МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное автономное бюджетное образовательное учреждение высшего образования

«**Севастопольский государственный университет**»

кафедра Информационных систем

Институт информационных технологий и управления в технических системах

Лисянский Александр Игоревич

курс 4 группа ИС/б-42-о

09.03.02 Информационные системы (уровень бакалавриата)

**ОТЧЁТ**

о лабораторном практикуме №3

РАЗРАБОТКА И ИССЛЕДОВАНИЕ ЭКСПЕРТНОЙ СИСТЕМЫ

по дисциплине «Методы и системы искусственного интеллекта»

Отметка о зачёте \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_

(дата)

Руководитель практикума

\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_Забаштанский А. К.\_

(должность) (подпись) (инициалы, фамилия)

Севастополь

2016

1. **ЦЕЛЬ РАБОТЫ**

Разработка экспертной системы продукционного типа на Прологе, исследование базовых принципов организации экспертных систем.

1. **ВАРИАНТ ЗАДАНИЯ**

22 Мобильные устройства. Реализовать продукционную экспертную систему в соответствии с номером варианта. При этом количество рассматриваемых объектов предметной области должно быть не менее 10 и характеризующих их атрибутов также — не менее 10. Система должна уметь давать объяснения вывода.

1. **ТЕКСТ ПРОГРАММЫ**

:-dynamic

сообщено/2.

определить\_операторы:-

op(950, xfx, то),

op(960, fx, если),

op(970, xfx, '::').

:-определить\_операторы.

%============обратный вывод======================================================

% реализуется предикатом найти(S,Стек,Д),где S - список проверяемых гипотез,

% Стек - стек из имен доказываемых гипотез и правил (используется при ответе на

% вопросы "почему), Д - дерево вывода целевого утверждения (используется при отве-

% те на вопросы "как"). Предикат получает на вход список [Н] и Стек=[H] и в про-

% цессе обратного вывода строит дерево вывода Д.

% Предикат "найти" для доказательства отдельных гипотез из списка S

% использует предикат найти1(Н,Стек,Дерево).

% случай1:если цель Н была подтверждена пользователем,

% то дерево вывода Д=сообщено(Н).

найти1(H,Стек,сообщено(H)):-сообщено(H,да).

найти1(H,Стек,сообщено(H)):-запрашиваемая(H),

not(сообщено(H,\_)),спроси(H,Стек).

% случай2:если цель Н подтверждается фактом, уже известным системе,

% то дерево вывода Д=Факт :: H

найти1(H,Стек,Факт :: H):-Факт :: H.

% случай3: если цель Н соответствует следствию одного из

% правил -> Правило :: если H1 то H

% и если Д1 дерево вывода для подцели Н1,

% то Д= Правило :: если Д1 то H и добавить № правила в Стек

найти1(H,Стек,Правило :: если Д1 то H):-

Правило :: если H1 то H,

найти(H1,[Правило | Стек],Д1).

% случай4: если доказывается конъюнкция гипотез, заданная списком гипотез,

% то найти доказательство первой гипотезы Н1 из списка

% с помощью найти1(H1,Стек,Дерево1), а затем найти доказательство оставшихся

% гипотез Т с помощью найти(T,Стек,Дерево) и

% объединить деревья вывода в общий список [Дерево1 | Дерево].

найти([],Стек,Дерево):-Дерево=[].

найти([H1|T],Стек,[Дерево1 | Дерево]):-

найти1(H1,Стек,Дерево1),найти(T,Стек,Дерево).

% проверка: является ли гипотеза признаком, значение которого можно спросить

запрашиваемая(H):-Факт :: признак(H).

%=========вывод вопросов и обработка ответов "да, нет, почему" ==================

%вывод вопроса и ввод ответа

спроси(H,Стек):-write(H),write('?'),nl,

read(O),ответ(H,O,Стек).

%обработка ответов: да, нет

ответ(H,да,Стек):-assert(сообщено(H,да)),!.

ответ(H,нет,Стек):-assert(сообщено(H,нет)),!,fail.

%обработка ответов - "почему"

% случай1: стек целей пустой

ответ(H,почему,[]):-!,write(' Вы задаете слишком много вопросов'),nl,

спроси(H,[]).

%случай2: в стеке осталась только первая введенная цель, т.е доказываемая гипотеза

ответ(P,почему,[H]):-!,write('моя гипотеза: '),

write(H),nl,спроси(P,[]).

%случай3: вывод заключения и номера правила для доказываемой текущей подцели Н

ответ(H,почему,[Правило | Стек]):-!,

Правило :: если H1 то H2,

write('пытаюсь доказать '),

write(H2),nl,

write('с помощью правила: '),

write(Правило),nl,

спроси(H,Стек).

%неправильный ответ: повторяем вопрос

ответ(H,\_,Стек):-write(' правильный ответ: да, нет, почему'),nl,

спроси(H,Стек).

как(H,Дерево):-как1(H,Дерево),!.

% вывод сообщения, если Н не найдено

как(H,\_):-write(H),tab(2),write('не доказано'),nl.

% случай1: если Н сообщено пользователем,

% то вывести "Н было введено"

как1(H,\_):-сообщено(H,\_),!,

write(H),write('было введено'),nl.

