**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ России**

**Федеральное государственное автономное образовательное учреждение высшего профессионального образования**

**«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**Инженерно-технологическая Академия**

**Институт компьютерных технологий и информационной безопасности**

**Кафедра Систем Автоматизированного Проектирования**

**Отчет по лабораторной работе № 1**

**на тему: «Проектирование баз знаний экспертных систем»**

по дисциплине «Методы машинного обучения»

Выполнили:

студенты группы КТбо4-4

Батагов С.С.\_\_\_\_\_\_\_\_\_\_\_\_\_

Москаленко М.В.\_\_\_\_\_\_\_\_\_\_\_\_\_

Проверил:

доцент кафедры САПР

Лебедев О.Б. \_\_\_\_\_\_\_\_\_\_\_\_\_

Таганрог 2022

**СОДЕРЖАНИЕ**

[1. ЦЕЛЬ РАБОТЫ 3](#_Toc122197093)

[2. ПОСТАНОВКА ЗАДАЧИ 3](#_Toc122197094)

[3 РАЗРАБОТКА ЭКСПЕРТНОЙ СИСТЕМЫ 5](#_Toc122197095)

[3.1 Разработка правил системы 5](#_Toc122197096)

[3.2 Работоспособность системы 8](#_Toc122197097)

[4. ВЫВОД 10](#_Toc122197098)

[ПРИЛОЖЕНИЕ А 11](#_Toc122197099)

# 1. ЦЕЛЬ РАБОТЫ

Целью данной лабораторной работы является: изучение процесса проектирования баз знаний, используя среду программирования экспертных систем CLIPS, анализ продукционной модели представления знаний, механизмов прямого и обратного логического вывода, фреймовой модели представления знаний, а также практическая реализация моделей знаний и механизмов вывода знаний.

# 2. ПОСТАНОВКА ЗАДАЧИ

**Задача**: диагностика заболеваний

Экспертная система диагностики заболеваний на языке CLIPS должна осуществлять помощь пользователю в определении диагноза болезни на основе некоторых симптомов. Диагностика должна осуществляться путем опроса пользователя на наличие определенных симптомов. Такая база знаний фактически представляет собой древовидную структуру, где в качестве узлов представлены симптомы (вопросы вида да/нет), а в качестве листьев заболевания. Количество диагностируемых заболеваний должно быть не менее 5. Данные по заболеваниям и их симптомам должны быть взяты из какой-либо книги или справочника. Реализовать прямой и обратный вывод, использовать продукционную и фреймовую модели представления знаний. Постройте набор порождающих правил и разработайте CLIPS-программу для медицинской экспертной системы диагностики заболеваний.

При прямом выводе необходимо последовательно задавать пользователю вопросы для определения наличия определенных симптомов (движение по дереву от корня к листьям). В конце необходимо вывести название заболевания и рекомендации к дальнейшим действиям.

В обратном выводе вначале необходимо вывести список заболеваний на выбор любого из них. После осуществления выбора заболевания необходимо проверить гипотезу о наличии или отсутствии данного заболевания. Проверка должна осуществляться путем задания вопросов на проверку симптомов (движение по дереву вверх от листьев к корню). В качестве результата работы алгоритма вывести «диагноз подтверждается» или «диагноз не подтверждается». Все вопросы задавать не нужно, если на некотором шаге возможно подтверждение или опровержение гипотезы о заболевании, то опрос должен быть завершен досрочно.

Ниже, в качестве примера приводится фрагмент базы заболеваний и примерный диагностический алгоритм постановки диагноза на основе болей живота ребенка (материалы взяты из медицинского справочника Т. Смита «Ваш семейный доктор. Домашний советчик»).

У вашего ребенка свыше 6 часов продолжаются боли в животе?

Есть ли у ребенка припухлости в паху или мошонке?

Есть ли у ребенка рвота?

Ребенок не дает вам даже слегка нажать на живот?

Есть ли боли при мочеиспускании?

