

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ  
БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ  
УНИВЕРСИТЕТ»

**М.А. Сергиенко**

**ЛАБОРАТОРНЫЙ ПРАКТИКУМ  
ПО РАЗРАБОТКЕ ЭКСПЕРТНЫХ СИСТЕМ  
С ПОМОЩЬЮ CLIPS и JAVA**

*Учебно-методическое пособие*

Воронеж  
Издательский дом ВГУ  
2017

## Содержание

Введение.....	4
Лабораторная работа № 1. Процедурное программирование .....	7
Лабораторная работа № 2. Факты и правила .....	11
Лабораторная работа № 3. Средства сопоставления с шаблонами .....	19
Лабораторная работа № 4. Объектно-ориентированное программирование .....	35
Лабораторная работа № 5. Решение задач на планирование.....	39
Лабораторная работа № 6. Разработка ЭС с UI .....	45
Библиографический список .....	46

Сдача каждой задачи сопровождается предоставлением отчета, содержащего:

- постановку задачи;
- реализацию:
  - о средства реализации;
  - о алгоритм реализации;
  - о интерфейс пользователя;
- листинг программы;
- список используемой литературы.

В конце приводится список литературы, используемый при подготовке учебного пособия.

Для демонстрации примеров используется CLIPS версии 6.3.

Для запуска программы в CLIPS необходимо:

- 1) скопировать код программы в любой текстовый редактор и сохранить его с расширением **.clp**;
- 2) загрузить этот файл в среду CLIPS с помощью меню **File→Load (Ctrl+L)**;
- 3) выполнить команду **reset**:
  - либо введите эту команду в командной строке интерпретатора  
CLIPS> (reset) ,
  - либо выберите в меню команду **Execution→Reset (CTRL+U)**;
- 4) запустить интерпретатор с помощью команды **run**:
  - либо введите эту команду в командной строке интерпретатора  
CLIPS> (run) ,
  - либо выберите в меню команду **Execution→Run (CTRL+R)**.

Разработанное методическое пособие предназначено для бакалавров и магистров, обучающихся по программе высшего профессионального образования, ориентированного на подготовку в сфере IT-технологий.

# Лабораторная работа № 1

## Процедурное программирование

**Цель работы:** изучение основных типов данных и принципов процедурного программирования в CLIPS.

### Теоретический минимум

1. Типы данных.
  - 1.1. Числовые типы.
    - 1.1.1. Тип *float*.
    - 1.1.2. Тип *integer*.
  - 1.2. Символьные типы.
    - 1.2.1. Тип *string*.
    - 1.2.2. Тип *symbol*.
  - 1.3. Ссылочные типы.
    - 1.3.1. Тип *fact-address*.
    - 1.3.2. Тип *instance-name*.
    - 1.3.3. Тип *instance-address*.
2. Процедурное программирование.
  - 2.1. Процедурные функции.
    - 2.1.1. Функция *if*.
    - 2.1.2. Функция *while*.
    - 2.1.3. Функция *switch*.
    - 2.1.4. Функция *loop-for-count*.
    - 2.1.5. Функция *break*.
    - 2.1.6. Функция *halt*.
  - 2.2. Конструкция *deffunction*.
    - 2.2.1. Функция *return*.
    - 2.2.2. Рекурсия.
    - 2.2.3. Предварительные объявления.
    - 2.2.4. Параметр с подстановочным символом.
    - 2.2.5. Команды для работы с конструкцией *deffunction*.

**Рекомендуемая литература:** [5, гл. 4, 8; 1, гл. 7, 10; 3, гл. 3, 6].

## Пример 1

Реализовать функцию, позволяющую вычислить длину гипотенузы прямоугольного треугольника с помощью теоремы Пифагора.

Допустим, что  $a$  и  $b$  – стороны, образующие прямой угол, а  $c$  – гипотенуза. В таком случае согласно теореме Пифагора можно записать следующее:

$$c = \sqrt{a^2 + b^2}.$$

Преобразование этой формулы в конструкцию *deffunction* представлено в листинге.

## Листинг программы

```
(deffunction hypotenuse (?a ?b)
  (** (+ (* ?a ?a) (* ?b ?b)) 0.5)
)
```

или

```
(deffunction hypotenuse (?a ?b)
  (bind ?result (** (+ (* ?a ?a) (* ?b ?b)) 0.5))
  (return ?result)
)
```

Результат работы программы представлен на рис. 1.1.

