



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»

**Отчёт по лабораторной работе №3
по курсу «Функциональное и логическое
программирование»**

**Тема: Определение функций
пользователя**

Студент: Сорокин А. П. ИУ7-66Б

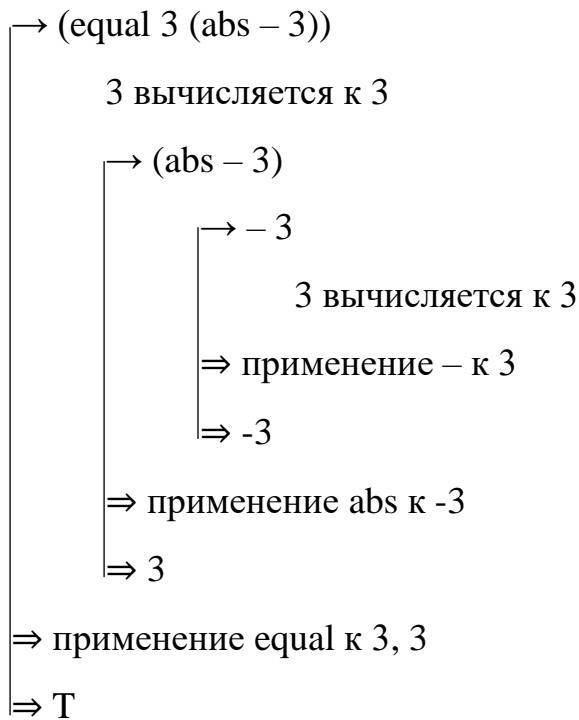
Преподаватели: Толпинская Н. Б.
Строганов Ю. В.

2021 г.

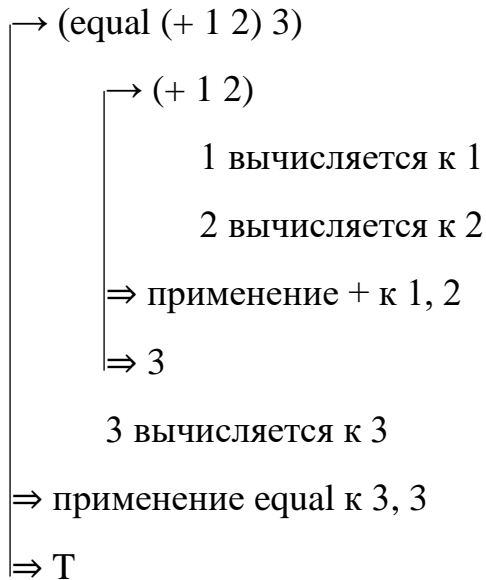
Задание 1

Составить диаграмму вычисления для указанных выражений.

1.1 (equal 3 (abs - 3))



1.2 (equal (+ 1 2) 3)



1.3 (equal (* 4 7) 21)

→ (equal (* 4 7) 21)
 → (* 4 7)
 4 вычисляется к 4
 7 вычисляется к 7
 ⇒ применение * к 4, 7
 ⇒ 28
 21 вычисляется к 21
⇒ применение equal к 28, 21
⇒ NIL

1.4 (equal (* 2 3) (+ 7 2))

→ (equal (* 2 3) (+ 7 2))
 → (* 2 3)
 2 вычисляется к 2
 3 вычисляется к 3
 ⇒ применение * к 2, 3
 ⇒ 6
 → (+ 7 2)
 7 вычисляется к 7
 2 вычисляется к 2
 ⇒ применение + к 7, 2
 ⇒ 9
⇒ применение equal к 6, 9
⇒ NIL

1.5 (equal (- 7 3) (* 3 2))

→ (equal (- 7 3) (* 3 2))
→ (- 7 3)
7 вычисляется к 7
3 вычисляется к 3
⇒ применение - к 7, 3
⇒ 4
→ (* 3 2)
3 вычисляется к 3
2 вычисляется к 2
⇒ применение * к 3, 2
⇒ 6
⇒ применение equal к 4, 6
⇒ NIL

1.6 (equal (abs (- 2 4)) 3)

