МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

Новгородский государственный университет имени Ярослава Мудрого (НовГУ) Кафедра «Электронных и информационных систем»

**Экспертные системы на языке Пролог**

Лабораторная работа по дисциплине:

«Системы искусственного интеллекта»

Студент гр. 1095

Золотов Э. В.

(подпись)

«\_\_\_»\_ 2024 г.

Преподаватель

\_\_Михайлов Д.В

(подпись)

« » 2024 г.

г. Великий Новгород

-2024-

# ЦЕЛЬ РАБОТЫ

Целью работы является изучение принципов построения и организации экспертных систем, базирующихся на логике и правилах.

# МОДЕЛЬ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ

Предметная область операционные системы. Данные ОС разделены на группы: Мобильные и Десктопные.

|  |  |  |
| --- | --- | --- |
| Группа | Операционная система | Свойства |
| Десктопная | Windows | Закрытый исходный код  Любые устройства  Windows NT семейство |
| Linux | Открытый исходный  Любые устройства  UNIX семейство |
| React OS | Открытый исходный код  Любые устройства  Windows NT семейство |
| macOS | Закрытый исходный код  Фирменные устройства  UNIX семейство |
| Solaris OS | Закрытый исходный код  Любые устройства  UNIX семейство |
| MS-DOS | Закрытый исходный код  Любые устройства  DOS семейство |
| freeDOS | Открытый исходный код  Любые устройства  DOS семейство |
| OS/2 | Закрытый исходный код  Фирменные устройства  OS/2 семейство |
| osFree | Открытый исходный код  Любые устройства  OS/2 семейство |
| Мобильная | iOS | Закрытый исходный код  Фирменные устройства |
| Android | Открытый исходный |
| Windows Phone | Закрытый исходный код  Windows NT семейство |

***Таблица 5. Описание предметной области***

# ОПИСАНИЕ ХАРАКТЕРИСТИК РАЗРАБОТАННЫХ ЭС

В зависимости от способов классификации и размещения информации Базы Знаний различают: продукционную, сетевую и фреймовую модели представления знаний.

Сетевая модель основана на представление знаний в виде сети, вершины которой соответствуют понятиям, а дуги — отношениям между ними.

В основе фреймовой модели лежит логическая группировка атрибутов объекта, при этом для хранения и обработки логические группы описываются во фреймах.

ЭС, базирующаяся на правилах, позволяет проектировщику строить правила, которые естественным образом объединяют в группы связанные фрагменты знаний.

При реализации ЭС, базирующейся на логике, БЗ представляется совокупностью утверждений в виде фактов. Вывод экспертного заключения при этом строится на основе стандартных средств работы со списками.

# ДИАГРАММА ПОТОКОВ ДАННЫХ

# 

# ТЕСТОВЫЕ НАБОРЫ ДАННЫХ

# *Для экспертной системы на правилах:*

"Вопрос : Используется на десктопных устройствах?

1 - да,

2 - нет

|: 1

Вопрос : Имеет закрытый исходный код?

1 - да,

2 - нет

|: 1

Вопрос : Устанавливается на любые устройства?

1 - да,

2 - нет

|: 2

Вопрос : Имеет открытый исходный код?

1 - да,

2 - нет

|: 2

Вопрос : Устанавливается на только фирменные устройства?

1 - да,

2 - нет

|: 1

Вопрос : Относится к семейству UNIX систем?

1 - да,

2 - нет

|: 1

Рекомендуемая Операционная Система : macOS."

# *Для экспертной системы на логике:*

"Вопрос : десктопная ОС?

1 - да,

2 - нет

|: 1

Вопрос : закрытый исходный код?

1 - да,

2 - нет

|: 2

Вопрос : открытый исходный код?

1 - да,

2 - нет

|: 1

Вопрос : любые устройства?

1 - да,

2 - нет

|: 1

Вопрос : UNIX семейство?

1 - да,

2 - нет

|: 2

Вопрос : Windows NT семейство?

1 - да,

2 - нет

|: 2

Вопрос : DOS семейство?

1 - да,

2 - нет

|: 2

Вопрос : OS/2 семейство?

1 - да,

2 - нет

|: 1

Рекомендуемая Операционная Система: osFree."

