**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра математического обеспечения и применения ЭВМ**

отчет

**по лабораторной работе №5**

**по дисциплине «Базы знаний и экспертные системы»**

**Тема: Разработка простой экспертной системы на языке Visual Prolog**

|  |  |  |
| --- | --- | --- |
| Студент гр. 6381 |  | Шевелева А.М. |
| Преподаватель |  | Сучков А.И. |

Санкт-Петербург

2019

Цель работы.

Реализация и отладка простой экспертной системы на языке Visual Prolog.

Основные теоретические положения

Экспертные системы (ЭС) – это системы искусственного интеллекта (интеллектуальные системы), предназначенные для решения плохо формализованных и слабо структурированных задач в определенных проблемных областях на основе заложенных в них знаний специалистов-экспертов. В настоящее время ЭС внедряются в различные виды человеческой деятельности, где использование точных математических методов и моделей затруднительно или вообще невозможно. К ним относятся: медицина, обучение, поддержка принятия решений и управление в сложных ситуациях, различные деловые приложения и т.д.

Основными компонентами ЭС являются рабочая память, называемая также базой данных (БД), база знаний (БЗ), блоки поиска решения, объяснения, извлечения и накопления знаний, обучения и организации взаимодействия с пользователем. Рабочая память, БЗ и блок поиска решений образуют ядро ЭС.

База данных (рабочая память) предназначена для хранения исходных и промежуточных данных решаемой в текущий момент задачи. Этот термин совпадает по названию, но не по смыслу с термином, используемым в информационно-поисковых системах (ИПС) и системах управления базами данных (СУБД) для обозначения всех данных (в первую очередь долгосрочных), хранимых в системе.

База знаний в ЭС предназначена для хранения долгосрочных данных, описывающих рассматриваемую предметную область (а не текущих данных), и правил, описывающих целесообразные преобразования данных этой области.

Для конструирования ЭС используются различные инструментальные средства: универсальные языки программирования, языки искусственного интеллекта, инструментальные системы и среды и системы-оболочки. Системы-оболочки являются наиболее простым средством формализации (автоформализации) экспертных знаний, практически не требующие участия посредников в лице инженера по знаниям или программиста при их использовании. Инженер по знаниям только помогает эксперту выбрать наиболее подходящую для его проблемной области оболочку.

Рассматриваемая в данной работе экспертная система функционирует следующим образом. Например, при работе с предметной областью «Бытовая техника», определяется предикат technique\_is(string), позволяющий идентифицировать бытовую технику, которая более всего удовлетворяет указанным пользователем признакам. При этом БЗ экспертной системы задается перечислением свойств объектов, которые у них имеются или же отсутствуют (см. рис. 1).

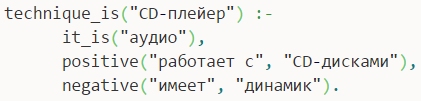


Рисунок 1 − Объект БЗ экспертной системы

Предикаты positive и negative необходимы для того, чтобы запросить у пользователя ответ о наличии у объекта какого-то свойства и сохранить этот ответ в рабочей памяти. Два встроенных предиката asserta и assertz позволяют добавлять новые утверждения в базу данных. Оба предиката действуют в точности одинаковым образом, за тем исключением, что asserta добавляет утверждение в начало базы данных, в то время как assertz (или assert) добавляет утверждение в ее конец.

Постановка задачи.

Разработать в среде Visual Prolog простую экспертную систему (ЭС). Порядок выполнения работы следующий:

1. Разработать простую ЭС для определения породы кошек.
2. Произвести отладку ЭС в среде Visual Prolog для различных наборов ответов пользователя на задаваемые экспертной системой вопросы.

Выполнение работы.

Разработана простая экспертная система с прямой цепочкой рассуждений для определения породы кошек. Количество возможных видов пород − 14, количество признаков пород − 7, максимальная глубина дерева определения породы по существующим признакам − 4. Схема ЭС представлена на рис. 2.

