|  |  |
| --- | --- |
| 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 (ИУ7)**

**Отчет**

|  |  |
| --- | --- |
| **по лабораторной работе №** | 02 |

**Дисциплина: Функциональное и логическое программирование**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ7И-62Б |  |  | Во Д. Х. Ф |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | Толпинская Н. Б. |
|  |  |  |  | Строганов Ю. В. |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2022

**Теоретические вопросы**

1. **Базис языка Lisp**

Базис Lisp образуют: атомы, структуру из простейших бинарных узлов, базовые функции (atom, eq, cons, car, cdr) и базовые функционалы (quote, lambda, eval).

1. **Классификация функций**

Классификация функций:

1. чистые математические функции (имеют фиксированное количество аргументов и один результат);
2. формы (имеют произвольное количество аргументов или эти аргументы  
   обрабатываются не все одинаково);
3. функциональные (в качестве одного из аргументов принимают описание  
   функции).

Классификация базисных функций:

1. Селекторы (car, cdr)
2. Конструкторы (cons, list)
3. Предикаты (atom, listp, consp, numberp, symbolp, ...)
4. Сравнение (eq, eql, equal, equalp, =, ...)
5. **Способы создание функций**
6. Макроопределение defun - не базовое. В начале создания языка была  
   реализована идея, когда пользователь не создавал собственных функций, но строил конструкции из десятка базовых. defun - форма, особым образом обрабатывает некоторые из своих аргументов.

(defun имя (список аргументов) (тело)).

1. Базовый способ определения функции - lambda . Лисп базируется на  
   математике. Лямбда-выражение - способ записи выражения, когда вначале записывается формула, которую надо вычислить, а затем значения, над которыми её вычислить.  
   (lambda (аргументы) (тело))
2. **Функции Car и Cdr**

Функция *car* обеспечивает доступ к первому элементу списка – его <голове>, а  
функция *cdr* – к укороченному на одни элемент списку – его <хвосту>, т.е. к тому,  
что остается после удаления головы.

1. **Назначение и отличие в работе Cons и List**

cons - является базисом языка, она на вход принимает ровно два аргумента и  
создает одну списковую ячейку(расставляет указатели)  
list - написана на базе функции cons, принимает любое количество аргументов и  
создает список

**Практические задание**

**Задание 1**. Составить диаграмму вычисления следующих выражений:

→ (equal 3 (abs -3))

3 вычисляется к 3

→ (abs -3)

-3 вычисляется к -3

⇒ применить abs к -3

⇒ результат: 3

⇒ применить equal к (3 3)

⇒ результат: T

→ (equal (\* 4 7) 21)

→ (\* 4 7)

4 вычисляется к 4

7 вычисляется к 7

⇒ применить \* к (4 7)

⇒ результат: 28

21 вычисляется к 21

⇒ применить equal к (28 21)

⇒ результат: NIL

→ (equal (+ 1 2) 3)

→ (+ 1 2)

1 вычисляется к 1

2 вычисляется к 2

⇒ применить + к (1 2)

⇒ результат: 3

3 вычисляется к 3

⇒ применить equal к (3 3)

⇒ результат: T

→ (equal (\* 2 3) (+ 7 2))

→ (\* 2 3)

2 вычисляется к 2

3 вычисляется к 3

⇒ применить \* к (2 3)

⇒ результат: 6

→ (+ 7 2)

7 вычисляется к 7

2 вычисляется к 2

⇒ применить + к (7 2)

⇒ результат: 9

⇒ применить equal к (6 9)

⇒ результат: NIL

→ (equal (- 7 3) (\* 3 2))

→ (- 7 3)

7 вычисляется к 7

3 вычисляется к 3

⇒ применить - к (7 3)

⇒ результат: 4

→ (\* 3 2)

3 вычисляется к 3

2 вычисляется к 2

⇒ применить \* к (3 2)

⇒ результат: 6

⇒ применить equal к (4 6)

⇒ результат: NIL→ (equal (abs (- 2 4)) 3)

