CLIPS – среда разработки интеллектуальных систем

Общие сведения о CLIPS

Особенности языка CLIPS

CLIPS (C Language Integrated Production System) является одним из распространенных инструментальных средств разработки экспертных систем (ЭС).
Представляя собой логически полную среду, содержащую встроенный редактор, интерпретатор и средства отладки, CLIPS является оболочкой ЭС.
Разработчиком CLIPS является Национальное Аэрокосмическое Агентство США. Первая версия системы вышла в 1984 году, текущая версия -6.1.

Особенности языка CLIPS

CLIPS поддерживает функциональную, процедурную и объектно-ориентированную парадигмы (стили) программирования, а также продукционную модель знаний.
CLIPS поддерживает продукционную модель представления знаний и содержит три основных элемента этой модели:

1. список фактов,
2. базу знаний,
3. блок вывода.

Особенности языка CLIPS

Принципиальным отличием данной системы от аналогов является то, что она полностью реализована на языке С. Причем исходные тексты ее программ опубликованы в сети Интернет.

Основные элементы программирования

CLIPS предоставляет три основных элемента для написания программ:

- простые типы данных;
- функции для манипулирования данными;
- конструкторы.

Простые типы данных

Для представления информации в CLIPS предусмотрено восемь простых типов данных: float, integer, symbol, string, external-address, fact address, instance-name и instance-address.
Для представления числовой информации используются типы float и integer, для символьной – symbol и string.

Числовые типы данных

При записи числа могут использоваться только цифры (0-9), десятичная точка (.), знак (+) или (-) и (e) при экспоненциальном представлении.
Число сохраняется либо как целое, либо как действительное. Любое число, состоящее только из цифр, перед которыми может стоять знак, сохраняется как целое (тип integer представляется внутри CLIPS как тип языка C long integer).

Числовые типы данных

Все остальные числа сохраняются как действительные (float - C double float).
Количество значащих цифр зависит от аппаратной реализации. В этой же связи могут возникать ошибки округления.
Особую осторожность необходимо проявлять при сравнении чисел с плавающей точкой, а также при сравнении с ними целых чисел.

Тип symbol

Последовательность символов, которая не удовлетворяет числовым типам, обрабатывается как тип данных symbol.
Тип данных symbol в CLIPS – последовательность символов, состоящая из одного или нескольких любых печатных символов кода ASCII. Как только в последовательности символов встречается символ - разделитель, symbol заканчивается.

Разделители в константах типа symbol

Следующие символы служат разделителями:
любой непечатный ASCII символ (включая пробел, символ табуляции, CR, LF), двойные кавычки,"(", ")", "&", "|", "<","~", ";". Символы-разделители не могут включаться в symbol за исключением символа "<", который может быть первым символом в symbol.
Кроме того, symbol не может начинаться с символа "?“ или последовательности символов "$?", поскольку эти символы зарезервированы для переменных.

Примеры констант типа symbol

Ниже приведены примеры выражений символьного типа:

foo Hello B76-HI bad_value

127A 742-42-42 @+=-% Search

Тип string

Тип данных string -это последовательность символов, состоящая из нуля и более печатных символов и заключенная в двойные кавычки. Если внутри строки встречаются двойные кавычки, то перед ними необходимо поместить символ (\). То же справедливо и для самого (\).

Примеры констант типа string

Несколько примеров:

"foo" "a and b" "I number" "a\"quote"

Отметим, что строка "abcd" не тоже самое, что abcd. Они содержат одинаковые наборы символов, но являются экземплярами различного типа.

Определение функции

Под функцией в CLIPS понимается фрагмент исполняемого кода, с которым связано уникальное имя и который возвращает полезное значение или имеет полезный побочный эффект (например, вывод информации на экран).

Типы функций

Существует несколько типов функций:

- Определенные пользователем внешние функции;
- Системные (внутренние) функции;
- Функции, определенные в среде CLIPS с помощью конструктора deffunction;
- Родовые функции.

Внешние функции

Пользовательские внешние и системные функции - это фрагменты кода, написанные на внешних языках (например, на C) и связанные со средой CLIPS.
Системными называются те функции, которые были определены изначально внутри среды CLIPS. Пользовательскими называются функции, которые были определены вне CLIPS.

Стандартные функции

Стандартные функции являются встроенными в среде CLIPS. CLIPS обладают широким набором встроенных функций. К ним относятся:

- Логические и математические функции;
- Функции обработки строк;
- Функции обработки составных величин;
- Функции ввода-вывода;
- Процедурные функции;
- Функции поддержки объектно-ориентированных возможностей;
- Конструкторы и т.д..