# ОПИСАНИЕ МЕХАНИЗМОВ ВЫВОДА

Написание на Visual Prolog-е ЭС, базирующейся на правилах, начинается с декларации БД. БД хранит ответы пользователя на вопросы СПИ. Эти данные являются утвердительными или отрицательными ответами. Далее строятся продукционные правила, описывающие фрагменты фактического знания.

Запросы пользователя трансформируются в форму, сопоставимую с формой правил БЗ. Механизм вывода инициализирует процесс сопоставления, начиная с «верхнего» правила. Обращение к правилу называется «вызовом». Вызов соответствующих правил в процессе сопоставления продолжается до тех пор, пока не произошло сопоставление или не исчерпана вся БЗ, а значение не найдено. Если МВ обнаруживает, что можно вызвать более одного правила, то запускается процесс разрешения конфликта. При разрешении конфликта приоритет отдается обычно тем правилам, которые более конкретны, либо правилам, которые учитывают больше текущих данных.

Экспертная Система, базирующаяся на логике, содержит в БЗ предложения, которые управляют поиском и сопоставлением. Интерпретатор сопоставляет эти предложения с элементами данных в БД.

Если существует возможность вызова более одного правила, то для разрешения конфликта система использует возможности механизма внутренней унификации Пролога.

Система получает результаты унификационного процесса автоматически, поэтому они направляются на нужное (логическое) устройство вывода информации.

Так же, как и в ЭС, базирующейся на правилах, данный циклический процесс является процессом распознавание-действие.

Основное отличие структуры ЭС, базирующейся на логике, состоит в описании объектов и атрибутов в виде фактов.

# ВЫВОД

Экспертные Системы, базирующиеся на правилах проще в разработке и первоначальном наполнение знаниями, Экспертные Системы, базирующиеся на логике, в свою очередь требуют больших усилий для изначальной разработки и наполнения, однако позволяют более гибко осуществлять добавление новых записей.

# ПРИЛОЖЕНИЕ A

# Код программы prod1.pl

:- dynamic xpositive/**2**, xnegative/**2**.

/\* Система пользовательского интерфейса \*/

**do\_expert\_job**:-

do\_consulting,

halt.

**do\_consulting**:-

**os\_is**(X),!,

**format**("Рекомендуемая Операционная Система : ~a.~n",[X]),

clear.

**do\_consulting**:-

**write**("Информация об интересующей Вас Операционной Системе отсутствует в БЗ."),nl,

clear.

**ask**(X,Y):-

**format**("Вопрос : ~a ~a?~n",[X,Y]),

**write**("1 - да,"),nl,

**write**("2 - нет"),nl,

**readloop**(Reply),

**remember**(X,Y,Reply).

/\* Проверка корректности ввода \*/

**readloop**(Response):-

%read(Response),

**read\_string**(user\_input, "\n", "\r",**\_**,Str),

**atom\_number**(Str, Response),

**legal\_response**(Response),!.

**legal\_response**(Response):-Response=**1**.

**legal\_response**(Response):-Response=**2**.

/\* Механизм вывода экспертного заключения \*/

**positive**(X,Y):-

**xpositive**(X,Y),!.

**positive**(X,Y):-

not(**getnegative**(X,Y)),!,**ask**(X,Y).

**getpositive**(X,Y):-

**xpositive**(X,Y),!.

**getnegative**(X,Y):-

**xnegative**(X,Y),!.

**negative**(X,Y):-

**xnegative**(X,Y),!.

**negative**(X,Y):-

not(**getpositive**(X,Y)),!,not(**ask**(X,Y)).