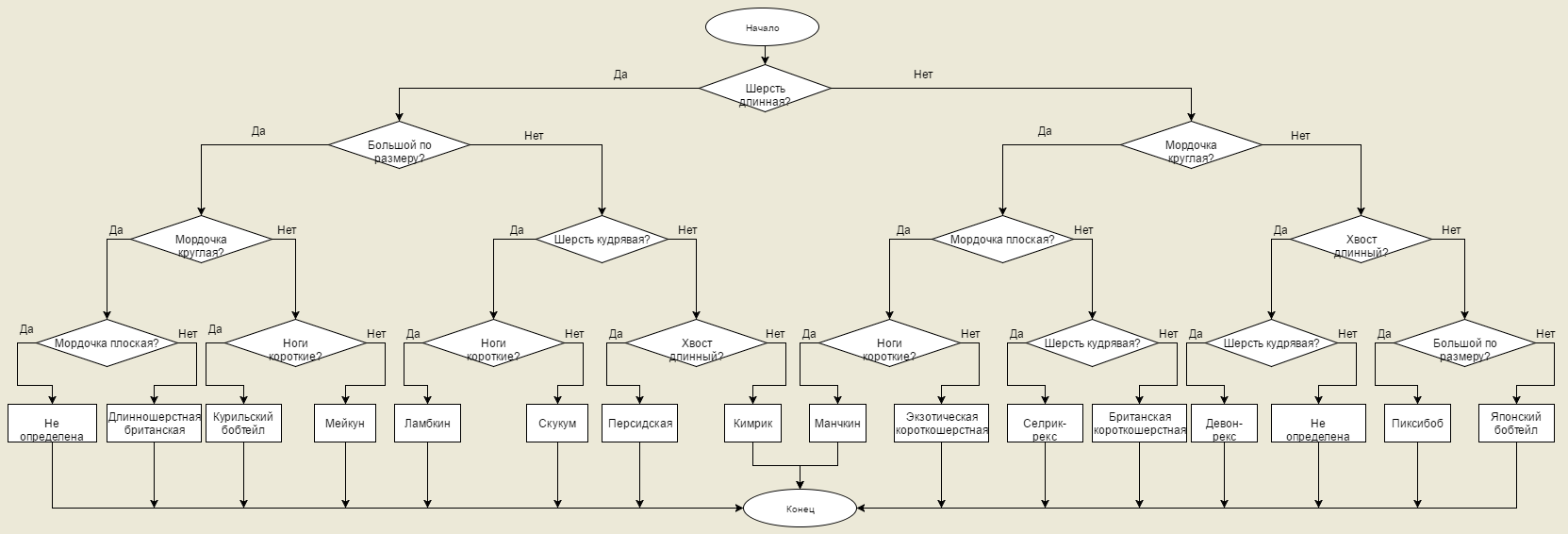


Рисунок 2 − Схема ЭС определения породы кошек

Графическое приложение с ЭС было реализовано на языке Visual Prolog. Полный код программы представлен в приложении А.

Пользователь программы последовательно отвечает на 4 вопроса (да/нет), отмечая галочкой, если ответ верный, и нажимает кнопку «Далее». После того, как нажата кнопка «Далее», она меняет свое содержание на да/нет, в зависимости от ответа пользователя. А также появляется следующий вопрос в зависимости от того, какой ответ дал пользователь. Пока не нажата кнопка «Далее» все последующие кнопки не активны. Пользователь может поменять свой ответ в уже активных кнопках, но тогда все последующие ответы будут очищены. После ответа на 4 вопрос, программа выдает породу кошки.

Также пользователь может нажать кнопку «Очистить экран», чтобы начать ввод заново. Начальное представление приложения показано на рис. 3.

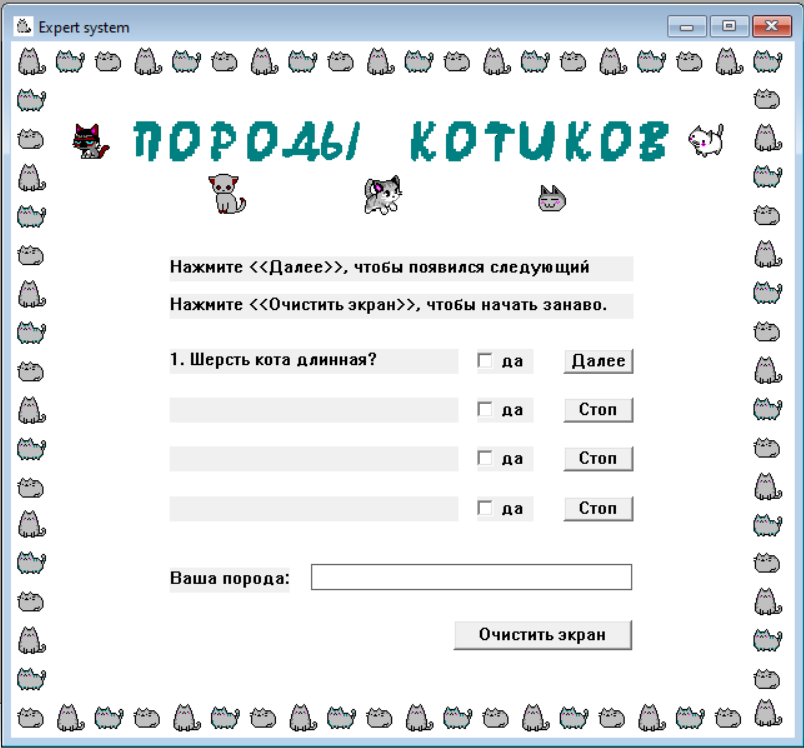


Рисунок 3 − Начальное окно приложения

Программа отлажена на следующих наборах ответов пользователя (см. рис. 4-7):

Определение породы «британская длинношерстная».