Примеры стандартных арифметических и математических функций

+	Сложение
-	Вычитание
*	Умножение
/	Деление
* *	Возведение в степень
abs	Определение абсолютного значения
sqrt	Вычисление квадратного корня
mod	Взятие по модулю
min	Нахождение минимума
max	Нахождение максимума

Конструкторы

В CLIPS существует несколько описывающих конструкций:
defmodule, defrule, deffacts, deftemplate, defglobal, deffunction, defclass, definstances, defmessage-handler, defgeneric.
При записи все они заключаются в скобки.
Определение конструкции отличается от вызова функции главным образом по производимому эффекту.

Отличие конструктора от функции

Обычно вызов функции оставляет состояние среды CLIPS без изменений (за рядом исключений, когда речь идет о функциях сброса, очистки, открытия файла и т.п.). Определение конструкции, напротив, в точности направлено на изменение состояния среды путем внесения изменений в базу знаний CLIPS. В отличие от функций конструкторы никогда не возвращают значений.

Переменные

Как и в других языках программирования, в CLIPS для хранения значений используются переменные. В отличие от констант, которые являются статическими, или неизменными, содержание переменной динамично и изменяется по мере того, как изменяется присвоенное ей значение.
Идентификатор переменной всегда начинается с вопросительного знака, за которым следует ее имя. В общем случае формат переменной выглядит следующим образом:
?<variable-name>

Переменные

Примеры переменных:
?x ?sensor ?noun ?color
Перед использованием переменной ей необходимо присвоить значение. Все переменные, кроме глобальных, считаются локальными и могут использоваться только в рамках описания конструкции. К этим локальным переменным можно обращаться внутри описания, но они не определены вне него.

Определение функций

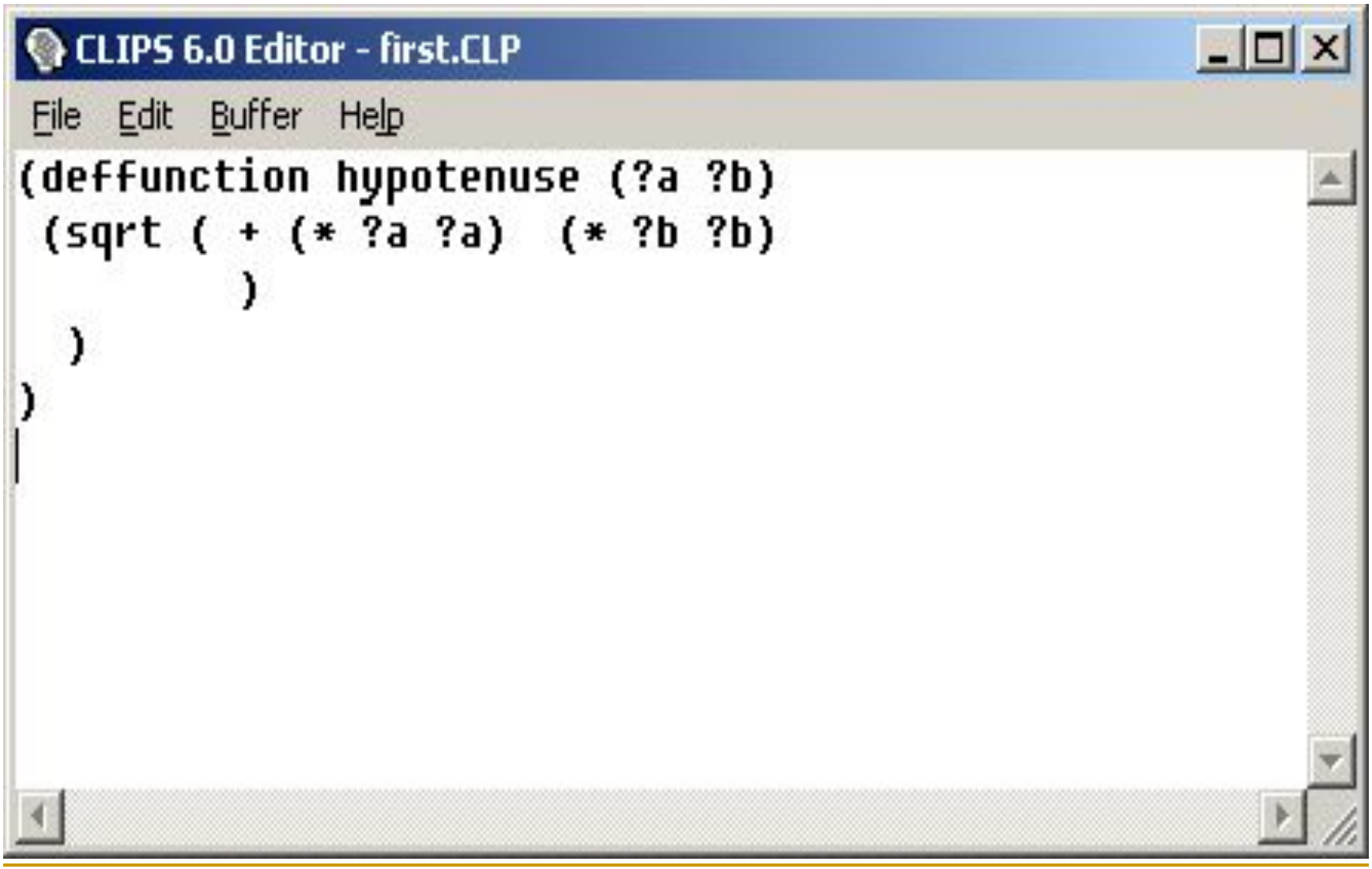
Функции в языке CLIPS определяются с помощью конструктора deffunction.
Определение функции аналогично определению функции в языке LISP. Существенное отличие состоит в том, что переменные должны иметь префикс “?” .
Формат определения функции в CLIPS следующий:
(deffunction <имя функций> (<аргумент> ... <аргумент>)
<выражение>
...
<выражение>)
Функция возвращает результат последнего выражения в списке. Иногда выполнение функции имеет побочные эффекты.

