

Московский государственный технический университет  
имени Н. Э. Баумана



Факультет: Информатика и системы управления

Кафедра: Программное обеспечение ЭВМ и информационные технологии

Дисциплина: Функциональное и логическое программирование

## Лабораторная работа №2

Выполнили: Никичкин А. С., Фокеев А. И.  
Группа: ИУ7–61

Москва, 2015 г.

# 1 Для указанных выражений составить диаграмму вычисления

**Задание 1.1** ( $equal\ 3\ (abs\ -3)$ )

$\longrightarrow (equal\ 3\ (abs\ -3))$ :

- вычисляется 3 к 3;

$\longrightarrow (abs\ -3)$ :

- вычисляется  $-3$  к  $-3$ ;

$\implies$  применение  $abs$  к  $-3$ ;

$\implies$  возвращаемый результат: 3;

$\implies$  применение  $equal$  к 3, 3;

$\implies$  возвращаемый результат: T.

**Задание 1.2** ( $equal\ (+\ 1\ 2)\ 3$ )

$\longrightarrow (equal\ (+\ 1\ 2)\ 3)$ :

$\longrightarrow (+\ 1\ 2)$ :

- вычисляется 1 к 1;
- вычисляется 2 к 2;

$\implies$  применение «+» к 1, 2;

$\implies$  возвращаемый результат: 3;

- вычисляется 3 к 3;

$\implies$  применение  $equal$  к 3, 3;

$\implies$  возвращаемый результат: T.

**Задание 1.3** ( $equal\ (*\ 4\ 7)\ 21$ )

$\longrightarrow (equal\ (*\ 4\ 7)\ 21)$ :

$\longrightarrow (*\ 4\ 7)$ :

- вычисляется 4 к 4;
- вычисляется 7 к 7;

$\implies$  применение «\*» к 4, 7;

$\implies$  возвращаемый результат: 28;

- вычисляется 21 к 21;

$\implies$  применение  $equal$  к 28, 21;

$\implies$  возвращаемый результат: NIL.

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

$\longrightarrow (equal (* 2 3) (+ 7 2))$ :

$\longrightarrow (* 2 3)$ :

- вычисляется 2 к 2;
- вычисляется 3 к 3;

$\implies$  применение «\*» к 2, 3;

$\implies$  возвращаемый результат: 6;

$\longrightarrow (+ 7 2)$ :

- вычисляется 7 к 7;
- вычисляется 2 к 2;

$\implies$  применение «+» к 7, 2;

$\implies$  возвращаемый результат: 9;

$\implies$  применение *equal* к 6, 9;

$\implies$  возвращаемый результат: NIL.

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

$\longrightarrow (equal (- 7 3) (* 3 2))$ :

$\longrightarrow (- 7 3)$ :

- вычисляется 7 к 7;
- вычисляется 3 к 3;

$\implies$  применение «-» к 7, 3;

$\implies$  возвращаемый результат: 4;

$\longrightarrow (* 3 2)$ :

- вычисляется 3 к 3;
- вычисляется 2 к 2;

$\implies$  применение «\*» к 3, 2;

$\implies$  возвращаемый результат: 6;

$\implies$  применение *equal* к 4, 6;

$\implies$  возвращаемый результат: NIL.

**Задание 1.6**  $(equal (abs (- 2 4)) 3)$

$\longrightarrow (equal (abs (- 2 4)) 3):$

$\longrightarrow (abs (- 2 4)):$

$\longrightarrow (- 2 4):$

• вычисляется 2 к 2;

• вычисляется 4 к 4;

$\implies$  применение «-» к 2, 4;

$\implies$  возвращаемый результат: -2;

$\implies$  применение *abs* к -2;

$\implies$  возвращаемый результат: 2;

• вычисляется 3 к 3;

$\implies$  применение *equal* к 2, 3;

$\implies$  возвращаемый результат: NIL.

## 2 Написать функцию и составить для неё диаграмму вычисления

**Задание 2.1** Функция вычисляет гипотенузу прямоугольного треугольника по заданным катетам.

```
1 (defun hypotenuse-rect-triangle (leg1 leg2)
2   "By Pythagorean theorem"
3   (sqrt (+ (* leg1 leg1)
4             (* leg2 leg2))))
```

→ (*hypotenuse-rect-triangle*  $q$   $w$ )

- вычисляется  $q$  к  $q$ ;
- вычисляется  $w$  к  $w$ ;

⇒ применение *hypotenuse-rect-triangle* к  $q$ ,  $w$ ;

- создаётся переменная *leg1* со значением  $q$ ;
- создаётся переменная *leg2* со значением  $w$ ;