% случай2: если дерево вывода Д представлено фактом, подтверждающим Н

как1(H,Факт :: H):-!,

write(H), write( 'является фактом'), write(Факт),nl.

% случай3: если дерево вывода Д - правило в заключение, которого есть Н,

% то отобразить это правило

как1(H,[Правило :: если \_ то H]):-!,

write(H),write(' было доказано с помощью'),nl,

Правило :: если H1 то H,

отобрази\_правило(Правило :: если H1 то H).

% случай4: если в дереве Д нет правила с заключением Н,

%то поиск Н надо выполнять в дереве вывода предпосылок, т.е. в Дерево

как1(H,[Правило :: если Дерево то \_]):-как(H,Дерево).

как1(H,[]):-!.

как1(H,[Д1|Д2]):-как(H,[Д1]),!;

как1(H,Д2).

%вывод правила на экран

отобрази\_правило(Правило :: если H1 то H):-

write(Правило), write( ':'),nl,

write('если '), write(H1), nl,

write('то '), write(H),nl.

/\* Вызов интерпретатора\*/

инициализация:-retractall(сообщено(\_,\_)).

start:-

/\* Загрузка базы знаний из файла\*/

reconsult('rules.pl'),

info, %отображение информации о базе знаний\*

go\_exp\_sys.

go\_exp\_sys:- инициализация,

Факт :: гипотеза(H),

найти([H],[H],Дерево),

write('решение:'),write(H),nl,

объясни(Дерево),

возврат.

%объяснение вывода утверждения

объясни(Дерево):-write( 'объяснить ? [цель/нет]:'), nl,read(H),

(H\=нет,!,как(H,Дерево),объясни(Дерево));!.

%поиск следующих решений

возврат:-write('Искать ещё решение [да/нет] ?: '),nl, read(нет).

:-write('Искать ещё решение [да/нет] ?: '),nl, read(нет).

info:-

nl,

write('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*'),nl,

write('\* Экспертная система \*'),nl,

write('\* \*'),nl,

write('\* \*'),nl,

write('\*---------------------------- \*'),nl,

write('\* Отвечайте на вопросы: \*'),nl,

write('\* да, нет, почему \*'),nl,

write('\* Для объяснения решения \*'),nl,

write('\* введите цель \*'),nl,

write('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*'), nl ,

write('Введите любой символ'),nl, %Ожидание ввода литеры

get0(\_).

правило1 :: если [раздел(мобильные\_устройства)]

то мобильные\_устройства.

правило2 :: если [мобильные\_устройства, количество\_диапазонов]

то 4.

правило3 :: если [мобильные\_устройства, частота\_диапозонов]

то "800, 900, 1800, 1900 МГ".

правило4 :: если [мобильные\_устройства, зарядный\_вход]

то micro\_usb.

правило5 :: если [мобильные\_устройства, операционная\_система]

то android.

правило6 :: если [раздел(мобильные\_технологии)]

то мобильные\_технологии.

правило7 :: если [мобильные\_технологии, что\_такое\_reflex]

то гибкий\_смартифон.

правило8 :: если [мобильные\_технологии, сканер\_отпечатка\_пальца]

то qualcomm.

правило9 :: если [мобильные\_технологии, воспроизведение\_видео]

то xevic.

правило10 :: если [мобильные\_технологии, фильтр\_камеры]

то рыбий\_глаз.

% гипотезы

h1 :: гипотеза(4).

h2 :: гипотеза("800, 900, 1800, 1900 МГ").

h3 :: гипотеза(micro\_usb).

h4 :: гипотеза(android).

h5 :: гипотеза(гибкий\_смартифон).

h6 :: гипотеза(qualcomm).

h7 :: гипотеза(xevic).

h8 :: гипотеза(рыбий\_глаз).

% признаки математики, истинность которых можно выяснить у пользователя

q1 :: признак(раздел(мобильные устройства)).

q2 :: признак(количество\_диапазонов).

q3 :: признак(частота\_диапозонов).

q4 :: признак(зарядный\_вход).

q5 :: признак(операционная\_система).

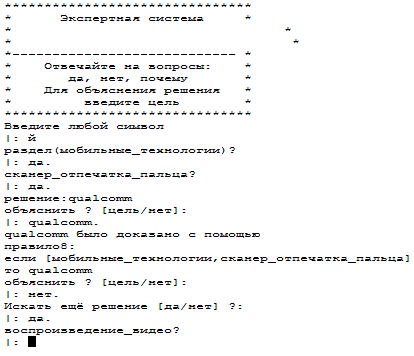
q6 :: признак(раздел(мобильные\_технологии)).

q7 :: признак(reflex).

q8 :: признак(сканер\_отпечатка\_пальца).

q9 :: признак(воспроизведение\_видео).

q10 :: признак(фильтр\_камеры).

**4. РЕЗУЛЬТАТЫ РАБОТЫ ПРОГРАММЫ**  


**ВЫВОД**

При выполнении лабораторной работы были изучены особенности разработки экспертной системы продукционного типа и исследованы основные базовые принципы организации экспертных систем.