: Error: invalid number of arguments.

(list 'a (b c))

Рез-т: Error: Undefined function: B. Undefined ^{variable:} function c

(list a ~~(b c)~~) (b c)

Результат: Error: The variable A is unbound.

(list (+ 1 (length '(1 2 3))))

Результат: Error: The value (length '(1 2 3)) is not type of number. ^{т.к. / 750 список невозможно число}

Задача 5.

Написать ф-ю longer-than с двух списков-аргументов, которая возвращает T, если первый аргумент имеет большую длину.

(defun longer-than (l1 l2) (> (length l1) (length l2)))

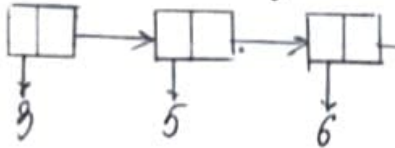
(longer-than '(a b c d) (a b)) → T

(longer-than '(a b) (a b c d)) → Nil

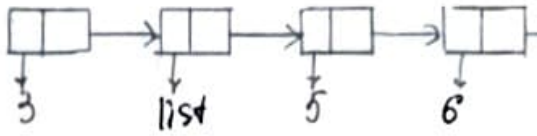
задание 8.

Каковы результаты вычисления следующих выражений?

$(\text{cons } 3 (\text{list } 5 \ 6)) \rightarrow (3 \ 5 \ 6)$



$(\text{cons } 3 '(\text{list } 5 \ 6)) \rightarrow (3 \ \text{list } 5 \ 6)$



$(\text{list } 3 \ \text{'from } 9 \ \text{'gives } (- \ 9 \ 3)) \rightarrow (3 \ \text{from } 9 \ \text{gives } 6)$

$(+ (\text{length } '(s \ \text{foo } 2 \ \text{too})) (\text{car } '(21 \ 22 \ 23))) \rightarrow 25$

$(\text{cdr } '(\text{cons is short for ans})) \rightarrow (\text{is short for ans})$

$(\text{car } (\text{list one two})) \rightarrow \text{The variable ONE is unbound.}$

$(\text{car } (\text{list 'one 'two})) \rightarrow \text{one}$

Теоретические вопросы

1) **Базис Lisp.**

Базис - минимальный набор средств языка, необходимый для решения любой задачи.

Базис Lisp:

- атомы
- бинарные узлы
- Ф-ции: atom, eq, cons, car, cdr
- Ф-ции: cond, quote, lambda, eval, label

2) **Классификация Ф-ций:**

с точки зрения действия:

- конструкторы
- селекторы
- преобразователи

С точки зрения организации:

- Численные математические
- Рекурсивные
- Сигнальные ф-ции (формы)
- Преобразующие
- Функции с вариантами значений из которых возвращается только одно значение
- Функции высших порядков (функционалы)

3) Список, представление и интерпретация списков.

Список - ^{если список не пустой, то у него есть голова и хвост.} выражение, которое может быть пустым и не пустым. Пустой список является атомом, то есть парой. Пустой список синтаксически обозначается либо Nil, либо (). Если список не пустой, то у него есть голова и хвост, которые в свою очередь тоже явл-ся списками и могут быть пустыми и не пустыми. Пустой список можно задать т/з точную пару (A.(B.(C.(D...))) или как послед-во атомов, раздел-х пробелами (A B C D). Любой список в памяти представляется ^{своей} списковой ячейкой, но если хвост не пустой, то он представ-ся другой списковой ячейкой и т.д.

4) Ф-ции car и cdr

car и cdr - базовые ф-ции доступа к данным. cdr принимает точную пару или пустой список и возвращает список, состоящий из всех эл-в, кроме первого. Если в списке меньше двух эл-в, то возвращ-т Nil. Car принимает точную пару или пустой список в качестве аргумента и возвращает первый эл-т или Nil, соответственно.

5) Назначение и отличие в работе const и list.

list и const исп-ся для создания списков.

const работает эффективно. Эта ф-я создает одну списковую ячейку и расставляет указатели на аргументы. Входит

list входит в ядро. Это - основные действия которые наиболее часто используются. Эта функция принимает переменное число аргументов и возвращает список, ^{входящий}

элементы которого - переработанные в ф-ю элементы.

Эта ф-я написана с использованием ф-ции const.