Ребенок простужен или жалуется на боль в горле?

До боли ребенок выглядел здоровым?

У ребенка необычно жидкий или частый стул?

Это регулярные боли в течение нескольких месяцев?

Рвота зеленовато-желтого цвета?

нет

нет

нет

нет

нет

нет

да

да

нет

*аппендицит*

*ущемление паховой грыжи*

да

да

*гастроэмпирит*

да

нет

*кишечная непроходимость*

да

*аппендицит*

да

*воспаление мочевыводящих путей*

*инфекция верхних дыхательных путей*

*естественные беспричинные боли*

*эмоциональный стресс*

да

да

нет

да

нет

Для чтения данных о заболеваниях и записи симптомов необходимо разработать и использовать фреймовую модель представления знаний. В качестве фреймов и их экземпляров представить базу заболеваний, где для каждого заболевания должны храниться следующие данные: название заболевания, описание действий, которые нужно предпринять пользователю после его диагностики. Кроме этого, необходимо создать фрейм «Пациент», экземпляры которого должны содержать в себе ФИО пациента (вводится в начале работы системы), список симптомов (фиксируются в процессе диагностики) и сам диагноз. После окончания диагностики должна быть возможность вывода на экран всех данных из этого экземпляра (просмотр знаний о пациенте).

# 3 РАЗРАБОТКА ЭКСПЕРТНОЙ СИСТЕМЫ

## 3.1 Разработка правил системы

В экспертной системе реализованы правила диагностики, которые в зависимости от той или иной ситуации будут задавать пользователю необхо­димые вопросы, а после получать ответ в строго заданной форме. Дальнейшая ди­агностика будет производиться с учетом предыдущих ответов на вопросы, заданные пользователю. Эти ответы будут формировать описание текущей ситуации с помощью фактов.

Ниже, на рисунке 1, представлен код функции, которая выводит вопрос для пользователя на экран. Полный листинг программы представлен в приложении А.

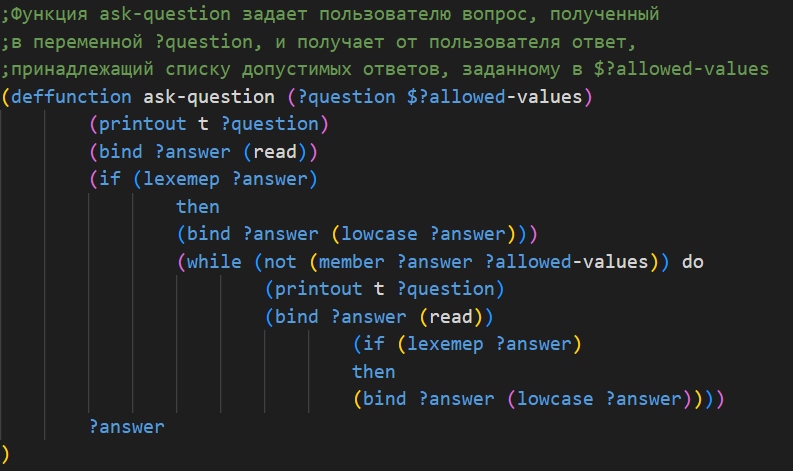


Рисунок 1 — Функция для вывода вопроса пользователю

Функция принимает два аргумента: простую переменную question, которая содержит текст вопроса, и составную переменную allowed-values с набором допустимых ответов. Сразу после своего вызова функция выводит на

экран соответствующий вопрос и читает ответ пользователя в переменную answer. Если переменная answer содержит текст, то она будет принудительно приве­дена к прописному алфавиту. После этого функция проверяет, является ли полученный ответ одним из заданных корректных ответов. Если нет, то процесс повторится до получения корректного ответа, иначе функция вер­нет ответ, введенный пользователем.