→ (*sqrt* (+ (\* *leg1* *leg1*) (\* *leg2* *leg2*))):

→ (+ (\* *leg1* *leg1*) (\* *leg2* *leg2*)):

→ (\* *leg1* *leg1*):

- вычисляется *leg1* к  $q$ ;
- вычисляется *leg1* к  $q$ ;

⇒ применение «\*» к  $q$ ,  $q$ ;

⇒ возвращаемый результат:  $q^2$ ;

→ (\* *leg2* *leg2*):

- *leg2* к  $w$ ;
- *leg2* к  $w$ ;

⇒ применение «\*» к  $w$ ,  $w$ ;

⇒ возвращаемый результат:  $w^2$ ;

⇒ применение «+» к  $q^2$ ,  $w^2$ ;

⇒ возвращаемый результат:  $q^2 + w^2$ ;

⇒ применение *sqrt* к  $q^2 + w^2$ ;

⇒ возвращаемый результат:  $\sqrt{q^2 + w^2}$ .

⇒ возвращаемый результат:  $\sqrt{q^2 + w^2}$ .

**Задание 2.2** Функция вычисляет объём прямоугольного параллелепипеда по 3-м его сторонам.

```
1 (defun volume-rect-parallepiped (leg1 leg2 leg3)
2   "Volume of rectangle parallepiped by 3 sides"
3   (* leg1 leg2 leg3))
```

→ (*volume-rect-parallepiped* *q w e*);

- вычисляется *q* к *q*;
- вычисляется *w* к *w*;
- вычисляется *e* к *e*;

⇒ применение *volume-rect-parallepiped* к *q, w, e*:

- создаётся переменная *leg1* со значением *q*;
- создаётся переменная *leg2* со значением *w*;
- создаётся переменная *leg3* со значением *e*;

→ (*\* leg1 leg2 leg3*):

- вычисляется *leg1* к *q*;
- вычисляется *leg2* к *w*;
- вычисляется *leg3* к *e*;

⇒ применение «\*» к *q, w, e*;

⇒ возвращаемый результат:  $q \cdot w \cdot e$ ;

⇒ возвращаемый результат:  $q \cdot w \cdot e$ .

**Задание 2.3** Функция по заданной гипотенузе и катету, вычисляет другой катет прямоугольного треугольника.

```

1 (defun leg-of-rect-triangle (leg hypotenuse)
2   (sqrt (- (* hypotenuse hypotenuse)
3             (* leg leg))))

```

→ (*leg-of-rect-triangle* *b c*):

- вычисляется *b* к *b*;
- вычисляется *c* к *c*;

⇒ применение *leg-of-rect-triangle*:

- создаётся переменная *leg* со значением *b*;
- создаётся переменная *hypotenuse* со значением *c*;

→ (*sqrt* (*-* (*\** *hypotenuse hypotenuse*) (*\** *leg leg*)))):

→ (*-* (*\** *hypotenuse hypotenuse*) (*\** *leg leg*)):

→ (*\** *hypotenuse hypotenuse*):

- вычисляется *hypotenuse* к *c*;
- вычисляется *hypotenuse* к *c*;

⇒ применение «*\**» к *c*, *c*;

⇒ возвращаемый результат:  $c^2$ ;

→ (*\** *leg leg*):

- вычисляется *leg* к *b*;
- вычисляется *leg* к *b*;

⇒ применение «*\**» к *b*, *b*;

⇒ возвращаемый результат:  $b^2$ ;

⇒ применение «*-*» к  $c^2$ ,  $b^2$ ;

⇒ возвращаемый результат:  $c^2 - b^2$ ;

⇒ применение *sqrt* к  $c^2 - b^2$ ;

⇒ возвращаемый результат:  $\sqrt{c^2 - b^2}$ ;

⇒ возвращаемый результат:  $\sqrt{c^2 - b^2}$ .

**Задание 2.4** Функция вычисляет площадь трапеции по её основаниям и высоте

```
1 (defun trapezoid-area (parallel-side1 parallel-side2 height)
2   "Area of trapezoid by 2 parallel sides and height"
3   (* 0.5
4     height
5     (+ parallel-side1 parallel-side2)))
```

→ (*trapezoid-area* *a b h*):

- вычисляется *a* к *a*;
- вычисляется *b* к *b*;
- вычисляется *h* к *h*;

⇒ применение *trapezoid-area* к *a*, *b*, *h*;

- создаётся переменная *parallel-side1* со значением *a*;
- создаётся переменная *parallel-side2* со значением *b*;
- создаётся переменная *height* со значением *h*;

→ (*\* 0.5 height (+ parallel-side1 parallel-side2)*):

- вычисляется 0.5 к 0.5;
- вычисляется *height* к *h*;

→ (*+ parallel-side1 parallel-side2*):

- вычисляется *parallel-side1* к *a*;
- вычисляется *parallel-side2* к *b*;

⇒ применение «+» к *a*, *b*;

⇒ возвращаемый результат: *a + b*;

⇒ применение «\*» к 0.5, *h*, *a + b*;

⇒ возвращаемый результат:  $0.5 \cdot h \cdot (a + b)$ ;

⇒ возвращаемый результат:  $0.5 \cdot h \cdot (a + b)$ .