* Шерсть кота длинная (да).
* Размер кота большой (да).
* Мордочка кота круглая (да).
* Мордочка кота плоская (нет).

Определение породы «персидская».

* Шерсть кота длинная (да).
* Размер кота большой (нет).
* Шерсть кота кудрявая (нет).
* Хвост кота длинный (обычный) (да).

Определение породы «девон-рекс».

* Шерсть кота длинная (нет).
* Мордочка кота круглая (нет).
* Хвост кота длинный (обычный) (да).
* Шерсть кота кудрявая (да).

Определение породы «пиксибоб».

* Шерсть кота длинная (нет).
* Мордочка кота круглая (нет).
* Хвост кота длинный (обычный) (нет).
* Размер кота большой (да).

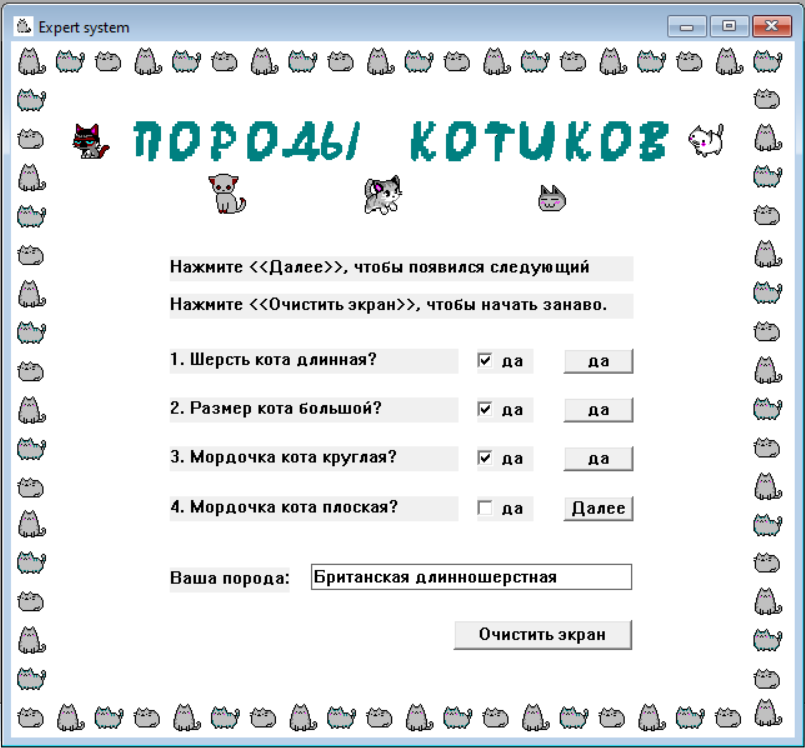


Рисунок 4 − Определение породы «Британская длинношерстная»

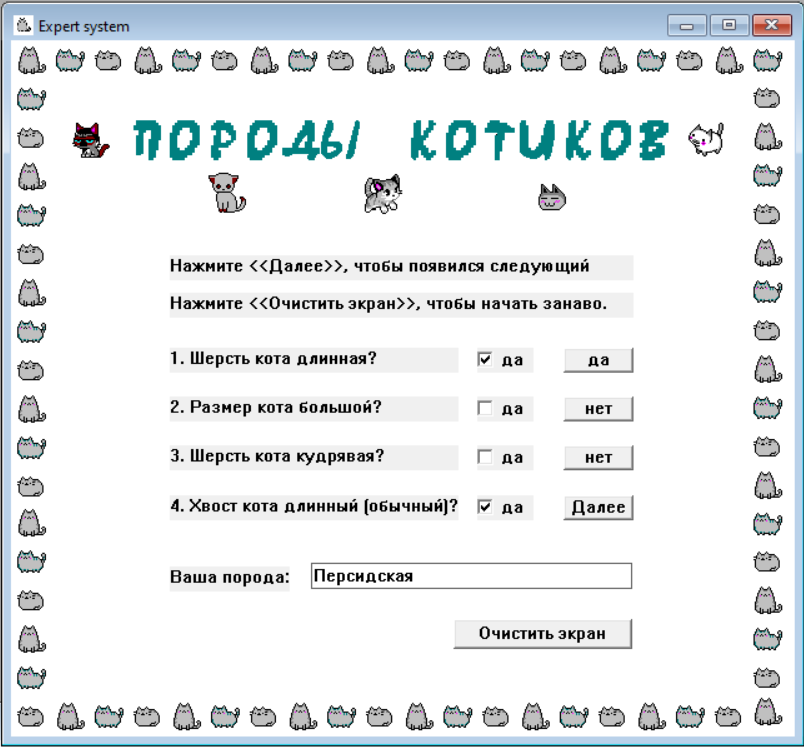


Рисунок 5 − Определение породы «Персидская»