Функция yes-or-no-p вызывает функцию ask-question с постоянным набо­ром допустимых ответов: yes, no, у и n. В случае если пользователь ввел от­вет yes или у, функция возвращает значение *true*, иначе — *false*. На рисунке 2, представлен код данной функции.

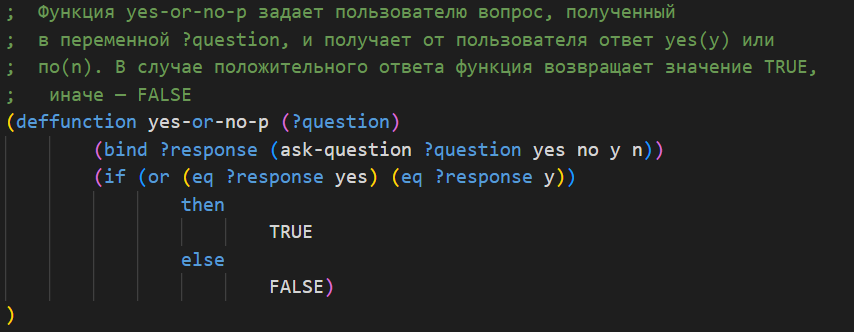


Рисунок 2 — Выводит вопрос пользователю и принимает ответ в строго заданной форме

Разработаем основные правила системы. На рисунке 3, представлено правило, определяющее температуру пациента.

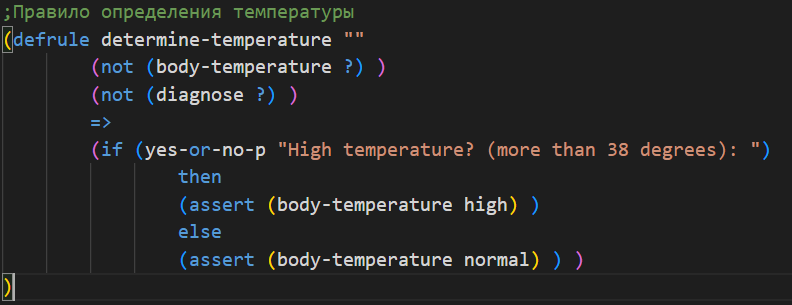


Рисунок 3 – Правило определяющее температуру пациента

## 3.2 Работоспособность системы

Для разработки системы использовался специальный программный инструмент CLIPS, который представлен на рисунке 4.