Пример определения функции

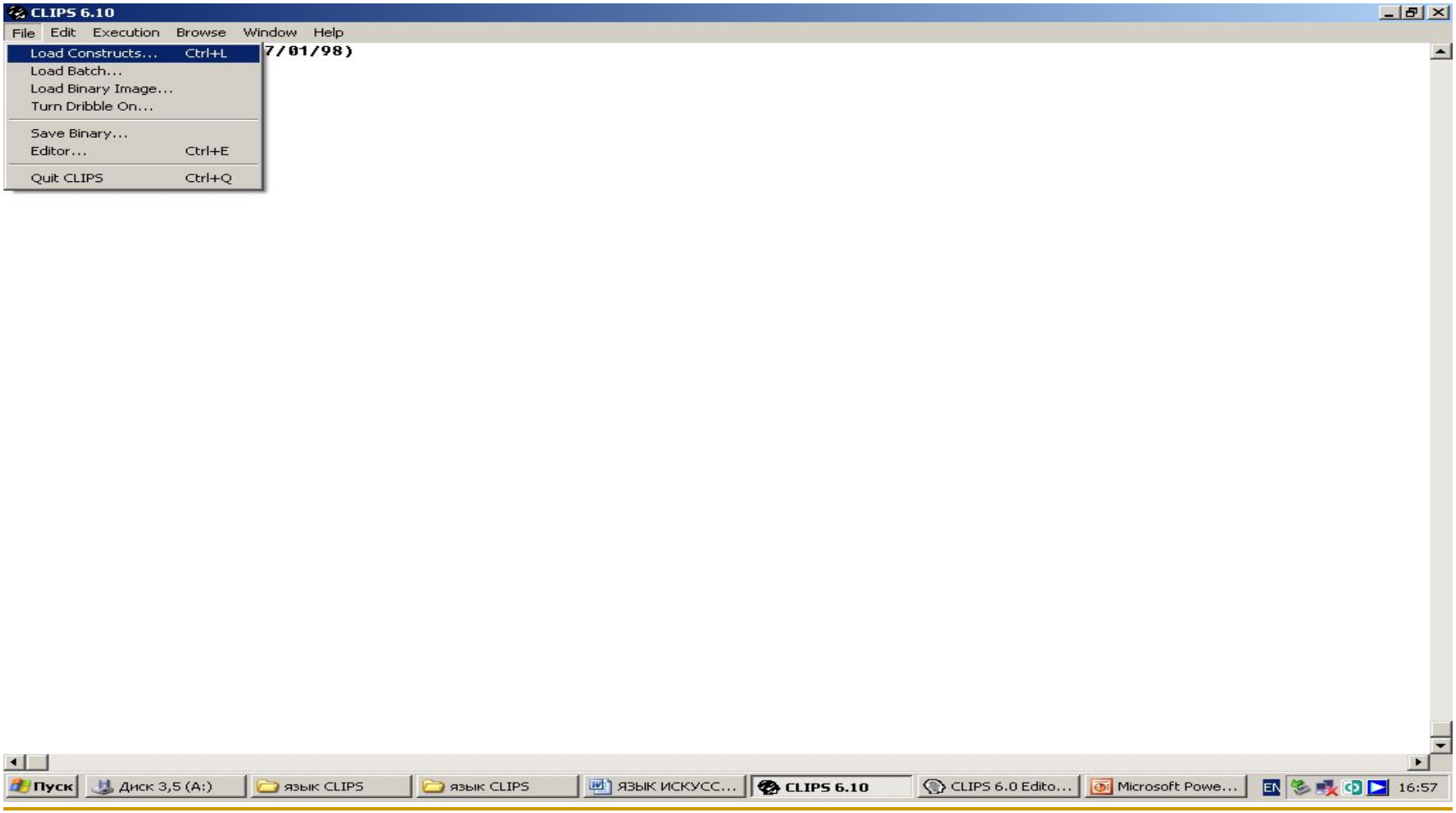
Функция определения длины гипотенузы треугольника в языке CLIPS определяются с помощью конструктора deffunction следующим образом:

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

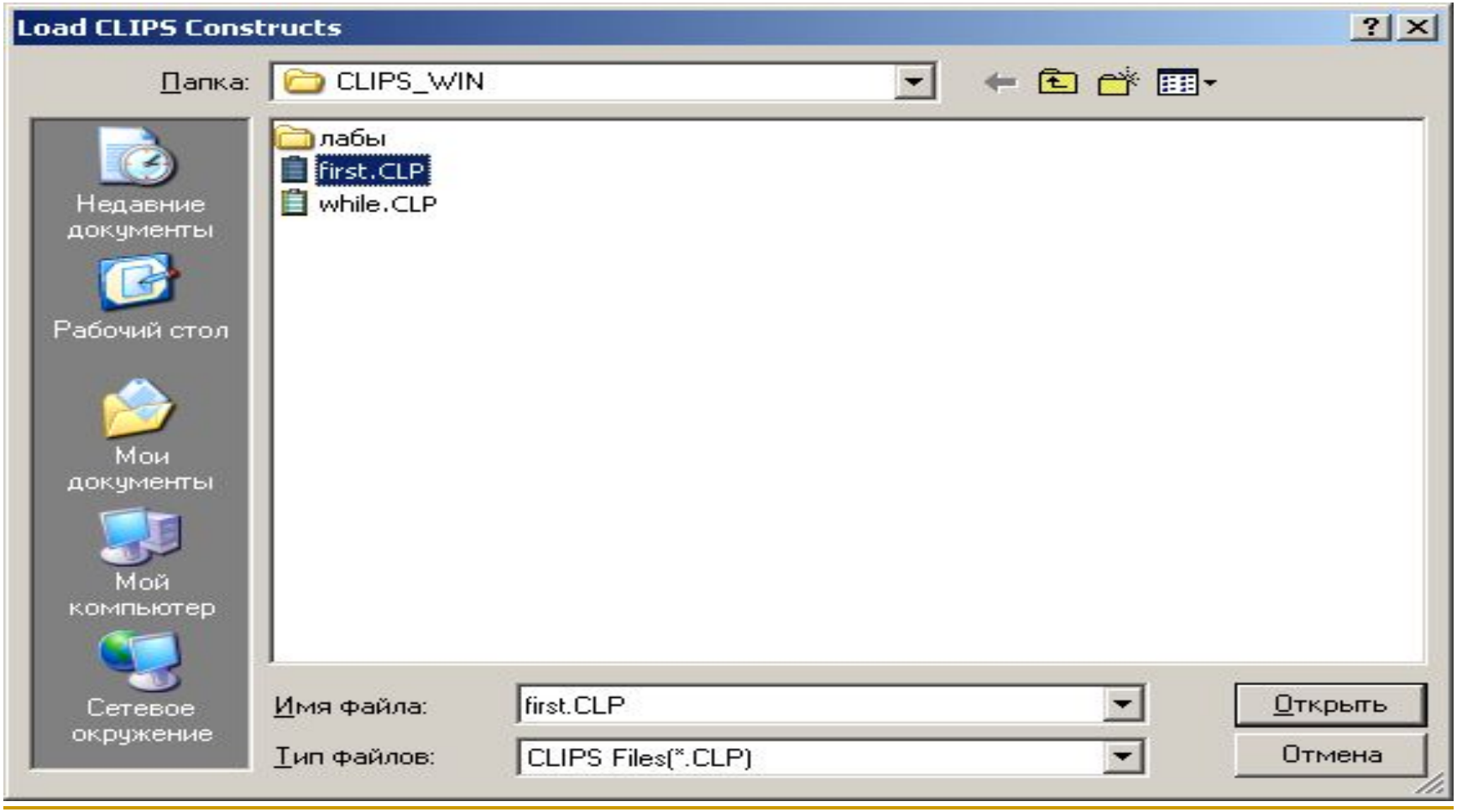
Пример создания функции в редакторе среды CLIPS



Загрузка конструкции deffunction в среду CLIPS



Загрузка конструкции deffunction в среду CLIPS



Загрузка конструкции deffunction в среду CLIPS и выполнение функции