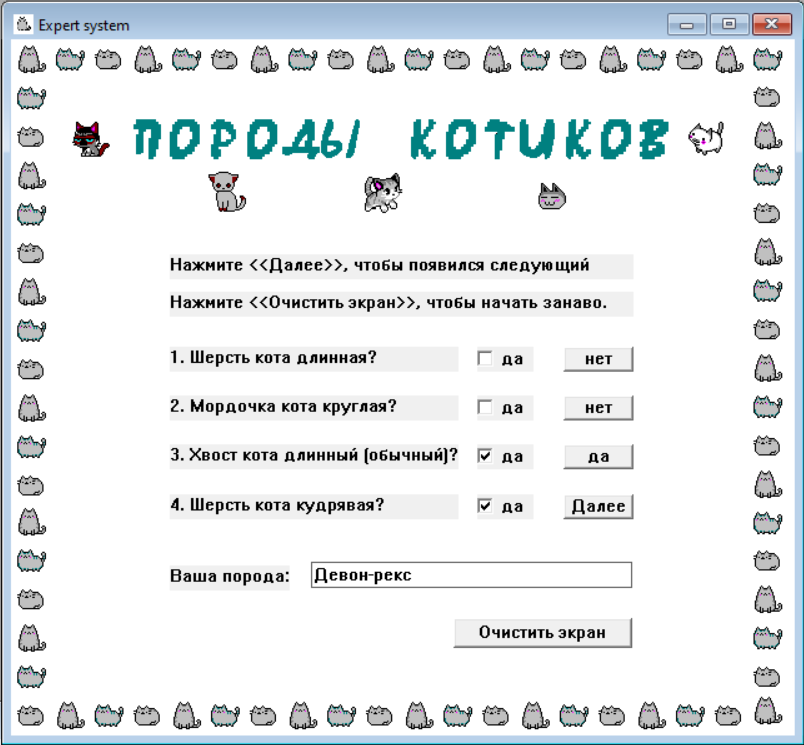


Рисунок 6 − Определение породы «Девон-рекс»

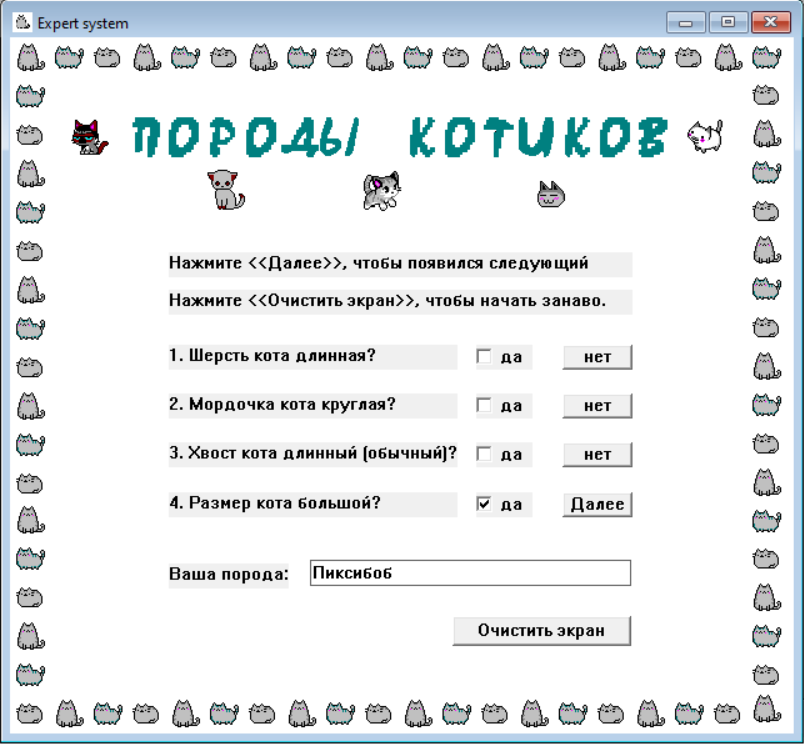


Рисунок 7 − Определение породы «Пиксибоб»

Выводы.

В программе Visual Prolog было создано графическое приложение на основе экспертной системы с прямой цепочкой рассуждений для нахождения породы кошек. Программа определяет 14 различных пород кошек за 4 вопроса (глубина дерева определения − 4). Всего различных вопросов в программе − 7.