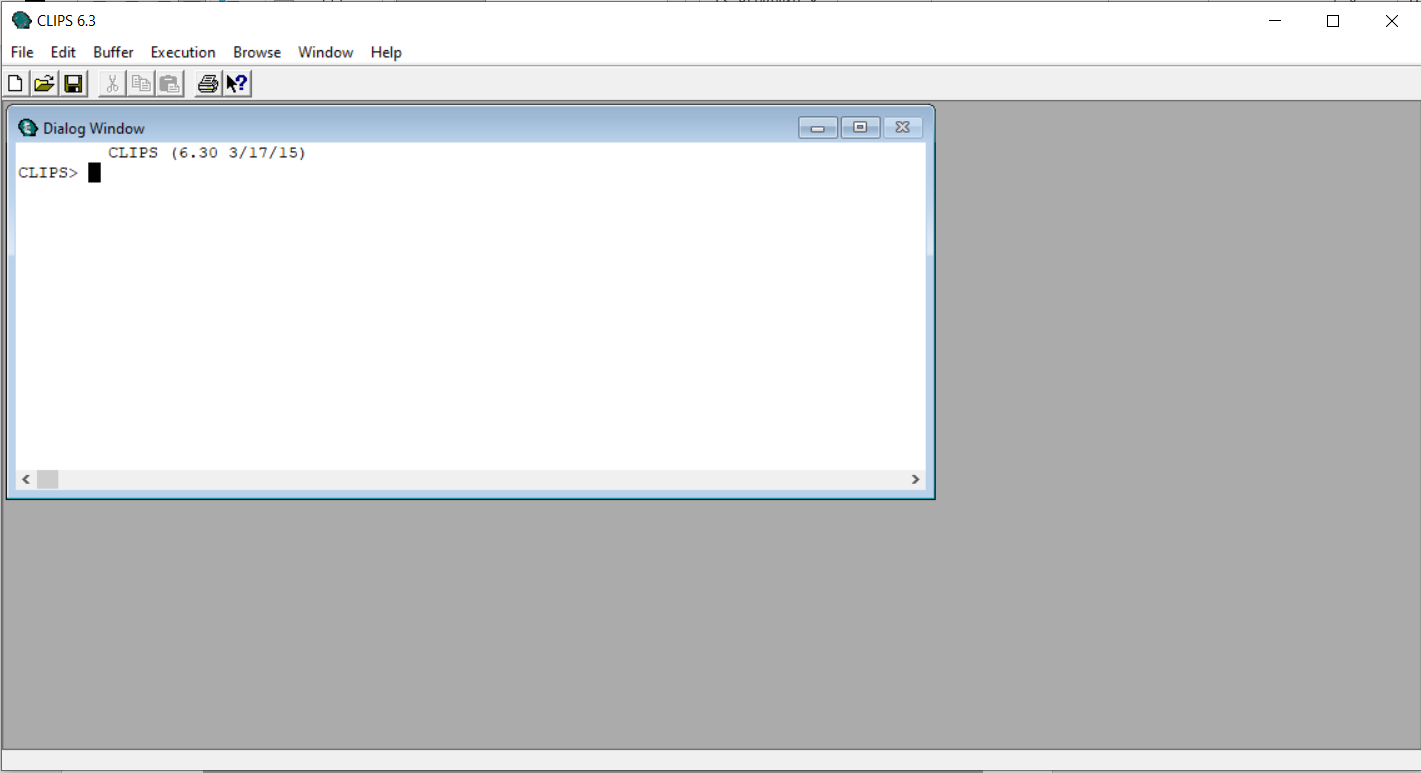


Рисунок 4 — Интерфейс CLIPS

Далее, необходимо загрузить скрипт экспертной системы, используя меню (“File” > “Load” затем выбрать нужный файл) (Рисунок 5).

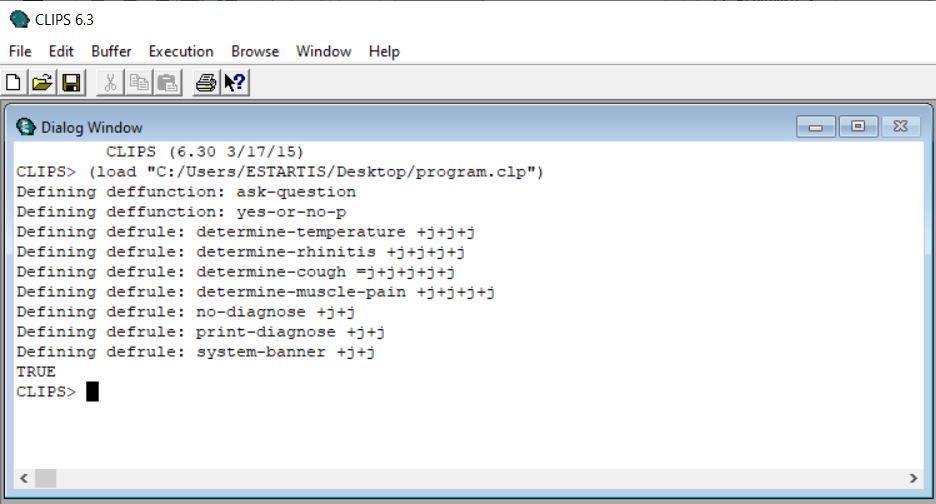


Рисунок 5 — Результат подключения файла

Далее необходимо ввести команду (reset), чтобы очистить список фактов, шаблонов и т.д. После нужно ввести команду (run), чтобы запустить систему (Рисунок 6-7).

