|  |  |
| --- | --- |
| 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 №4**

**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. **Практическая часть**

Задание 7.Написать функцию, которая переводит температуру в системе Фаренгейта *в* температуру по Цельсию (defun f-to-c(temp)…) Формулы: ; Как бы назывался роман Р. Брэдбери “+451 по Фаренгейту” в системе по Цельсию?

(defun f-to-c(temp)(\* (/ 5.0 9.0) (- temp 32)))

Роман Р. Брэдбери “+451 по Фаренгейту” в системе по Цельсию назывался бы “+232.77779 по Цельсию”.

Задание 8. Что получится при вычислении каждого из выражений?

8.1 (list ‘cons T Nil)

Результат: (cons T Nil)

8.2 (eval (eval (list ‘cons T Nil)))

Результат: The function COMMON-LISP:T is undefined.

8.3 (apply #’cons ‘(T Nil))

Результат: (T)

8.4 (list ‘eval Nil)

Результат: (eval Nil)

8.5 (eval (list ‘cons T Nil))

Результат: (T)

8.6 (eval Nil)

Результат: Nil

8.7 (eval (list ‘eval Nil))

Результат: Nil

Дополнительные задания:

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

(defun f(gip k) (sqrt (- (\* gip gip) (\* k k))))

> (f 13 12)

обработка функции f:

вычисление первого аргумента 13: 13;

вычисление второго аргумента 12: 12;

применение f к 13 и 12:

создание переменной gip со значением 13;

создание переменной k со значением 12;

обработка функции sqrt:

вычисление аргумента:

обработка функции -:

вычисление первого аргумента:

обработка функции \*:

вычисление п. а. gip: 13;

вычисление в. а. gip: 13;

применение \* к 13 и 13;

возврат 169;

вычисление второго аргумента:

обработка функции \*:

вычисление п. а k: 12;

вычисление в. а. k: 12;

применение \* к 12 и 12;

возврат 144;

применение - к 169 и 144;

возврат 25;

применение sqrt к 25;

возврат 5;

возврат 5;

((lambda (gip k)(sqrt (- (\* gip gip) (\* k k)))) 13 12)

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

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

> (S 6 8 5)

Обработка функции S:

вычисление первого аргумента 6: 6;

вычисление второго аргумента 8: 8;

вычисление третьего аргумента 5: 5;

применение S к 6, 8 и 5:

создание переменной a со значением 6;

создание переменной b со значением 8;

создание переменной h со значением 5;

обработка функции \*:

вычисление первого аргумента h: 5;

вычисление второго аргумента:

обработка функции /:

вычисление первого аргумента:

обработка функции +:

вычисление п. а. a: 6;

вычисление в. а. b: 8;

применим + к 6 и 8;

возврат 14;

вычислим второй аргумент 2: 2;

применим / к 14 и 2;

возврат 7;

применим \* к 5 и 7;

возврат 35;

возврат 35;

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

1. Теоретическая часть

3.1 Синтаксическое представление программы на Lisp, хранение программы в памяти

Программы на Lisp представлены в виде S-выражения. Интерпретатор Lisp читает входящие команды, имеющие вид S-выражений, вычисляет значение каждого из введённых выражений, и возвращает результат. Lisp при работе программы может изменять программу за счет использования списков. Некоторые программы в Lisp строятся из рекурсивных функций над S-выражениями. Определения и вывозы этих функций так же имеют вид S-выражений, то есть формально они могут быть обработаны как обычное данные, полученные в процессе вычислений, и преобразованы как значения. Переменные, константы, выражения ветвления, вызовы функций представляются в виде S-выражений.

Самая простая форма выражения – переменная. Она может быть представлена как атом. Lisp-интерпретатор в ходе своей работы поддерживает специальную таблицу символьных атомов (таблица символов) в которой хранится информация обо всех атомах, встретившихся в тексте интерпретируемой программы или в качестве обрабатываемых данных. При обработке очередного символьного атома интерпретатор проверяет, занесен ли он в таблицу. Атом может иметь несколько независимых друг от друга значений. Символьный атом может использоваться как имя функции или как имя функционального параметра. Нужная интерпретация определяется из контекста его применения.

Символьный атом представляется в памяти 5 указателями.

|  |  |
| --- | --- |
| Name | Имя атома |
| Value | Значение атома если он используется как переменная |
| Function | Связянная функция |
| Properties | Список свойство (имя свойства, значение) |
| package | Имя пакета содержащего этот атом |

3.2 Как трактуются элементы списка

Список — это динамическая структура, которая может быть пустой и не пустой. Если не пустой, то состоит из двух элементов: первый (голова) – любая структура, второй (хвост) – список. Список состоит из бинарных узлов. Каждый бинарный узел имеет небольшой объем, достаточный для хранения двух типизированных указателей (CAR и CDR, голова и хвост).

3.3 Порядок реализации программы

Реализация программы определяется базовой функцией eval, которая запускается автоматически