Приложение динамически задает вопросы пользователю в зависимости от его ответа на предыдущие вопросы. После ответа на 4 вопрос пользователь получает результат. В случае, если программа не может определить породу, то она выдает ответ − «Порода не определена».

Пользователь может очистить ввод и начать заново, но программа не может очистить поля ввода галочек.

Приложение было протестировано на разных запросах и работает в соответствии со схемой ЭС.

**ПРИЛОЖЕНИЕ А**

**КОД ПРОГРАММЫ, ЭКСПЕРТНОЙ СИСТЕМЫ ДЛЯ ОПРЕДЕЛЕНИЯ ПОРОДЫ КОШЕК**

DATABASE - breed

xpositive(string,string)

xnegative(string,string)

found(string)

predicates

win\_expert\_system\_eh : EHANDLER

if\_check(boolean,string)

remember(string,string,string)

positive(string,string)

find\_breed

run(string)

test1(string)

test2(string,string)

test3(string,string,string)

test4(string,string,string,string)

t2(string,string)

t3(string,string,string)

t4(string,string,string,string)

button2(string,string,string,string,string,string,string,string)

button3(string,string, string, string,string,string,string)

button4(string,string, string, string,string,string)

clear\_facts

clauses

if\_check(X,Y):- X=1, Y="да".

if\_check(X,Y):- X<>1, Y="нет".

clear\_facts :- retractall(xpositive(\_,\_)).

clear\_facts :- retractall(xnegative(\_,\_)).

positive(X,Y) :- xpositive(X,Y),!.

positive(X,Y) :- not(xnegative(X,Y)).

remember(X,Y,"да") :- asserta(xpositive(X,Y)).

remember(X,Y,"нет") :- asserta(xnegative(X,Y)).

run(Y) :- find\_breed, found(X), Y=X, !.

find\_breed :- test1(X),test2(X,Y),test3(X,Y,Z),

test4(X,Y,Z,S),!.

find\_breed.

test1("да") :- positive("Шерсть кота","длинная"),!.

test1("нет").

test2("да","да") :- positive("Размер кота","большой"),!.

test2("да","нет").

test2("нет","да") :- positive("Мордочка кота","круглая"),!.

test2("нет","нет").

test3("да","да","да") :- positive("Мордочка кота","круглая"),!.

test3("да","да","нет").

test3("да","нет","да") :- positive("Шерсть кота", "кудрявая"),!.

test3("да","нет","нет").

test3("нет","да","да") :- positive("Мордочка кота","плоская"),!.

test3("нет","да","нет").

test3("нет","нет","да"):- positive("Хвост кота","длинный"),!.

test3("нет","нет","нет").

test4("да","да","да","да") :- positive("Мордочка кота","плоская"),

asserta(found("Не определена")),!.

test4("да","да","да","нет") :- asserta(found("Британская длинношерстная")),!.

test4("да","да","нет","да") :- positive("Ноги кота", "короткие"),

asserta(found("Курильский бобтейл")),!.

test4("да","да","нет","нет") :- asserta(found("Мейкун")),!.

test4("да","нет","да","да"):- positive("Ноги кота", "короткие"),

asserta(found("Ламбкин")),!.

test4("да","нет","да","нет") :- asserta(found("Скукум")),!.

test4("да","нет","нет","да"):- positive("Хвост кота","длинный"),

asserta(found("Персидская")),!.

test4("да","нет","нет","нет") :- asserta(found("Кимрик")),!.

test4("нет","да","да","да"):- positive("Ноги кота", "короткие"),

asserta(found("Манчкин")),!.

test4("нет","да","да","нет") :- asserta(found("Экзотическая короткошерстная")),!.

test4("нет","да","нет","да"):- positive("Шерсть кота", "кудрявая"),

asserta(found("Селрик-рекс")),!.

test4("нет","да","нет","нет") :- asserta(found("Британская короткошерстная")),!.

test4("нет","нет","да","да"):- positive("Шерсть кота", "кудрявая"),

asserta(found("Девон-рекс")),!.

test4("нет","нет","да","нет") :- asserta(found("Не определена")),!.

test4("нет","нет","нет","да"):- positive("Размер кота", "большой"),

asserta(found("Пиксибоб")),!.

test4("нет","нет","нет","нет"):- asserta(found("Японский бобтейл")),!.

t2("да",Y):- Y="2. Размер кота большой?",!.

t2("нет",Y):- Y="2. Мордочка кота круглая?",!.

t3("да","да",Y):- Y="3. Мордочка кота круглая?",!.

t3("да","нет",Y):- Y="3. Шерсть кота кудрявая?",!.

t3("нет","да",Y):- Y="3. Мордочка кота плоская?",!.