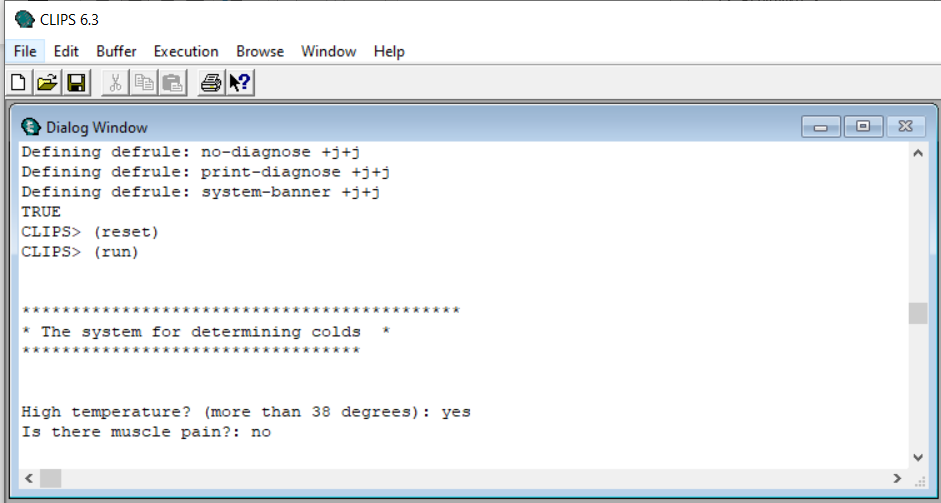


Рисунок 6 — Взятие информации у пациента о состоянии здоровья

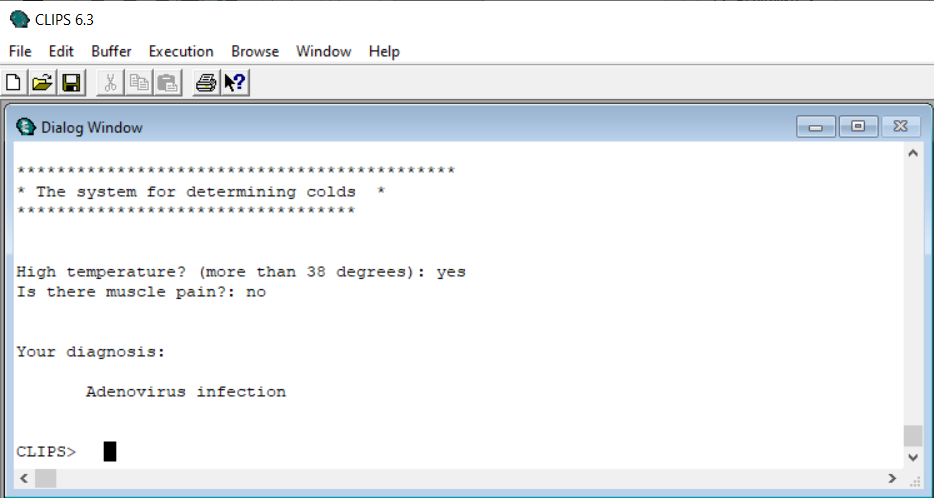


Рисунок 7 — Предложение решение по ремонту автомобиля

# 4. ВЫВОД

Во время выполнения лабораторной работы была разработана экспертная система по диагностированию простудных заболеваний. Изучен функционал языка разработки экспертных систем Clips, а также среда исполнения кода экспертных систем CLIPS (ver. 6.3).

Данная система обладает рядом недостатков, вызванных ограниченностью времени разработки. Реализован базовый функционал определения заболевания на основе поверхностных симптомов. В части улучшения системы можно выделить расширения списка доступных симптомов и заболеваний для более точного определения диагноза, также вспомогательные функции, автоматически определяющие зависимые симптомы, для избежания использования избыточных вопросов.