→ (abs (- 2 4))

→ (- 2 4)

2 вычисляется к 2

4 вычисляется к 4

⇒ применить - к (2 4)

⇒ результат: -2

⇒ применить abs к -2

⇒ результат: 2

3 вычисляется к 3

⇒ применить equal к (2 3)

⇒ результат: NIL

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

(defun v (a b c)

(\* a b c))

→ (v 2 3 4)

2 вычисляется к 2

3 вычисляется к 3

4 вычисляется к 4

→ (\* a b c)

создаётся a со значением 2

создаётся b со значением 3

создаётся c со значением 4

a, b и с вычисляется как 2, 3 и 4 соответсвенно

⇒ применить \* к (2 3 4)

⇒ результат: 24

⇒ применить v к (2 3 4)

⇒ результат: 24

**Задание 2.** Написать функцию, вычисляющую гипотенузу прямоугольного треугольника по заданным катетам и составить диаграмму её вычисления.

(defun hypotenuse (a b)

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

→ (hypotenuse3 4)

3 вычисляется к 3

4 вычисляется к 4

→ (sqrt (+ (\* a a) (\* b b)))

создаётся a со значением 3

создаётся b со значением 4

→ (+ (\* a a) (\* b b))

→ (\* a a)

a вычисляется как 3

a вычисляется как 3

⇒ применить \* к (3 3)

⇒ результат: 9

→ (\* b b)

b вычисляется как 4

b вычисляется как 4

⇒ применить \* к (4 4)

⇒ результат: 16

⇒ применить + к (9 16)

⇒ результат: 25

⇒ применить sqrt к 25

⇒ результат: 5.0

⇒ применить hypotenuse к (3 4)

⇒ результат: 5.0

**Задание 4.** Каковы результаты вычисления следующих выражений? (объяснить возможную ошибку и варианты ее устранен6ия)

(list 'a c) => (list 'a 'c) = (A C)

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

(cons 'a '(b c)) = (A B C)

(caddy (1 2 3 4 5)) => (caddr '(1 2 3 4 5)) = 3

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

(list 'a (b c)) => (list 'a '(b c)) = (A (B C))

(list a '(b c)) => (list 'a '(b c)) = (A (B C))

(list (+ 1 '(length '(1 2 3)))) => (list (+ 1 (length '(1 2 3)))) = (4)

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

(defun longer-than (lst1 lst2)

(cond ((null (cdr lst1)) (not lst2))

(T (longer-than (cdr lst1) (cdr lst2))))

**Задание 6.** 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)) (car '(21 22 23))) => 24

(cdr '(cons is short for ans)) => (IS SHORT FOR ANS)

(car (list one two)) => ‘ошибка’

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

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

Какие результаты вычисления следующих выражений?

(mystery (one two)) => ошибка; (mystery '(one two)) => (TWO ONE)

(mystery one 'two)) => ошибка

(mystery (last one two)) => ошибка; (mystery (list 'one 'two)) => (TWO ONE)

(mystery free) => ошибка

**Задание 8.** Написать функцию, которая переводит температуру в системе Фаренгейта температуру по Цельсию (defun f-to-c (temp)…).

Формулы: c = 5/9\*(f-32); f= 9/5\*c+32.0.

Как бы назывался роман Р.Брэдбери "+451 по Фаренгейту" в системе по Цельсию?

(defun f-to-c (temp)

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

Ответ: 2095/9 = 232,(7)

**Задание 9.** Что получится при вычисления каждого из выражений?

(list 'cons t NIL) = (CONS T NIL)

(eval (list 'cons t NIL)) = (T.NILL) = (T)

(eval (eval (list 'cons t NIL))) = ошибка (eval (T))

**Дополнительно**

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

(defun f1 (c a)

(sqrt (- (\* c c) (\* a a))))

2. Написать функцию, вычисляющую площадь трапеции по ее основаниям и  
высоте, и составить диаграмму ее вычисления.

(defun f2 (a b h)

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