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»

ОТЧЕТ

к лабораторной работе №4

По курсу: «Функциональное и логическое программирование»

Тема: «Использование управляющих структур, работа со списками».

Студент: Якуба Д.В.
Группа: ИУ7-63Б
Преподаватели: Толпинская Н. Б.,
Строганов Ю. В.

Москва, 2021 г.

Практическая часть

Задание 1. Написать функцию, которая переводит температуру в системе Фаренгейта в температуру по Цельсию (defun f-to-c (temp_ …).

Формулы:

$$c = \frac{5}{9} \cdot (f - 32.0); f = \frac{9}{5} \cdot c + 32.0.$$

Как бы назывался роман Р. Брэдбери «+451° по Фаренгейту» в системе по Цельсию?

Решение:

```
(defun f-to-c (temperature)
    (* (/ 5 9) (- temperature 32.0)))
```

«+233° по Цельсию».

Задание 2. Что получится при вычислении каждого из выражений?

(list `cons t NIL);

(eval (list `cons t NIL));

(eval (eval (list `cons t NIL)));

(apply #`cons `(t NIL));

(eval NIL);

(list `eval NIL);

(eval (list `eval NIL)).

Решение:

(list `cons t NIL) -> (CONS T NIL);

(eval (list `cons t NIL)) -> (T). Так как функция eval вычисляет переданное ей в качестве аргумента s-выражение, то вызов (eval `(cons t nil)) может быть аналогично записан как (cons t nil), что создаст список, состоящий из одного элемента;

(eval (eval (list `cons t NIL))) –> функция T не определена;

(apply #`cons `(t NIL)) -> (T).Функция apply применяет первый аргумент (как функцию) к элементам списка, составляющим второй аргумент;

(eval NIL) -> NIL;

(list `eval NIL) -> (EVAL NIL);

(eval (list `eval NIL)) -> NIL.

Задание 3. Написать функцию, вычисляющую катет по заданной гипотенузе и другому катету прямоугольного треугольника, и составить диаграмму её вычисления.

Решение:

```
(defun findCat (gip cat)
    (sqrt ( - (* gip gip) (* cat cat))))
```

```
(findCat 5 4)
    5 вычисляется к 5
    4 вычисляется к 4
    вызов findCat с аргументами 5 и 4
    (sqrt (- (* gip gip) (* cat cat))) с аргументами 5 и 4
        создаётся gip со значением 5
        создаётся cat со значением 4
        (- (* gip gip) (* cat cat))
            (* gip gip)
                gip вычисляется к 5
                * применяется к 5 и 5
                25
            (* cat cat)
                cat вычисляется к 4
                * применяется к 4 и 4
                16
            - применяется к 25 и 16
            9
        sqrt применяется к 9
        3
    3
```

Задание 4 см. на следующей странице.

Задание 4. Написать функцию, вычисляющую площадь трапеции по её основаниям и высоте, и составить диаграмму её вычисления.

Решение:

```
(defun trapezoid (baseTop baseBot height)
    (* height (/ (+ baseTop baseBot) 2.0)))
```

```
(trapezoid 2 3 4)
    2 вычисляется к 2
    3 вычисляется к 3
    4 вычисляется к 4
    вызов trapezoid с аргументами 2, 3, 4
    (* height (/ (+ baseTop baseBot) 2.0)) с аргументами 2, 3, 4
        создаётся baseTop со значением 2
        создаётся baseBot со значением 3
        создаётся height со значением 4
        height вычисляется к 4
        (/ (+ baseTop baseBot) 2)
            (+ baseTop baseBot)
                baseTop вычисляется к 2
                baseBot вычисляется к 3
                + применяется к 2 и 3
                5
            2 вычисляется к 2
            / применяется к 5 и 2
            2.5
        применяется * к 4 и 2.5
        10
    10
```

Теоретическая часть

1. Синтаксическая форма и хранение программы в памяти.