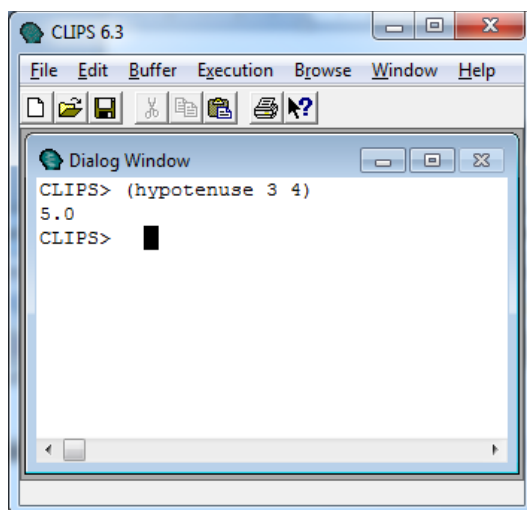


Рис. 1.1. Результат работы программы

## Пример 2

Реализовать рекурсивную функцию, позволяющую вычислить факториал от положительного целого числа.

$$factorial(n) = \begin{cases} n \cdot factorial(n-1), & \text{если } n > 1 \\ 1, & \text{если } n = 0 \end{cases}$$

Преобразование этой формулы в конструкцию *deffunction* представлено в листинге.

## Листинг программы

```
(deffunction factorial (?n)
  (if (or (not (integerp ?n)) (< ?n 0))
      then
        (printout t "argument is not valid" crlf)
      else
        (if (= ?n 0)
            then
              1
            else
              (* ?n (factorial (- ?n 1)))
        )
  )
)
```

Результат работы программы представлен на рис. 1.2.

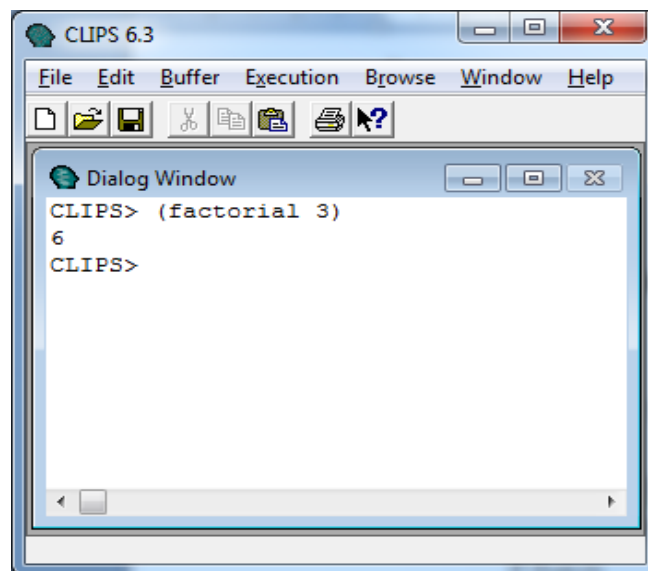


Рис. 1.2. Результат работы программы

## Задачи

1. Напишите конструкцию *deffunction*, в которой не используется рекурсия для вычисления факториала целого числа N.
2. Напишите конструкцию *deffunction*, которая определяет, содержится ли одна строка в другой.
3. Напишите конструкцию *deffunction*, которая определяет количество вхождений одной строки в другой строке.
4. Напишите конструкцию *deffunction* для определения, является ли заданное число простым.
5. Напишите конструкцию *deffunction*, которая определяет все простые числа от 1 до указанного целого числа и возвращает эти простые числа в виде многозначного значения.
6. Напишите конструкцию *deffunction*, которая определяет все нечетные числа от 1 до указанного целого числа и возвращает эти нечетные числа в виде многозначного значения.
7. Напишите конструкцию *deffunction*, которая определяет все четные числа от 1 до указанного целого числа и возвращает эти четные числа в виде многозначного значения.
8. Напишите конструкцию *deffunction*, которая принимает от нуля и больше параметров и возвращает многозначное значение, содержащее значения параметров в обратном порядке.
9. Напишите конструкцию *deffunction*, которая преобразовывает двоичную строку, состоящую из нулей и единиц, в десятичное число.
10. Напишите конструкцию *deffunction* для сравнения двух целых чисел:

$$\text{int-compare}(a, b) = \begin{cases} -1, & \text{если } a < b \\ 0, & \text{если } a = b \\ 1, & \text{если } a > b \end{cases}$$