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 по лабораторной работе №2 по курсу "Функциональное и логическое программирование"

Тема Определение функций пользователя

Студент Цветков И.А.

Группа ИУ7-63Б

Оценка (баллы)

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

Москва — 2022 г.

1 Практические задания

1.1 Задание 1

Условие: составить диаграмму вычисления выражений.

Решение приложено к отчету.

1.2 Задание 2

Условие: написать функцию, вычисляющую гипотенузу прямоугольного треугольника по заданным катетам и составить диаграмму её высчиления.

```
1    (defun hypotenuse (a b) (sqrt (+ a a) (*b b)))); (HYPOTENUSE 3 4) -> 5.0
```

Диаграмма приложена к отчету.

1.3 Задание 3

Условие: написать функцию, вычисляющую объем параллелепипеда по 3-м его сторонам, и составить диаграмму ее вычисления.

```
1    (defun volume (a b c)  a (*b c))); (volume 4 4 4) -> 64
```

Диаграмма приложена к отчету.

1.4 Задание 4

Условие: каковы результаты вычисления следующих выражений? (объяснить возможную ошибку и варианты ее устранения)

```
1    (list 'a c) ; The variable C is unbound.
2    (cons 'a (b c)) ; The variable C is unbound.
3    (cons 'a '(b c)) ; (A B C)
4    (caddy (1 2 3 4 5)) ; Undefined function: CADDY
5    (cons 'a 'b 'c) ; invalid number of arguments: 3
6    (list 'a (b c)) ; The variable C is unbound.
```

```
7     (list a '(b c)) ; The variable A is unbound.
8     (list (+ 1 '(length '(1 2 3)))) ; The value (LENGTH '(1 2 3)) is not of
          type NUMBER
```

Исправления ошибок представлены ниже.

```
1     (list 'a 'c) ; (A C)
2     (cons 'a '(b c)) ; (A B C)
3     (cons 'a '(b c)) ; (A B C)
4     (caddr '(1 2 3 4 5)) ; 3
5     (cons 'a 'b) ; (A B)
6     (list 'a '(b c)) ; (A (B C))
7     (list 'a '(b c)) ; (A (B C))
8     (list (+ 1 (length '(1 2 3)))) ; (4)
```

1.5 Задание 5

Условие: написать функцию `longer_then` от двух списков-аргументов, которая возвращает T, если первый аргумент имеет большую длину.

```
1     (defun longer (a b) (cond ((> (length a) (length b)) T) (Nil)))
2     ; (longer '(a b) '(a)) -> T
3     ; (longer '(a) '(a)) -> Nil
4     ; (longer '(a) '(a b)) -> Nil
```

1.6 Задание 6

Условие: каковы результаты вычисления следующих выражений?

```
1     (cons 3 (list 5 6)) ; (3 5 6)
2
3     (list 3 'from 9 'lives (- 9 3)) ; (3 FROM 9 LIVES 6)
4
5     (+ (length for 2 too))(car '(21 22 23))) ; The variable FOR is unbound.
6     ; Fix: (+ (length '(for 2 too))(car '(21 22 23))) ; 24
7
8     (cdr '(cons is short for ans)) ; (IS SHORT FOR ANS)
9
10    (car (list one two)) ; The variable ONE is unbound.
11    ; Fix: (car (list 'one 'two)) -> ONE
12
13    (cons 3 '(list 5 6)) ; (3 LIST 5 6)
14
```

```
15    (car (list 'one 'two)) ; ONE
```

1.7 Задание 7

Условие: дана функция `(defun mystery (x) (list (second x) (first x)))`.Какие результаты вычисления следующих выражений?

```
1    (mystery (one two)) ; The variable TWO is unbound.
2    ; Fix: (mystery '(one two)) -> (TWO ONE)
3    (mystery (one 'two)) ; The function COMMON-LISP-USER::ONE is undefined.
4    (mystery (last one two)) ; The variable ONE is unbound.
5    (mystery free) ; The variable FREE is unbound.
```

1.8 Задание 8

Условие: написать функцию, которая переводит температуру в системе Фаренгейта 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?

```
1    (defun f-to-c (temp)  5/9 (- temp 32.0))); (f-to-c 451) -> 232.77779
```

Роман бы назывался «+232 по Цельсию».

