Федеральное государственное бюджетное образовательное учреждение высшего

образования

«Чувашский государственный университет им. И.Н. Ульянова»

Кафедра вычислительной техники

Лабораторная работа №8

**Функции в языке Лисп. Вычисления в Лиспе**

Вариант 9

Выполнил: Иванов В.С.

студент группы ИВТ-41-22

Проверил: кандидат технических наук

Обломов Игорь Александрович

Чебоксары, 2024

Цель работы: изучение лямбда-функций, пользовательских функций, использование их на практике.

***Основные теоретические сведения***

Ключевое понятие, базис функционального программирования *лямбда-функция*. Понятие лямбда-функции впервые было введено Алонсом Черчем в начале 20 столетия. Это понятие в дальнейшем послужило основой для создания теории лямбда-исчисления.

В Лиспе лямбда-выражение имеет вид

**(lambda(x1,x2,x3,…xn) fn.).**

Символ lambda означает, что речь идет об определении функции. Символы xi  – *формальные параметры* определения, a fn – *тело функции*. Телом функции является произвольная *форма.*

Лямбда-выражение – это определение вычислений и параметров функции в чистом виде без *фактических параметров*.

Лямбда-выражение является абстрактным механизмом для определения и описания вычислений. Также лямбда-выражение иногда называют безымянной функцией которая пропадает тотчас после вычислений, ее нельзя вызвать по имени.

Дать имя и определить новую функцию можно с помощью специализированной функции defun (define function), которая имеет следующий формат:

**(defun name lambda-list body).**

Программа на языке Лисп состоит из последовательностей форм и функций. Под формой понимается любое символьное выражение, значение которого может быть вычислено интерпретатором. Лямбда-выражение без фактических параметров не является формой. Формы бывают трех видов:

1*. Самоопределенные* (self-evaluating) формы. Это лисповские объекты, представляющие лишь самих себя. К ним относят константы языка (T, NIL), числа.

2. Символы*,* используемые в качестве имен переменных.

3. Формы в виде списочной структуры, которыми являются:

3.1. Вызовы функций и лямбда-вызовы.

3.2. *Специальные формы* (special form), формы предназначенные для управления вычислительным процессом и контекстом.

3.3*. Макровызовы.*

Работа предложения происходит по следующему сценарию: последовательно вычисляются значения предикатов predi до тех пор, пока значение одного из них не будет NIL, после чего вычисляется значение связанной с предикатом формы, которое выдается в качестве результата всего предложения. Если истинного предиката не обнаруживается, то в качестве результата выдается NIL. Рекомендуется в качестве последнего предиката использовать символ Т (true), и соответствующее ему результирующее выражение будет вычисляться всегда, когда ни одно другое условие не выполняется.

Индивидуальное задание: определить лямбда-вызов для вычисления площади поверхности шара. Определить функцию пользователя для предыдущего задания. Пользуясь предложениями cond, if, when или unless, описать функцию, позволяющую определить, попадет ли точка с заданными координатами в область, ограниченную функцией sin и осью абсцисс.

Код программы

((lambda (r) (\* pi r r)) 5)

(defun plosh (r) ((lambda (r) (\* pi r r)) r))

(defun popadanie (x y)

    (cond ((and (>= y 0) (<= y (sin x))) t)

    (t nil))

)

(defun popadanie\_if (x y)

    (if (and (>= y 0) (<= y (sin x)))

    t

    nil)

)

(defun popadanie\_when (x y)

    (unless (or (< y 0) (> y (sin x)))

    t)

)

(defun popadanie\_unless (x y)

    (if (and (>= y 0) (<= y (sin x)))

    t

    nil)

)

(write (plosh 5))

(terpri)

(write (popadanie 1 0.5))

Вывод: изучил лямбда-функции, пользовательские функции, использовал их на практике.