t3("нет","нет",Y):- Y="3. Хвост кота длинный (обычный)?",!.

t4("да","да","да",Y):- Y="4. Мордочка кота плоская?",!.

t4("да","нет","да",Y):- Y="4. Ноги кота короткие?",!.

t4("нет","да","да",Y):- Y="4. Ноги кота короткие?",!.

t4("нет","нет","да",Y):- Y="4. Шерсть кота кудрявая?",!.

t4("да","да","нет",Y):- Y="4. Ноги кота короткие?",!.

t4("да","нет","нет",Y):- Y="4. Хвост кота длинный (обычный)?",!.

t4("нет","да","нет",Y):- Y="4. Шерсть кота кудрявая?",!.

t4("нет","нет","нет",Y):- Y="4. Размер кота большой?",!.

button2("Стоп",\_,\_,X1,X2,Y1,Y2,Y3):-X1="",X2="",

Y1="Стоп",Y2="Стоп",Y3="Стоп",!.

button2(\_,"да","да",X1,X2,Y1,Y2,Y3):-t3("да","да",Y),X1=Y,X2="",

Y1="да",Y2="Далее",Y3="Стоп",!.

button2(\_,"нет","да",X1,X2,Y1,Y2,Y3):-t3("да","нет",Y),X1=Y,X2="",

Y1="нет",Y2="Далее",Y3="Стоп",!.

button2(\_,"да","нет",X1,X2,Y1,Y2,Y3):-t3("нет","да",Y),X1=Y,X2="",

Y1="да",Y2="Далее",Y3="Стоп",!.

button2(\_,"нет","нет",X1,X2,Y1,Y2,Y3):-t3("нет","нет",Y),X1=Y,X2="",

Y1="нет",Y2="Далее",Y3="Стоп",!.

button3("Стоп",\_,\_,\_,X1,Y1,Y2):-X1="",Y1="Стоп",Y2="Стоп",!.

button3(\_,"да","да","да",X1,Y1,Y2):-t4("да","да","да",Y),

X1=Y,Y1="да",Y2="Далее",!.

button3(\_,"да","да","нет",X1,Y1,Y2):-t4("нет","да","да",Y),

X1=Y,Y1="да",Y2="Далее",!.

button3(\_,"да","нет","да",X1,Y1,Y2):-t4("да","нет","да",Y),

X1=Y,Y1="да",Y2="Далее",!.

button3(\_,"да","нет","нет",X1,Y1,Y2):-t4("нет","нет","да",Y),

X1=Y,Y1="да",Y2="Далее",!.

button3(\_,"нет","да","да",X1,Y1,Y2):-t4("да","да","нет",Y),

X1=Y,Y1="нет",Y2="Далее",!.

button3(\_,"нет","да","нет",X1,Y1,Y2):-t4("нет","да","нет",Y),

X1=Y,Y1="нет",Y2="Далее",!.

button3(\_,"нет","нет","да",X1,Y1,Y2):-t4("да","нет","нет",Y),

X1=Y,Y1="нет",Y2="Далее",!.

button3(\_,"нет","нет","нет",X1,Y1,Y2):-t4("нет","нет","нет",Y),

X1=Y,Y1="нет",Y2="Далее",!.

button4("Стоп",\_,\_,\_,\_,X):-X="",!.

button4(\_,"да","да","да","да",X):-clear\_facts,

remember("Шерсть кота","длинная","да"),

remember("Размер кота","большой","да"),

remember("Мордочка кота","круглая","да"),

remember("Мордочка кота","плоская","да"),

run(Y),X=Y,!.

button4(\_,"нет","да","да","да",X):-clear\_facts,

remember("Шерсть кота","длинная","да"),

remember("Размер кота","большой","да"),

remember("Мордочка кота","круглая","да"),

remember("Мордочка кота","плоская","нет"),

run(Y),X=Y,!.

button4(\_,"да","нет","да","да",X):-clear\_facts,

remember("Шерсть кота","длинная","да"),

remember("Размер кота","большой","да"),

remember("Мордочка кота","круглая","нет"),

remember("Ноги кота", "короткие","да"),

run(Y),X=Y,!.

button4(\_,"нет","нет","да","да",X):-clear\_facts,

remember("Шерсть кота","длинная","да"),

remember("Размер кота","большой","да"),

remember("Мордочка кота","круглая","нет"),

remember("Ноги кота", "короткие","нет"),

run(Y),X=Y,!.

button4(\_,"да","да","нет","да",X):-clear\_facts,

remember("Шерсть кота","длинная","да"),

remember("Размер кота","большой","нет"),

remember("Шерсть кота", "кудрявая","да"),

remember("Ноги кота", "короткие","да"),

run(Y),X=Y,!.

