

	<p>Министерство образования и науки 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)</p>
---	--

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Отчет по лабораторной работе № 4

По курсу: "Функциональное и логическое
программирование"

Студент:

Турсунов Жасурбек Рустамович

Группа: ИУ7-66Б

Преподаватели:

Толпинская Наталья Борисовна

Строганов Юрий Владимирович

Москва, 2021 г.

Содержание

Введение	2
Задание 7	3
Задание 8	3
Дополнительно 1	3
Дополнительно 2	4
Ответы на вопросы	5

Цель работы: приобрести навыки работы с управляющими структурами Lisp.

Задачи работы: изучить работу функций с произвольным количеством аргументов, функций разрушающих и неразрушающих структуру исходных аргументов.

Введение

Многие стандартные функции Lisp являются формами и реализуют особый способ работы со своими аргументами. К таким функциям относятся функции, позволяющие работать с произвольным количеством аргументов: and, or, append, или особым образом обрабатывающее свои аргументы: функции cond, if, append, remove, reverse, substitute.

Если на вход функции подается структура данных(списков), то возникает вопрос: сохранится ли возможность в дальнейшем работать с исходными структурами, или они изменятся в процессе реализации функции. В Lisp существуют функции, использующие списки в качестве аргументов и разрушающие структуру исходных аргументов при этом часть из них позволяет использовать произвольное количество аргументов, а часть нет.

Задание 7

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

```
1 (defun f-to-c (temp)
2   (* (- temp 32.0) 5/9)
3 )
```

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

```
1 >(f-to-c 451)
2 ;232.77779
```

Задание 8

Что получится при вычисления каждого из выражений?

Выражение	Результат
(list 'cons t Nil)	(CONS T NIL)
(eval(list 'cons t Nil))	T
(eval(eval(list 'cons t Nil)))	Undefined function T
(eval Nil)	NIL
(list 'eval Nil)	(EVAL NIL)
(eval(list 'eval Nil))	NIL
(apply #cons "(t nil))	bad syntax for complex number: #CONS

Дополнительно 1

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

```
1 (defun kateta(a b)
2   (sqrt (-(* a a) (* b b)))
3 )
4
```

```

5      >(kateta 5 4)
6      ;3

```

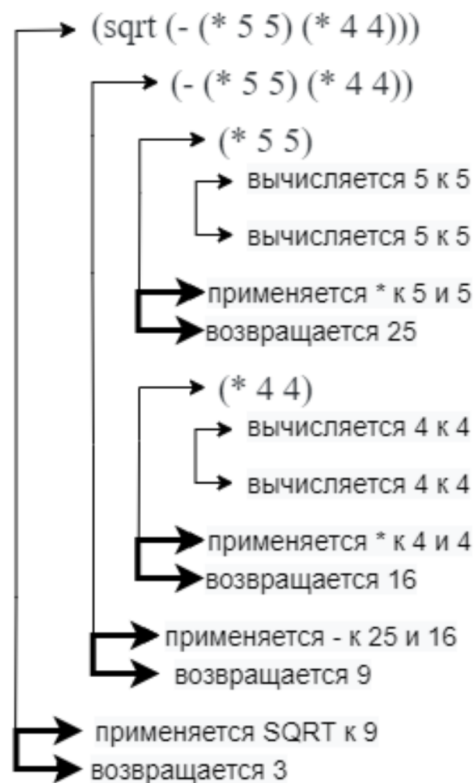


Рис. 1: диаграмма функции вычисляющую катет

Дополнительно 2

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

```

1      (defun area(a b h)
2          (* (+ a b) h 0.5)
3      )
4
5      >(area 3 4 5)
6      ;17.5

```



Рис. 2: диаграмма функции вычисляющую площадь трапеции

Ответы на вопросы:

1) Трактовка элементов списка:

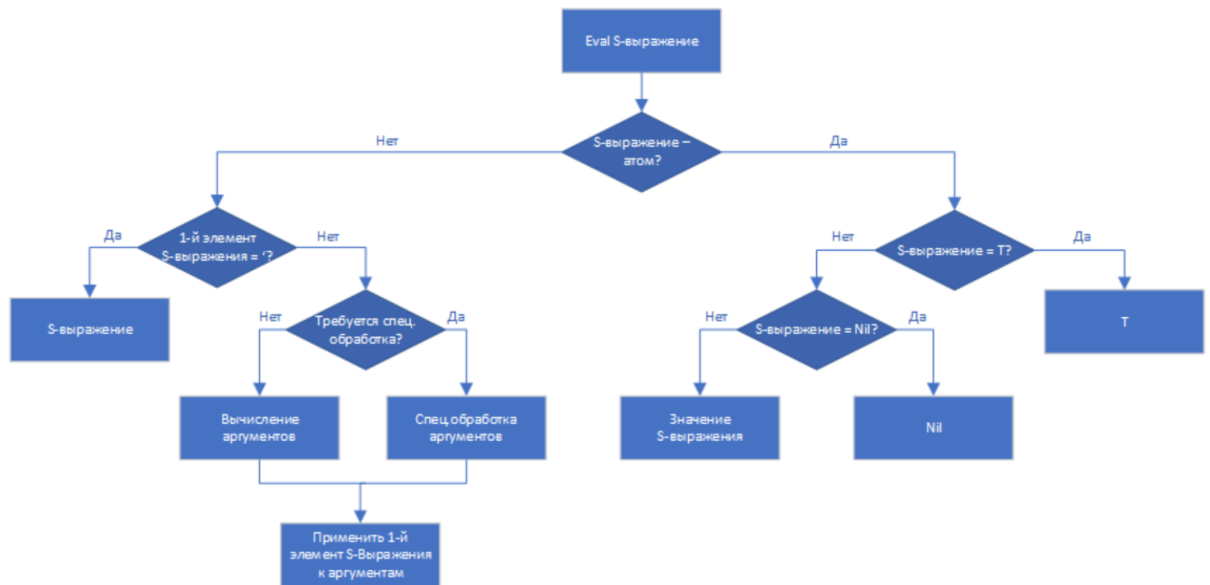
Списки представлены с помощью списковых ячеек. Списочная ячейка состоит из двух частей, полей `car` и `cdr`. Каждое из полей содержит указатель. Указатель может ссылаться на другую списочную ячейку или на некоторый другой Lisp объект, как, например, атом.

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.

Если это не стоит блокировка вычисления (`quote`, `'`), то первый элемент трактуется как имя функции, остальные — как аргументы.

2) Порядок реализации программы:

Реализации программы определяется базовой функцией `eval`:



3) Способы определения функций

Новые функции можно определить с помощью оператора `defun`. Он принимает три или более аргументов: имя, список параметров и ноль или более выражений, которые составляют тело функции.

```
(defun func_name (arg1 arg2 ... argN) func_body)
```

Но функция не обязательно должна иметь имя, для того, чтобы определить функцию, не имеющую имени, необходимо воспользоваться лямбда-выражением. Лямбда-выражение – это список, содержащий символ `lambda` и следующие за ним список аргументов и тело, состоящее из нуля или более выражений.

```
(lambda (arg1 arg2 .. argN) func_body)
```

4) Синтаксическая форма и хранение программы в памяти

Синтаксическая программа на LISP выполняется в виде S-выражения (обычно - списка). Программы в памяти хранятся точно так же, как и данные. Чтобы в LISP выдавать информацию то за данные, то за программу, надо чтобы все помещалось в кучу, при этом ведется работа с указателями.