1.9 Задание 9

Условие: Что получится при вычисления каждого из выражений?

```
1    (list 'cons t nil) ; (CONS T NIL)
2    (eval (list 'cons t NIL)) ; (T)
3    (eval (eval (list 'cons t NIL))) ; The function COMMON-LISP:T is undefined.
4    (apply #cons ''(t NIL)) ; illegal complex number format: #CONS
5    ; Fix: (apply #'cons '(t NIL)) -> (T)
6    (eval Nil) ; Nil
7    (list 'eval Nil) ; (EVAL NIL)
8    (eval (list 'eval Nil)) ; NIL
```

1.10 Дополнительное задание 1

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

```
1 (defun catet (b c) (sqrt (- c c) (*b b)))); (catet 4 5) -> 3.0
```

Диаграмма приложена к отчету.

1.11 Дополнительное задание 2

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

```
1 (defun trap\_square (a b h) (+ a b) 1/2 h)); (trap_square 2 3 1) -> 2.5
```

Диаграмма приложена к отчету.

2 Ответы на вопросы к лабораторной работе

2.1 Базис Lisp.

Базис языка – минимальный набор конструкций языка и структур данных, с помощью которых можно решить любую задачу.

Базис состоит из:

1. атомы и структуры (представляющиеся бинарными узлами);
2. базовые (несколько) функций и функционалов: встроенные примитивные функции (`atom, eq, cons, car, cdr`); специальные функции и функционалы (`quote, cond, lambda, eval, apply, funcall`).

2.2 Классификация функций.

Функции в Lisp классифицируют следующим образом:

- «чистые» математические функции;
- рекурсивные функции;
- специальные функции — формы (от вызова к вызову может меняться количество аргументов, или обрабатываться по-разному);
- псевдофункции (создают эффект на внешнем устройстве);
- функции с вариативными значениями, из которых выбирается 1;
- функционалы (в качестве аргумента – функцмя, или возаращает в качестве результата функцию).

По назначению функции разделяются следующим образом:

1. конструкторы — создают значение (`cons`, например);

2. селекторы — получают доступ по адресу (`car`, `cdr`);

3. предикаты — возвращают `Nil`, `T`.

2.3 Способы создания функций.

Построить функцию можно с помощью Lambda-выражения (базисный способ)

Lambda-определение безымянной функции:

```
1 (lambda <Lambda-список> <форма>),
```

где Lambda-список – список аргументов, а форма – тело функции.

Lambda-вызов функции:

```
1 (<Lambda-выражение> <формальные параметры>)
```

Функции с именем. В таких функциях `defun` связывает символьный атом с Lambda-определением:

```
1 (defun f <Lambda-выражение>)
```

Упрощенное определение:

```
1 (defun f(x1, ..., xk) (<формы>))
```

2.4 Функции CAR, CDR.

`car` – функция получения первого элемента точечной пары, `cdr` – функция получения второго элемента точечной пары.

Примеры:

S-выражение	Результат выполнения `car`	Результат выполнения `cdr`
(A . B)	A	B
((A . B) . C)	(A . B)	C
(A . (B . C))	A	(B . C)

2.5 Назначение и отличие в работе Cons и List.

`cons` – имеет фиксированное количество аргументов (два). В случае, когда аргументами являются атомы создает точечную пару. В случает, когда первый аргумент атом а второй список, атом становится головой, а второй аргумент (список) становится хвостом.

```
1 (cons 'a 'b)          ; (A . B)
2 (cons 'a '(b c d))    ; (A B C D)
3 (cons 'a 'b 'c)       ; Error (invalid number of arguments: 3)
```

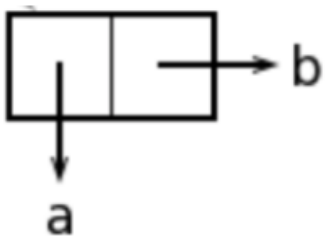

Рисунок 2.1 – Результат работы (cons(A B))

`list` – не имеет фиксированное количество аргументов. Создает список, у которого голова - это первый аргумент, хвост - все остальные аргументы.

```
1 (list 'a 'b)      ; (A B)
2 (cons 'a 'b 'c)   ; (A B C)
```

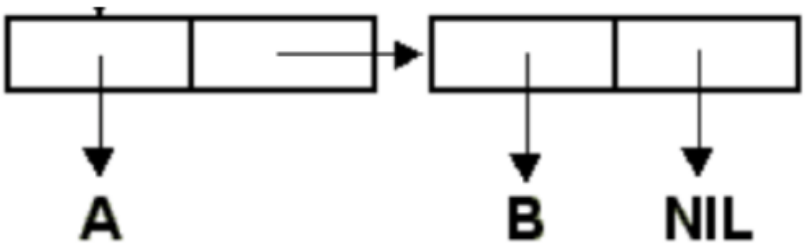

Рисунок 2.2 – Результат работы (list(A B))