button4(\_,"нет","да","нет","да",X):-clear\_facts,

remember("Шерсть кота","длинная","да"),

remember("Размер кота","большой","нет"),

remember("Шерсть кота", "кудрявая","да"),

remember("Ноги кота", "короткие","нет"),

run(Y),X=Y,!.

button4(\_,"да","нет","нет","да",X):-clear\_facts,

remember("Шерсть кота","длинная","да"),

remember("Размер кота","большой","нет"),

remember("Шерсть кота", "кудрявая","нет"),

remember("Хвост кота","длинный","да"),

run(Y),X=Y,!.

button4(\_,"нет","нет","нет","да",X):-clear\_facts,

remember("Шерсть кота","длинная","да"),

remember("Размер кота","большой","нет"),

remember("Шерсть кота", "кудрявая","нет"),

remember("Хвост кота","длинный","нет"),

run(Y),X=Y,!.

button4(\_,"да","да","да","нет",X):-clear\_facts,

remember("Шерсть кота","длинная","нет"),

remember("Мордочка кота","круглая","да"),

remember("Мордочка кота","плоская","да"),

remember("Ноги кота", "короткие","да"),

run(Y),X=Y,!.

button4(\_,"нет","да","да","нет",X):-clear\_facts,

remember("Шерсть кота","длинная","нет"),

remember("Мордочка кота","круглая","да"),

remember("Мордочка кота","плоская","да"),

remember("Ноги кота", "короткие","нет"),

run(Y),X=Y,!.

button4(\_,"да","нет","да","нет",X):-clear\_facts,

remember("Шерсть кота","длинная","нет"),

remember("Мордочка кота","круглая","да"),

remember("Мордочка кота","плоская","нет"),

remember("Шерсть кота", "кудрявая","да"),

run(Y),X=Y,!.

button4(\_,"нет","нет","да","нет",X):-clear\_facts,

remember("Шерсть кота","длинная","нет"),

remember("Мордочка кота","круглая","да"),

remember("Мордочка кота","плоская","нет"),

remember("Шерсть кота", "кудрявая","нет"),

run(Y),X=Y,!.

button4(\_,"да","да","нет","нет",X):-clear\_facts,

remember("Шерсть кота","длинная","нет"),

remember("Мордочка кота","круглая","нет"),

remember("Хвост кота","длинный","да"),

remember("Шерсть кота", "кудрявая","да"),

run(Y),X=Y,!.

button4(\_,"нет","да","нет","нет",X):-clear\_facts,

remember("Шерсть кота","длинная","нет"),

remember("Мордочка кота","круглая","нет"),

remember("Хвост кота","длинный","да"),

remember("Шерсть кота", "кудрявая","нет"),

run(Y),X=Y,!.

button4(\_,"да","нет","нет","нет",X):-clear\_facts,

remember("Шерсть кота","длинная","нет"),

remember("Мордочка кота","круглая","нет"),

remember("Хвост кота","длинный","нет"),

remember("Размер кота","большой","да"),

run(Y),X=Y,!.

button4(\_,"нет","нет","нет","нет",X):-clear\_facts,

remember("Шерсть кота","длинная","нет"),

remember("Мордочка кота","круглая","нет"),

remember("Хвост кота","длинный","нет"),

remember("Размер кота","большой","нет"),

run(Y),X=Y,!.

win\_expert\_system\_eh(\_Win,e\_Control(idc\_push\_button,\_CtrlType,\_CtrlWin,\_CtlInfo),0):-

!,

Ctl1=win\_GetCtlHandle(\_Win,idc\_push\_button1),

Ctl2=win\_GetCtlHandle(\_Win,idc\_push\_button2),

Ctl3=win\_GetCtlHandle(\_Win,idc\_push\_button3),

Ctl4=win\_GetCtlHandle(\_Win,idc\_push\_button4),

Idct2=win\_GetCtlHandle(\_Win,idct\_2),

Idct3=win\_GetCtlHandle(\_Win,idct\_3),

Idct4=win\_GetCtlHandle(\_Win,idct\_4),

Ctl=win\_GetCtlHandle(\_Win,idc\_expert),

win\_SetText(Ctl,""),

win\_SetText(Ctl1,"Далее"),

win\_SetText(Ctl2,"Стоп"),

win\_SetText(Ctl3,"Стоп"),

win\_SetText(Ctl4,"Стоп"),

win\_SetText(Idct2,""),

win\_SetText(Idct3,""),

win\_SetText(Idct4,""),

!.

win\_expert\_system\_eh(\_Win,e\_Control(idc\_push\_button1,\_CtrlType,\_CtrlWin,\_CtlInfo),0):-

