|  |  |
| --- | --- |
| 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**

**по курсу “Функциональное и логическое программирование”**

**по теме “Определение функций пользователя”**

|  |  |
| --- | --- |
| Студент: | Уласик Е.А. |
| Группа: | ИУ7-61 |
| Преподаватель: | Толпинская Н.Б. |

*2020 г.*

1. **Цель работы и задачи работы**

Цель работы: приобрести навыки создания и использования функций пользователя в Lisp.

Задачи работы: изучить способы создания и использования именованных и неименованных функций пользователя для обработки списков.

1. **Теоретическая часть**
   1. Базис языка Lisp

Базис – базовые структуры и атомы и базовые функции.

* 1. Классификация функций
* Чистые математические функции (фиксированное количество аргументов и один результат);
* Специальные формы (функции с произвольным числом аргументов или аргументы могут быть не вычислены);
* Функционалы;
  1. Представление списка в памяти

Списки в Lisp состоят из списочных ячеек. Списочная ячейка состоит из двух частей, полей. Каждое из полей содержит указатель. Указатель может ссылаться на другую списочную ячейку или некоторый другой Lisp объект.

* 1. Функции CAR и CDR

Базовыми функциями доступа к данным в Lisp являются функции CAR и CDR.

* CAR – в качестве аргумента принимает список (1 аргумент) и возвращает первый элемент списка, если список не пустой и Nil – иначе;
* CDR – в качестве аргумента принимает список (1 аргумент) и возвращает список, состоящий из всех элементов, кроме первого, если список не пустой и Nil – иначе;
  1. Отличия работы функций list и cons
* CONS – принимает два аргумента, каждый из которых является S-выражением. Возвращает: 1) точечную пару, если второй аргумент не список; 2) Новый список, если второй аргумент список, а первый аргумент добавляет в начало.

(cons ‘A ‘B) = (A . B) (cons ‘A ‘(B)) = (A B)

* LIST – всегда создаёт список, не является чистой функцией, так как имеет произвольное количество аргументов.

(list ‘A ‘B ‘C ‘D) = (A B C D)

1. **Практическая часть**

Задание 1. Составить диаграмму вычисления следующих выражений:

* 1. (equal 3 (abs – 3))

обработка функции equal:

вычисление первого аргумента: 3;

вычисление второго аргумента:

обработка функции abs:

вычисление аргумента -3: -3;

применение abs к -3;

возврат 3;

применение equal к 3 и 3;

возврат T;

* 1. (equal (+ 1 2) 3)

обработка функции equal:

вычисление первого аргумента:

обработка функции +:

вычисление первого аргумента 1: 1;

вычисление второго аргумента 2: 2;

применение + к 1 и 2;

возврат 3;

вычисление второго аргумента 3: 3;

применение equal к 3 и 3;

возврат T;

* 1. (equal (\* 4 7) 21)

обработка функции equal

вычисление первого аргумента:

обработка функции \*:

вычисление первого аргумента 4: 4;

вычисление второго аргумента 7: 7;

применение \* к 4 и 7;

возврат 28;

вычисление второго аргумента 21: 21;

применение equal к 28 и 21;

возврат Nil;

* 1. (equal (\* 2 3) (+ 7 2))

обработка функции equal:

вычисление первого аргумента:

обработка функции \*:

вычисление первого аргумента 2: 2;

вычисление второго аргумента 3: 3;

применение \* к 2 и 3;

возврат 6;

вычисление второго аргумента:

обработка функции +:

вычисление первого аргумента 7: 7;

вычисление второго аргумента 2: 2;

применение + к 7 и 2;

возврат 9;

применение equal к 6 и 9;

возврат Nil;

* 1. (equal (- 7 3) (\* 3 2))

обработка функции equal:

вычисление первого аргумента:

обработка функции -:

вычисление первого аргумента 7: 7;

вычисление второго аргумента 3: 3;

применение – к 7 и 3;

возврат 4;

вычисление второго аргумента:

обработка функции \*:

вычисление первого аргумента 3: 3;

вычисление второго аргумента 2: 2;

применение \* к 3 и 2;

возврат 6;