Формы представления данных и программы – s-выражения. Поэтому программы во время выполнения могут обрабатывать и преобразовывать другие программы и самих себя.

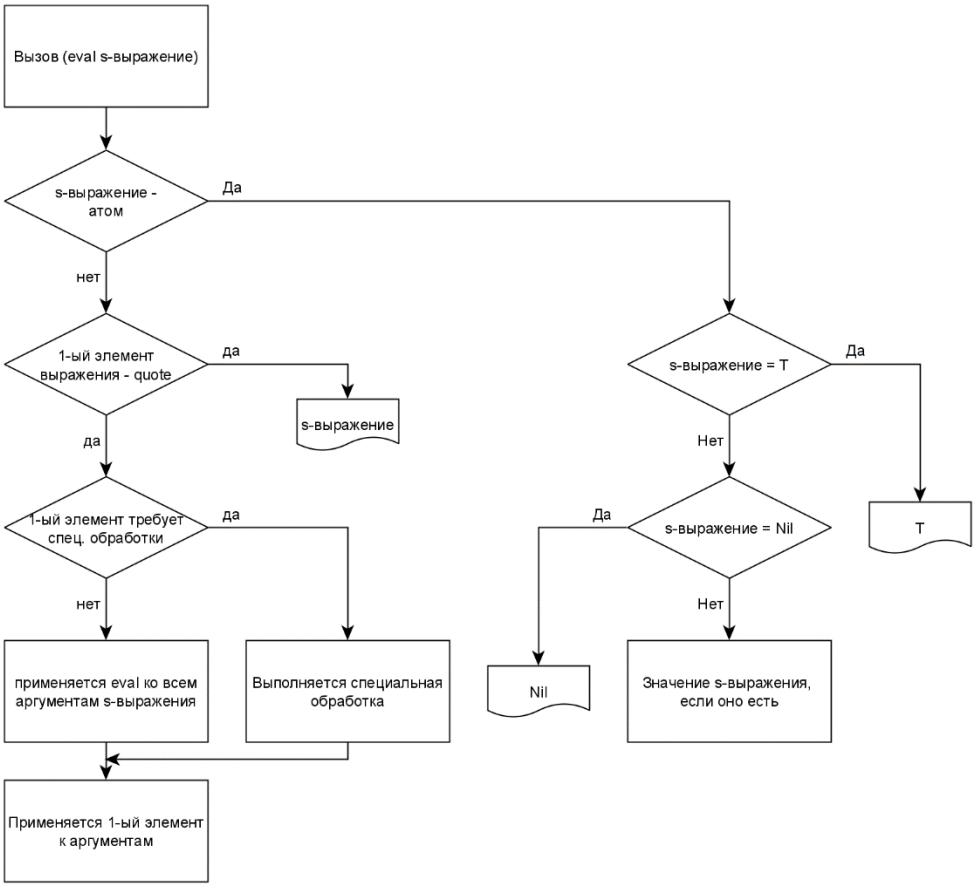
2. Трактовка элементов списка.

При отсутствии блокировки на вычисление (quote или `) первый элемент списка – имя функции, а последующие – аргументы, с которыми она будет вызвана.

3. Порядок реализации программы.

Программа работает в цикле: ожидает ввода s-выражения, передаёт введённое выражение функции eval, выводит последний полученный результат и вновь начинает работу с первого указанного пункта.

Работа функции eval:



4. Способы определения функции.

Первый способ:

```
(defun *имя функции* (*список параметров*) (
      *тело функции*
      )
)
```

Пример:

```
(defun findCat (gip cat)
    (sqrt ( - (* gip gip) (* cat cat))))

(findCat 5 4) -> 3.0;
```

Второй способ:

```
(lambda (*список аргументов*)) (*тело функции*))
```

Пример:

```
((lambda (a) (* a 3)) 4) -> 12
```

lamda-функции называются «безымянными». Суть такой функции состоит в том, что задается алгоритм вычисления, но не задается имени функции. Подобную функцию можно применить к списку аргументов и сразу получить результат.