!,

Ctl1=win\_GetCtlHandle(\_Win,idc\_push\_button1),

Ctl2=win\_GetCtlHandle(\_Win,idc\_push\_button2),

Ctl3=win\_GetCtlHandle(\_Win,idc\_push\_button3),

Ctl4=win\_GetCtlHandle(\_Win,idc\_push\_button4),

Idct2=win\_GetCtlHandle(\_Win,idct\_2),

Idct3=win\_GetCtlHandle(\_Win,idct\_3),

Idct4=win\_GetCtlHandle(\_Win,idct\_4),

Check=win\_GetCtlHandle(\_Win,idc\_check\_box),

IsCheck= win\_IsChecked(Check),

if\_check(IsCheck,String),

t2(String,String2),

win\_SetText(Ctl1,String),

win\_SetText(Ctl2,"Далее"),

win\_SetText(Ctl3,"Стоп"),

win\_SetText(Ctl4,"Стоп"),

win\_SetText(Idct2,String2),

win\_SetText(Idct3,""),

win\_SetText(Idct4,""),

Ctl=win\_GetCtlHandle(\_Win,idc\_expert),

win\_SetText(Ctl,""),

!.

win\_expert\_system\_eh(\_Win,e\_Control(idc\_push\_button2,\_CtrlType,\_CtrlWin,\_CtlInfo),0):-

!,

Ctl1=win\_GetCtlHandle(\_Win,idc\_push\_button1),

Ctl2=win\_GetCtlHandle(\_Win,idc\_push\_button2),

Ctl3=win\_GetCtlHandle(\_Win,idc\_push\_button3),

Ctl4=win\_GetCtlHandle(\_Win,idc\_push\_button4),

Idct3=win\_GetCtlHandle(\_Win,idct\_3),

Idct4=win\_GetCtlHandle(\_Win,idct\_4),

Check=win\_GetCtlHandle(\_Win,idc\_check\_box1),

IsCheck= win\_IsChecked(Check),

if\_check(IsCheck,String),

StringB1 = win\_GetText(Ctl1),

StringB2 = win\_GetText(Ctl2),

button2(StringB2,String,StringB1,X1,X2,Y1,Y2,Y3),

win\_SetText(Ctl2,Y1),

win\_SetText(Ctl3,Y2),

win\_SetText(Ctl4,Y3),

win\_SetText(Idct3,X1),

win\_SetText(Idct4,X2),

Ctl=win\_GetCtlHandle(\_Win,idc\_expert),

win\_SetText(Ctl,""),

!.

win\_expert\_system\_eh(\_Win,e\_Control(idc\_push\_button3,\_CtrlType,\_CtrlWin,\_CtlInfo),0):-

!,

Ctl1=win\_GetCtlHandle(\_Win,idc\_push\_button1),

Ctl2=win\_GetCtlHandle(\_Win,idc\_push\_button2),

Ctl3=win\_GetCtlHandle(\_Win,idc\_push\_button3),

Ctl4=win\_GetCtlHandle(\_Win,idc\_push\_button4),

Idct4=win\_GetCtlHandle(\_Win,idct\_4),

Check=win\_GetCtlHandle(\_Win,idc\_check\_box2),

IsCheck= win\_IsChecked(Check),

if\_check(IsCheck,String),

StringB1 = win\_GetText(Ctl1),

StringB2 = win\_GetText(Ctl2),

StringB3 = win\_GetText(Ctl3),

button3(StringB3,String,StringB2,StringB1,X1,Y1,Y2),

win\_SetText(Ctl3,Y1),

win\_SetText(Ctl4,Y2),

win\_SetText(Idct4,X1),

Ctl=win\_GetCtlHandle(\_Win,idc\_expert),

win\_SetText(Ctl,""),

!.

win\_expert\_system\_eh(\_Win,e\_Control(idc\_push\_button4,\_CtrlType,\_CtrlWin,\_CtlInfo),0):-

!,

Ctl1=win\_GetCtlHandle(\_Win,idc\_push\_button1),

Ctl2=win\_GetCtlHandle(\_Win,idc\_push\_button2),

Ctl3=win\_GetCtlHandle(\_Win,idc\_push\_button3),

Ctl4=win\_GetCtlHandle(\_Win,idc\_push\_button4),

Check=win\_GetCtlHandle(\_Win,idc\_check\_box3),

IsCheck= win\_IsChecked(Check),

if\_check(IsCheck,String),

StringB1 = win\_GetText(Ctl1),

StringB2 = win\_GetText(Ctl2),

StringB3 = win\_GetText(Ctl3),

StringB4 = win\_GetText(Ctl4),

Ctl=win\_GetCtlHandle(\_Win,idc\_expert),

button4(StringB4,String,StringB3,StringB2,StringB1,X),

win\_SetText(Ctl,X),

!.

%END Expert system, e\_Menu, Parent window

%END\_WIN Expert system