|  |  |
| --- | --- |
| 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:** Работа интерпретатора в lisp.  **Студент: Зейналов З. Г.**  **Группа: ИУ7-61Б**  **Оценка (баллы) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Преподаватель:** Толпинская Н. Б. |  |

Москва.

2020 г.

Цель: Приобрести навыки использования стандартных функций в lisp.

Описание ошибок, встречающихся в заданиях:

The Variable X is unbound – Возникает при попытке получить значение символа не связанного со значением.  
The function F is undefined – вызываемая функция не определена.  
Invalid numbers of arguments – В функцию было передано неверное количество аргументов.

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))

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

(defun f (a b)

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

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

(defun V (a b c)

(\* a b c)

1. Каковы результаты вычисления следующих выражений?

(list 'a c) -> Var C is unbound.

(cons'a'b'c) -> invalid argument count

(cons'a (b c)) -> Var C is unbound

(list 'a (b c)) -> Var C is unbound

(cons'a '(b c)) -> (A B C)

(list a '(b c)) -> Var a is unbound

(caddy (1 2 3 4 5)) -> caddy is not a function

(list (+ 1 '(length '(1 2 3)))) -> length ‘(1 2 3) is not a number type

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

(defun test (a b)

(> (length a)(length b)))

1. 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)) = Var For is unbound

(car '(21 22 23))) = 21

(cdr ' (cons is short for ans))

(car (list one two)) = Var one is unbound

(car (list 'one 'two)) = one

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

Какие результаты вычисления следующих выражений?

(mystery (one two)) -> The var two is unbound

(mystery one 'two)) -> The var one is unbound

(mystery (list one two)) -> The var one is unbound

(mystery free) -> The var free is unbound

1. Написать функцию, которая переводит температуру в системе Фаренгейта

температуру по Цельсию (defum f-to-c (temp)…).).

Формулы: c = 5/9\*(f-320); f= 9/5\*c+32.0.

Как бы назывался роман Р.Брэдбери "+451 по Фаренгейту" в системе по Цельсию? 2095/9 градусов по цельсию

(defun f-to-c (temp)

(/(\* 5 (- temp 32)) 9))

(defun c-to-f (temp)

(+(/(\* 9 temp) 5) 32))

1. 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 COMMON-LISP:T is undefined.

(apply #cons "(t NIL))

(eval NIL) => Nil

(list 'eval NIL) => (eval, Nil)

(eval (list 'eval NIL)) => Nil

Дополнительно:

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

(defun katet (gip katet)

(sqrt(-(\* gip gip)(\* katet katet))))

2. Написать функцию, вычисляющую площадь трапеции по ее основаниям и

высоте, и составить диаграмму ее вычисления.

(defun trapSquare (a b h)

(/(\*(+ a b) h) 2))

Ответы на вопросы:

1. Базис Lisp

Базис – это минимально необходимый набор конструкций, с помощью которого можно

реализовать задачу.

Базис Lisp образуют: атомы, структуры, базовые функции, базовые функционалы.

1. Классификация функций Lisp

• Базовые функции — принимают фиксированное количество аргументов

* Селекторы (car, cdr)
* Конструкторы (cons)
* Предикаты (atom, Null, lisp, ..)
* Функции сравнения (eq, eql, =, equal, equalp)
* Формы – принимают не фиксированное количество аргументов или обрабатывают аргументы по разному
* Функционалы – используют другие функции в качестве аргументов или вырабатывают в качестве результатов

1. Как представлены списки

Списки представлены с помощью списковых ячеек. Списочная ячейка состоит из двух частей, полей car и cdr. Каждое из полей содержит указатель. Указатель может ссылаться на другую списочную ячейку или на некоторый другой Lisp объект, как, например, атом.

Указатели между ячейками образуют цепочку, по которой можно из предыдущей ячейки попасть в следующую и так, наконец, до атомарных объектов. Каждый известный системе атом записан в определённом месте памяти лишь один раз.

1. Как работают CAR и CDR

CAR и CDR являются базовыми функциями доступа к данным. CAR принимает точечную пару или пустой список в качестве аргумента и возвращает первый элемент или nil, соответственно. CDR принимает точечную пару или пустой список и возвращает список состоящий из всех элементов, кроме первого. Если в списке меньше двух элементов, то

возвращается Nil.

1. Отличия list и сons

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

Вопросы к Лр 4.

1. Как синтаксически оформляется программа на Lisp.
2. Как трактуются элементы списка?

Списки представлены с помощью списковых ячеек. Списочная ячейка состоит из двух частей, полей car и cdr. Каждое из полей содержит указатель. Указатель может ссылаться на другую списочную ячейку или на некоторый другой Lisp объект, как, например, атом.

Указатели между ячейками образуют цепочку, по которой можно из предыдущей ячейки попасть в следующую и так, наконец, до атомарных объектов. Каждый известный системе атом записан в определённом месте памяти лишь один раз.

1. Порядок реализации