### 3 Вычислить результаты выражений

#### Задание 3.1

```
(list 'a 'b c) => Unbound variable C
```

#### Задание 3.2

```
(cons 'a (b c)) => Unbound variable C
```

#### Задание 3.3

```
(cons 'a '(b c)) => (A B C)
```

#### Задание 3.4

```
(caddr '(1 2 3 4 5)) => 3
```

#### Задание 3.5

```
(cons 'a 'b 'c) => Too many arguments
```

#### Задание 3.6

```
(list 'a (b c)) => Unbound variable C
```

#### Задание 3.7

```
(list a '(b c)) => Unbound variable A
```

#### Задание 3.8

```
(list (+ 1 (length '(1 2 3)))) => (4)
```

#### Задание 3.9

```
(cons 3 (list 5 6)) => (3 5 6)
```

#### Задание 3.10

```
(list 3 'from 9 'gives (- 9 3)) => (3 FROM 9 GIVES 6)
```

#### Задание 3.11

```
(+ (length '(1 foo 2 too)) (car '(21 22 23))) => 25
```

**Задание 3.12**

```
(cdr '(cons is short for ans)) => (IS SHORT FOR ANS)
```

**Задание 3.13**

```
(car (list one two)) => Unbound variable ONE
```

**Задание 3.14**

```
(cons 3 '(list 5 6)) => (3 LIST 5 6)
```

**Задание 3.15**

```
(car (list 'one 'tow)) => ONE
```

**Задание 3.16**

```
(list 'cons t NIL) => (CONST T NIL)
```

**Задание 3.17**

```
(eval (eval (list 'cons t NIL))) => Undefined function T
```

**Задание 3.18**

```
(apply #'cons '(t NIL)) => (T)
```

**Задание 3.19**

```
(list 'eval NIL) => (EVAL NIL)
```

**Задание 3.20**

```
(eval (list 'cons t nil)) => (T)
```

**Задание 3.21**

```
(eval NIL) => NIL
```

**Задание 3.22**

```
(eval (list 'eval NIL)) => NIL
```

**Задание 3.23**

```
(equal (+ 1 2) 3) => T
```

### Задание 3.24

```
(equal (abs (- 2 4)) 3) => NIL
```

### Задание 3.25

```
(equal (abs (- 2 4)) 2) => T
```

### Задание 3.26

```
(eq1 (abs (- 2 4)) 2.0) => NIL
```

### Задание 3.27

```
(eq1 (abs (- 2 4)) 2) => T
```

### Задание 3.28

```
(eq1 (list (abs (- 2 4))) '(2.0)) => NIL
```

### Задание 3.29

```
(equal (list (abs (- 2 4))) '(2)) => T
```

### Задание 3.30

```
(equalp (list (abs (- 2 4))) '(2.0)) => T
```

## 4 Написать функцию

**Задание 4.1** Функция от двух списков-аргументов, которая возвращает T, если первый аргумент имеет большую длину.

```
1 (defun longer-then (arg1 arg2)
2   (> (length arg1)
3     (length arg2)))
```

**Задание 4.2** Функция переводит температуру из системы Фаренгейта в температуру по Цельсию.

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

## 5 Исследование функции

Имеется следующая функция

```
1 (defun mystery (x)
2   (list (second x)
3         (first x)))
```

Необходимо вычислить результаты выражений.

### Задание 5.1

```
(mystery '(one two)) => (TWO ONE)
```

### Задание 5.2

```
(mystery 'free) => value FREE is not LIST
```

### Задание 5.3

```
(mystery (last 'one 'two)) => no list for LAST
```

### Задание 5.4

```
(mystery 'one 'two) => too many arguments
```

## 6 Заключение

В данной лабораторной работе было рассмотрено составление *диаграммы вычисления*, а так же изучены функции EQL, EQUAL, EQUALP, ABS, SQRT, LENGTH, EVAL, APPLY, FIRST, SECOND.