# ПРИЛОЖЕНИЕ А

**Листинг экспертной системы**

;Экспертная система диагностирования простудных заболеваний

;Вспомогательные функции

;============================================================================

;Функция ask-question задает пользователю вопрос, полученный

;в переменной ?question, и получает от пользователя ответ,

;принадлежащий списку допустимых ответов, заданному в $?allowed-values

(deffunction ask-question (?question $?allowed-values)

        (printout t ?question)

        (bind ?answer (read))

        (if (lexemep ?answer)

                then

                (bind ?answer (lowcase ?answer)))

                (while (not (member ?answer ?allowed-values)) do

                        (printout t ?question)

                        (bind ?answer (read))

                                (if (lexemep ?answer)

                                then

                                (bind ?answer (lowcase ?answer))))

        ?answer

)

;  Функция yes-or-no-p задает пользователю вопрос, полученный

;  в переменной ?question, и получает от пользователя ответ yes(у) или

;  по(n). В случае положительного ответа функция возвращает значение TRUE,

;   иначе — FALSE

(deffunction yes-or-no-p (?question)

        (bind ?response (ask-question ?question yes no у n))

        (if (or (eq ?response yes) (eq ?response y))

                then

                        TRUE

                else

                        FALSE)

)

;=====================================================================================

;Диагностические правила

;Правило определения температуры

(defrule determine-temperature ""

        (not (body-temperature ?) )

        (not (diagnose ?) )

        =>

        (if (yes-or-no-p "High temperature? (more than 38 degrees): ")

                then

                (assert (body-temperature high) )

                else

                (assert (body-temperature normal) ) )

)

;Правило определения ринита

(defrule determine-rhinitis ""

        (body-temperature normal)

        (not (rhinitis ?) )

        (not (diagnose ?) )

        =>

        (if (yes-or-no-p "Rhinitis is observed?: ")

                then

                (assert (rhinitis yes) )

                else

                (assert (diagnose "Parainfluenza")) )

)

;Правило определения кашля

(defrule determine-cough ""

        (body-temperature normal)

        (rhinitis yes)

        (not (cough ?) )

        (not (diagnose ?) )

        =>

        (if (yes-or-no-p "Is your cough dry?: ")

                then

                (assert (cough yes) )

                (assert (diagnose "Respiratory syncytial infection"))

                else

                (assert (cough no) )

                (assert (diagnose "Rhinovirus infection")) )

)

;Правило определения боли в мышцах

(defrule determine-muscle-pain ""

        (body-temperature high)

        (not (muscle-pain ?))

        (not (diagnose ?) )

        =>

        (if (yes-or-no-p "Is there muscle pain?: ")

                then

                (assert (muscle-pain yes) )

                (assert (diagnose "Flu"))

                else

                (assert (muscle-pain yes) )

                (assert (diagnose "Adenovirus infection")) )

)

;Правило no-diagnose запускается в случае, если ни одно

;из диагностических правил не способно определить диагноз.

;Правило корректно прерывает выполнение экспертной системы и предлагает

;обратитья к врачу

(defrule no-diagnose ""

        (declare (salience -10))

        (not (diagnose ?))

        =>

         (assert (diagnose "The diagnosis is not determined, consult a doctor"))

)

;Вывести диагноз

(defrule print-diagnose ""

        (declare (salience 10))

        (diagnose ?item)

        =>

        (printout       t       crlf crlf)

        (printout       t       "Your diagnosis:")

        (printout       t       crlf crlf)

        (format t       "       %s%n%n%n" ?item)

)

;- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

;Правило system-banner выводит на экран название экспертной системы

;при каждом новом запуске.

(defrule system-banner " каждом новом запуске."

        (declare (salience 10) )

         =>; каждом новом запуске.

        (printout t crlf crlf)

        (printout t "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" crlf)

        (printout t "\* The system for determining colds  \*" crlf)

        (printout t "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" crlf)

        (printout t crlf crlf)

)