**remember**(X,Y,**1**):-!,

**assertz**(**xpositive**(X,Y)).

**remember**(X,Y,**2**):-!,

**assertz**(**xnegative**(X,Y)),fail.

/\* Продукционные правила \*/

**os\_is**("Windows"):-

**it\_is**("Десктопная"),

**positive**("Имеет", "закрытый исходный код"),

**positive**("Устанавливается на","любые устройства"),

**positive**("Относится к семейству", "Windows NT систем"),!.

**os\_is**("Linux"):-

**it\_is**("Десктопная"),

**positive**("Имеет", "открытый исходный код"),

**positive**("Устанавливается на", "любые устройства"),

**positive**("Относится к семейству", "UNIX систем"),!.

**os\_is**("React OS"):-

**it\_is**("Десктопная"),

**positive**("Имеет", "открытый исходный код"),

**positive**("Устанавливается на", "любые устройства"),

**positive**("Относится к семейству", "Windows NT систем"),!.

**os\_is**("macOS"):-

**it\_is**("Десктопная"),

**positive**("Имеет", "закрытый исходный код"),

**positive**("Устанавливается на", "только фирменные устройства"),

**positive**("Относится к семейству", "UNIX систем"),!.

**os\_is**("Solaris OS"):-

**it\_is**("Десктопная"),

**positive**("Имеет", "закрытый исходный код"),

**positive**("Устанавливается на", "любые устройства"),

**positive**("Относится к семейству", "UNIX систем"),!.

**os\_is**("MS-DOS"):-

**it\_is**("Десктопная"),

**positive**("Имеет", "закрытый исходный код"),

**positive**("Устанавливается на", "любые устройства"),

**positive**("Относится к семейству", "DOS систем"),!.

**os\_is**("freeDOS"):-

**it\_is**("Десктопная"),

**positive**("Имеет", "открытый исходный код"),

**positive**("Устанавливается на", "любые устройства"),

**positive**("Относится к семейству", "DOS систем"),!.

**os\_is**("OS/2"):-

**it\_is**("Десктопная"),

**positive**("Имеет", "закрытый исходный код"),

**positive**("Устанавливается на", "только фирменные устройства"),

**positive**("Относится к семейству", "OS/2 систем"),!.

**os\_is**("osFree"):-

**it\_is**("Десктопная"),

**positive**("Имеет", "открытый исходный код"),

**positive**("Устанавливается на", "любые устройства"),

**positive**("Относится к семейству", "OS/2 систем"),!.

**os\_is**("iOS"):-

**it\_is**("Мобильная"),

**positive**("Имеет", "закрытый исходный код"),

**positive**("Устанавливается на", "только фирменные устройства"),!.

**os\_is**("Android"):-

**it\_is**("Мобильная"),

**positive**("Имеет", "открытый исходный код"),!.

**os\_is**("Windows Phone"):-

**it\_is**("Мобильная"),

**positive**("Имеет", "закрытый исходный код"),

**positive**("Относится к семейству", "Windows NT систем"),!.

**it\_is**("Десктопная"):-

**positive**("Используется на", "десктопных устройствах"),!.

**it\_is**("Мобильная"):-

**positive**("Используется на", "мобильных устройствах"),!.

/\* Уничтожение в базе данных всех ответов yes (да) и no (нет) \*/

**clear**:-**retract**(**xpositive**(**\_**,**\_**)),**retract**(**xnegative**(**\_**,**\_**)),fail,!.

clear.

:- do\_expert\_job.

# Код программы log1.pl

:- dynamic yes/**1**, no/**1**.