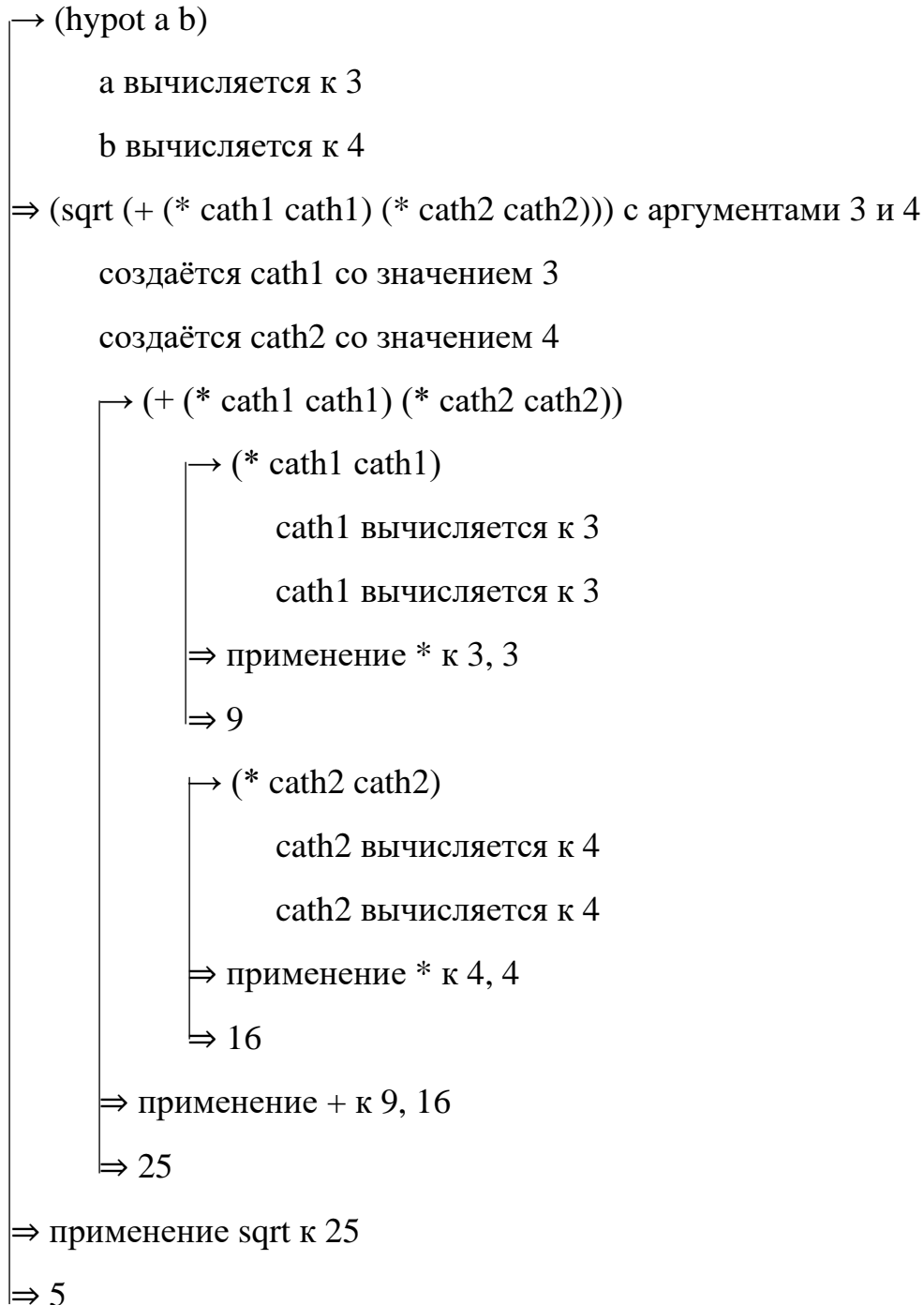
→ (equal (abs (- 2 4)) 3)
→ (abs (- 2 4))
→ (- 2 4)
2 вычисляется к 2
4 вычисляется к 4
⇒ применение - к 2, 4
⇒ -2
⇒ применение abs к -2
⇒ 2
3 вычисляется к 3
⇒ применение equal к 2, 3
⇒ NIL

Задание 2

Функция, вычисляющая гипотенузу прямоугольного треугольника по заданным катетам:

```
(defun hypot (cath1 cath2) (sqrt (+ (* cath1 cath1) (* cath2 cath2))))
```

Диаграмма вычисления функции:



Задание 3

Функция, вычисляющая объём параллелепипеда по трём его сторонам:

```
(defun v (a1 a2 a3) (* a1 a2 a3))
```

Диаграмма вычисления функции:

→ (v a b c)

а вычисляется к 2

b вычисляется к 3

с вычисляется к 4

⇒ (* a1 a2 a3) с аргументами 2, 3, 4

создаётся a1 со значением 2

создаётся a2 со значением 3

создаётся a3 со значением 4

a1 вычисляется к 2

a2 вычисляется к 3

a3 вычисляется к 4

⇒ применение * к 2, 3, 4

⇒ 24

Задание 4

Результаты выражений

Выражение	Результат
(list 'a 'c)	Ошибка: UNBOUND-VARIABLE c
(cons 'a (b c))	Ошибка: UNBOUND-VARIABLE c
(cons 'a '(b c))	(a b c)
(caddr (1 2 3 4 5))	Ошибка: Illegal function call
(cons 'a 'b 'c)	Ошибка: Invalid number of arguments
(list 'a (b c))	Ошибка: UNBOUND-VARIABLE c
(list a '(b c))	Ошибка: UNBOUND-VARIABLE a
(list (+ 1 '(length '(1 2 3))))	Ошибка: TYPE-ERROR

