|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **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 № 3**

|  |  |
| --- | --- |
| **Тема** Списки в Lisp. Использование стандартных функций.  **Студент** Белоусова Ю.С.  **Группа** ИУ7-61Б  **Оценка (баллы) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Преподаватель** Толпинская Н.Б. |  |

Москва.

2020 г.

1. Составить диаграмму вычисления следующих выражений:

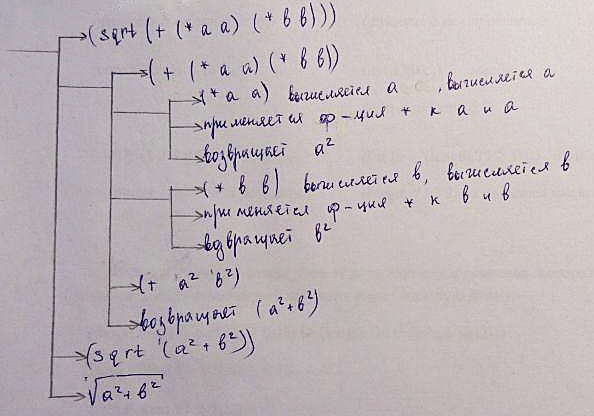
|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

2. Написать функцию, вычисляющую гипотенузу прямоугольного

треугольника по заданным катетам и составить диаграмму её вычисления.

(defun func (a b) (sqrt (+ (\* a a) (\* b b))))

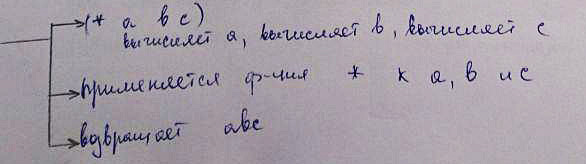
(func 12 9) - *15*

**

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

(defun func (a b c) (\* a b c))

(func 1 2 3) *- 6*



4. Каковы результаты вычисления следующих выражений?

|  |  |
| --- | --- |
| (list 'a c) *- ошибка*  *(нет значения с)* | (cons'a'b'c) *- ошибка*  *(слишком много аргументов)* |
| (cons'a (b c)) *- ошибка*  *(функция b не определена)* | (list 'a (b c)) *- ошибка*  *(функция b не определена)* |
| (cons'a '(b c)) - *(a b c)* | (list a '(b c)) *- ошибка*  *(нет значения a)* |
| (caddy (1 2 3 4 5)) *- ошибка*  *(функция caddy не определена)* | (list (+ 1 '(length '(1 2 3)))) *- ошибка*  *(length '(1 2 3) не является числом)* |

5. Написать функцию longer\_then от двух списков-аргументов, которая возвращает Т, если первый аргумент имеет большую длину.

(defun longer\_then (list1 list2) (> (length list1) (length list2)))

(longer\_then '(1 2 3) '(4 5 6 7)) *- Nil*

(longer\_then '(1 2 3) '(4 5 6)) *- Nil*

(longer\_then '(1 2 3) '(4 5)) *- T*

6. Каковы результаты вычисления следующих выражений?

(cons 3 (list 5 6))  *- (3 5 6)*

(list 3 'from 9 'lives (- 9 3)) *- (3 from 9 lives 6)*

(+ (length for 2 too)) (car '(21 22 23))) *- ошибка (нет значения for)*

(cdr ' (cons is short for ans)) *- (is short for ans)*

(car (list one two)) *- ошибка (нет значения one)*

(cons 3 '(list 5 6)) *- (3 list 5 6)*

(car (list 'one 'two)) *- one*

7. Дана функция (defun mystery (x) (list (second x) (first x))).

Какие результаты вычисления следующих выражений?

(mystery (one two)) *- ошибка (функция one не определена)*

(mystery (last one two)) *- ошибка (нет значения one)*

(mystery free) *- ошибка (нет значения free)*

(mystery one 'two)) *- ошибка (нет значения one)*

**Ответы на вопросы**

1.Базис языка Lisp

Атомы: -Символ

-Самоопределяемые атомы(числа: целые, действительные, рациональные; строки)

-Специальные символы(T, Nil)

Точечные пары:

-(<атом> . <атом>)

-(<точ. пара> . <атом>)

-(<атом> . <точ. пара>)

-(<точ. пара> . <точ. пара>)

S-выражения:

-<атом> | <точ.пара>

Списки:

-(<s-выражение> . <список>)

-(<пустой список>)|Nil

2.Классификация функций:

-Чистые(математические);

-Базисные;

-Формы;

-Функционалы;

Классификация базисных функций:

-Конструкторы(cons, list)

-Селекторы(car, cdr)

-Предикаты(позволяющие определить структуру элементов)

-Сравнение(позволяющие сравнивать элементы)

3.Список

Список - это динамическая структура данных, которая может быть пустой или не пустой, если не пустая состоит из 2х элементов: головы - может быть представлена любой структурой - и хвоста, являющегося списком.

4.Как выполняются CAR и CDR?

car - переходит по car указателю и возвращает голову

cdr - переходит по cdr указателю и возвращает хвост(остаток)

5.В чем отличие выполнения функций list и cons?

cons- является базисом языка, она на вход принимает ровно два аргумента и создает одну списковую ячейку(расставляет указатели)

cons- двухаргументная функция, создаёт одну списковую ячейку и расставляет указатели;

(cons 'a 'b) -> (a.b)

(cons 'a '(b)) -> (a b)

list- написана на базе функции cons, принимает любое количество аргументов и создает список

list- создаёт столько списковых ячеек, сколько аргументов

(list 'a 'b) -> (a b)

cons работает быстрее, чем list, но может быть организован не список