/\* База знаний. \*/

/\* Размещение в резидентной БД информации из утверждений БЗ ЭС \*/

**assert\_database**:-

**rule**(Rule\_number,Category,Type\_of\_breed,Conditions),

**assertz**(**d\_rule**(Rule\_number,Category,Type\_of\_breed,Conditions)),fail.

**assert\_database**:-

**cond**(Cond\_number,Condition),

**assertz**(**d\_cond**(Cond\_number,Condition)),fail.

**assert\_database**:-!.

/\* Условия-характеристики различных ОС.\*/

**cond**(**1**,"десктопная ОС").

**cond**(**2**,"мобильная ОС").

**cond**(**3**,"закрытый исходный код").

**cond**(**4**,"открытый исходный код").

**cond**(**5**,"любые устройства").

**cond**(**6**,"фирменные устройства").

**cond**(**7**, "UNIX семейство").

**cond**(**8**, "Windows NT семейство").

**cond**(**9**, "DOS семейство").

**cond**(**10**, "OS/2 семейство").

/\* Данные о типах ОС \*/

**topic**("десктопная").

**topic**("мобильная").

/\* Данные о конкретных ОС \*/

**rule**(**1**,"ОС","десктопная",[**1**]).

**rule**(**2**,"ОС","мобильная",[**2**]).

**rule**(**3**,"десктопная","Windows",[**3**,**5**,**8**]).

**rule**(**4**,"десктопная","Linux",[**4**,**5**,**7**]).

**rule**(**5**,"десктопная","macOS",[**3**,**6**,**7**]).

**rule**(**6**,"десктопная","SolarisOS",[**3**,**5**,**7**]).

**rule**(**7**,"десктопная","ReactOS",[**4**,**5**,**8**]).

**rule**(**8**,"десктопная","MS-DOS",[**3**,**5**,**9**]).

**rule**(**9**,"десктопная","freeDOS",[**4**,**5**,**9**]).

**rule**(**10**,"десктопная","OS/2",[**3**,**6**,**10**]).

**rule**(**11**,"десктопная","osFree",[**4**,**5**,**10**]).

**rule**(**12**,"мобильная","iOS",[**3**,**6**]).

**rule**(**13**,"мобильная","Android",[**4**]).

**rule**(**14**,"мобильная","WindowsPhone",[**3**,**8**]).

/\* Система пользовательского интерфейса \*/

**main**:-

assert\_database,

do\_expert\_job.

**do\_expert\_job**:-

do\_consulting,

halt.

**do\_consulting**:-

**go**([],"ОС"),!.

**do\_consulting**:-

**write**("Информация об интересующей вас Операционной Системе отсутствует в БЗ."), nl,

clear.

/\* Выдача подсказки \*/

**info**:-

**write**("База знаний содержит информацию о категориях Операционных Систем:"), nl,

**topic**(X), **format**('\t~w~n',[X]).

/\* Запрос и получение ответов yes и no от пользователя \*/

**ask\_question**(Breed\_cond,Text):-

**format**("Вопрос : ~a?~n",[Text]),

**write**("1 - да,"),nl,

**write**("2 - нет"),nl,

**readloop**(Response),

**do\_answer**(Breed\_cond,Response).

/\* Проверка корректности ввода \*/

**readloop**(Response):-

%read(Response),

**read\_string**(user\_input, "\n", "\r",**\_**,Str),

**atom\_number**(Str, Response),

**legal\_response**(Response),!.

**legal\_response**(Response):-Response=**1**.

**legal\_response**(Response):-Response=**2**.

/\* Предикаты ЕЯ-интерфейса \*/

/\* Принадлежность элемента списку \*/

**member**(Head,[Head|**\_**]):-!.

**member**(Elem,[**\_**|T]):-

**member**(Elem,T).

/\* Механизм вывода \*/

/\* Начальное правило механизма вывода \*/

**go**(**\_**,Mygoal):-

not(**rule**(**\_**,Mygoal,**\_**,**\_**)),!,

**format**("Рекомендуемая Операционная Система: ~a.~n",[Mygoal]).

**go**(History,Mygoal):-

**rule**(Rule\_number,Mygoal,Type\_of\_breed,Conditions),

**check**(Rule\_number,History,Conditions),

**go**([Rule\_number|History],Type\_of\_breed).

/\* Сопоставление входных данных пользователя со списками атрибутов

отдельных языков программирования \*/

**check**(Rule\_number,History,[Breed\_cond|Rest\_breed\_cond\_list]):-

**yes**(Breed\_cond),!,

**check**(Rule\_number,History,Rest\_breed\_cond\_list).

**check**(**\_**,**\_**,[Breed\_cond|**\_**]):-

**no**(Breed\_cond),!,fail.

**check**(Rule\_number,History,[Breed\_cond|Rest\_breed\_cond\_list]):-

**cond**(Breed\_cond,Text),

**ask\_question**(Breed\_cond,Text),

**check**(Rule\_number,History,Rest\_breed\_cond\_list).

**check**(**\_**,**\_**,[]).

**do\_answer**(Cond\_number,**1**):-!,

**assertz**(**yes**(Cond\_number)).

**do\_answer**(Cond\_number,**2**):-!,

**assertz**(**no**(Cond\_number)),fail.

/\* Исключение данных из базы знаний после завершения цикла "Распознавание-действие" \*/

**erase**:-**retract**(**\_**),fail.

erase.

/\* Уничтожение в базе данных всех ответов yes (да) и no (нет) \*/

**clear**:-**retract**(**yes**(**\_**)),**retract**(**no**(**\_**)),**retract**(**goes**(**\_**,**\_**)),fail,!.

clear.

:- main.