применение equal к 4 и 6;

возврат Nil;

* 1. (equal (abs (- 2 4) 3))

обработка функции equal:

вычисление первого аргумента:

обработка функции abs:

вычисление аргумента:

обработка функции -:

вычисление первого аргумента 2: 2;

вычисление второго аргумента 4: 4;

применение – к 2 и 4;

возврат -2;

применение abs к -2;

возврат 2;

вычисление второго аргумента 3: 3;

применение equal к 2 и 3;

возврат Nil;

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

(defun f(k1 k2)(sqrt (+ (\* k1 k1) (\* k2 k2))))

> (f 5 12)

обработка функции f:

вычисление первого аргумента 5: 5;

вычисление второго аргумента 12: 12;

применение f к 5 и 12:

создание переменной k1 со значением 5;

создание переменной k2 со значением 12;

обработка функции sqrt:

вычисление аргумента:

обработка функции +:

вычисление первого аргумента:

обработка функции \*:

вычисление п. а. k1: 5;

вычисление в. а. k1: 5;

применение \* к 5 и 5;

возврат 25;

вычисление второго аргумента:

обработка функции \*:

вычисление п. а k2: 12;

вычисление в. а. k2: 12;

применение \* к 12 и 12;

возврат 144;

применение + к 25 и 144;

возврат 169;

применение sqrt к 169;

возврат 13;

возврат 13;

((lambda (k1 k2)(sqrt (+ (\* k1 k1) (\* k2 k2)))) 5 12)

Задание 3. Написать функцию, вычисляющую объём параллелепипеда по 3-м сторонам, и составить диаграмму её вычисления.

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

> (f 3 4 5)

обработка функции f:

вычисление первого аргумента 3: 3;

вычисление второго аргумента 4: 4;

вычисление третьего аргумента 5: 5;

применение f к 3, 4 и 5:

создание переменной a со значением 3;

создание переменной b со значением 4;

создание переменной c со значением 5;

обработка функции \*:

вычисление первого аргумента a: 3;

вычисление второго аргумента b: 4;

вычисление третьего аргумента c: 5;

применение \* к 3, 4 и 5;

возврат 60;

возврат 60;

Задание 4. Каковы результаты вычисления следующих выражений?

4.1 (list ‘a ‘b c)

Результат: The variable C is unbound.

4.2 (cons ‘a (b c))

Результат: The variable C is unbound.

4.3 (cons ‘a ‘(b c))

Результат: (a b c)

4.4 (caddr (1 2 3 4 5))

Результат: illegal function call

4.5 (cons ‘a ‘b ‘c)

Результат: invalid number of arguments: 3

4.6 (list ‘a (b c))

Результат: The variable C is unbound.

4.7 (list a ‘(b c))

Результат: The variable A is unbound.

4.8 (list (+ 1 ‘(length ‘(1 2 3))))

Результат: The value (LENGTH '(1 2 3)) is not of type NUMBER

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

(defun f(ls1 ls2)(> (length ls1) (length ls2)))

Задание 6. Каковы результаты вычисления следующих выражений?

6.1 (cons 3 (list 5 6))

Результат: (3 5 6)

6.2 (list 3 ‘from 9 ‘gives (- 9 3))

Результат: (3 from 9 gives 6)

6.3 (+ (length ‘(1 foo 2 too))(car ‘(21 22 23)))

Результат: 25

6.4 (cdr ‘(cons is short for ans))

Результат: (is short for ans)

6.5 (car (list one two))

Результат: The variable ONE is unbound.

6.6 (cons 3 ‘(list 5 6))

Результат: (3 list 5 6)

6.7 (car (list ‘one ‘two))

Результат: one

Задание 6. Дана функция (defun mystery(x) (list (second x) (first x))) Какие результаты вычисления следующих выражений?

6.1 (mystery ‘(one two))

Результат: (two one)

6.2 (mystery (last ‘one ‘two))

Результат: The value ONE is not of type LIST

6.3 (mystery ‘free)

Результат: The value FREE is not of type LIST

6.4 (mystery ‘one ‘two)

Результат: invalid number of arguments: 2