Задание 5

Функция `longer_then` от двух списков-аргументов, которая возвращает Т, если первый аргумент имеет большую длину:

1) с использованием *length*

```
(defun longer_then (list1 list2) (> (length list1) (length list2)))
```

2) с использования базисных функций

```
(defun longer_then (list1 list2)
  (cond ((null list1) nil)
        ((null list2) T)
        (T (longer_then (cdr list1) (cdr list2)))))
```

Задание 6

Результаты выражений

Выражение	Результат
<code>(cons 3 (list 5 6))</code>	<code>(3 5 6)</code>
<code>(cons 3 '(list 5 6))</code>	<code>(3 list 5 6)</code>
<code>(list 3 'from 9 'gives (- 9 3))</code>	<code>(3 from 9 gives 6)</code>
<code>(+ (length '(1 foo 2 too)) (car '(21 22 23)))</code>	<code>25</code>
<code>(cdr '(cons is short for ans))</code>	<code>(is short for ans)</code>
<code>(car (list one two))</code>	Ошибка: UNBOUND-VARIABLE
<code>(car (list 'one 'two))</code>	<code>one</code>

Задание 6 (2)

Функция `mystery`:

```
(defun mystery (x) (list (second x) (first x)))
```

Результаты выражений

Выражение	Результат
<code>(mystery '(one two))</code>	<code>(two one)</code>
<code>(mystery 'free)</code>	Ошибка: TYPE-ERROR
<code>(mystery (last 'one 'two))</code>	Ошибка: TYPE-ERROR
<code>(mystery 'one 'two)</code>	Ошибка: Invalid number of arguments

Ответы на вопросы

1. Базис Lisp.

Базис – это минимальный набор необходимых конструкций, с помощью которого можно реализовать задачу.

В базис Lisp входят:

- атомы и структуры (бинарные узлы)
- базисные функции и функционалы (atom, eq, car, cdr, cons, quote, cond, lambda, label, eval).

Особенности Lisp:

- как язык функционального программирования ориентирован на символьную обработку;
- и данные, и программа в Lisp представляются в виде символьных выражений – S-выражений;
- не требуется явное описание типов данных.

2. Классификация функций

- чисто математические: принимают фиксированное число аргументов;
- рекурсивные;
- специальные функции (формы): принимают произвольное число аргументов, которые по-разному обрабатываются;
- псевдофункции: создают эффект на внешних устройствах;
- функции с вариантными значениями, из которых выбирается только одно;
- функции высших порядков (функционалы): используются для синтаксически управляемых программ.

Классификация базисных функций:

- селекторы: car, cdr;
- конструкторы: cons, list;
- предикаты: null, atom, numberp, symbolp и т. д.
- сравнения: eq, eql, equal, = и т. д.

3. Список, представление и интерпретация списков.

Список – динамическая структура данных, которая может быть пустая или непустая, состоящая из головы и хвоста, который является списком. В Lisp список является частным случаем S-выражения.

Один список представляется одной списковой ячейкой, которая хранит два указателя: car-указатель (на первый элемент, или голову) и cdr-указатель (на остальной список, или хвост).

Первый элемент списка интерпретируется как имя функции, остальные элементы списка – как её аргументы. Если присутствует блокировка вычисления (функция QUOTE, или '), то первый элемент также интерпретируется как аргумент.

4. Функции CAR и CDR.

Функции CAR и CDR являются базисными функциями-селекторами, т. е. они осуществляют доступ к элементам списка. Обе функции – чисто математические: они принимают в качестве аргумента точечную пару или список.

Функция CAR переходит по car-указателю к первому элементу списка. В случае пустого списка вернёт Nil.

Функция CDR переходит по cdr-указателю к остальному списку. Если в списке меньше двух элементов, то функция вернёт Nil.

5. Назначение и отличие в работе CONS и LIST.

Функции CONS и LIST – функции-конструкторы. Обе функции могут использоваться для создания списков.

Функция CONS является базисной, чисто математической функцией. CONS создаёт бинарный узел и устанавливает его указатели на два принятых аргумента.

Примеры:

```
(cons 'A 'B) → (A . B)
(cons 'A '(B)) → (A B)
(cons 'A NIL) → (A)
```

Функция LIST является формой, т. е. принимает переменное число аргументов. LIST возвращает список, элементами которого являются аргументы функции: car-указатели ссылаются на аргументы, а cdr-указатели «сцепляют» списковые ячейки в список.

Примеры:

```
(list 'A 'B) → (A B)
(list 'A '(B)) → (A (B))
(list 'A NIL) → (A NIL)
```