



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(ДГТУ)**

Кафедра «Робототехника и мехатроника»

**МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ДЛЯ ПРАКТИЧЕСКИХ И
САМОСТОЯТЕЛЬНЫХ РАБОТ ПО ДИСЦИПЛИНЕ
«РАЗРАБОТКА ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ
ПРИНЯТИЯ РЕШЕНИЙ»**

Ростов-на-Дону

2022

Составители: к.т.н., доц. Лукьянов Е.А.

Методические рекомендации для практической и самостоятельной работы по дисциплине «Разработка интеллектуальных систем принятия решений» для студентов магистратуры направления подготовки 15.04.06 Мехатроника и робототехника.

Содержатся темы практических занятий, контрольные вопросы и задания для изучения дисциплины.

Введение

Целями освоения дисциплины «Разработка интеллектуальных систем принятия решений» является формирование у студентов комплекса знаний и практических основ в области создания и применения систем поддержки принятия решений.

Семантические сети являются одним из инструментов, который используется для представления знаний и выполнения некоторых операций над знаниями.

Темы практических занятий

Построение семантической сети /Пр/ 8

Фреймы /Пр/ 4

Продукционные системы /Пр/ 4

Нечеткий логический вывод /Пр/ 8

Кластеризация методом с-средних /Пр/

Базовым принципом языка является равнозначность представления программы и данных (декларативность), отчего утверждения языка одновременно являются и записями, подобными записям в базе данных, и правилами, несущими в себе способы их обработки. Сочетание этих качеств приводит к тому, что по мере работы Prolog-системы знания накапливаются. Накапливаются как данные, так и правила. Поэтому Prolog-системы считают естественной средой для накопления базы знаний. База знаний — важный компонент интеллектуальной системы. Они предназначены для поиска способов решения проблем из некоторой предметной области, основываясь на записях базы знаний и на пользовательском описании ситуации.

Интерфейс Visual Prolog включает: главное меню, панель инструментов,

окно проекта. Если во время последнего использования системы Visual Prolog там был открытый проект, то система автоматически вновь откроет этот проект.

На рис.1.1 изображен внешний вид среды Visual Prolog после запуска. В окне проекта отображаются модули открытого проекта route.prj: karta.pro, route.pro, VPITools.pro.

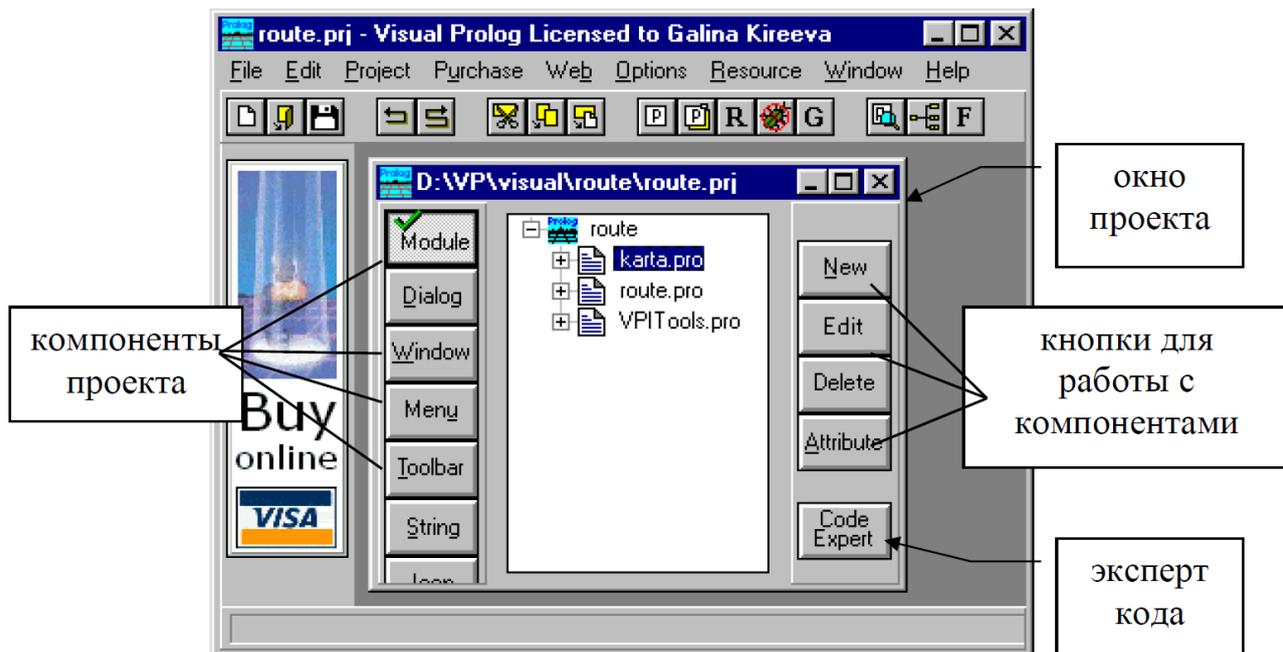


рис.1.1 Среда разработки Visual Prolog 6.1

Левая панель кнопок в окне проекта позволяет выбирать нужный компонент проекта: модуль, окно, меню и т.д. С помощью кнопок правой панели выбранный компонент можно редактировать(кнопка Edit), удалять(кнопка Delete), а также добавлять новый(кнопка New).

Пункт меню File содержит команды для работы с файлами. Чтобы создавать новое окно редактирования, можно использовать команду File | New.

Эта команда создаст новое окно редактора с заголовком "NONAME".

В меню Edit представлены команды, позволяющие редактировать текст программы. Встроенный редактор системы по интерфейсу похож на обычный текстовый редактор. Можно производить вырезку, копирование и вставку текста, операции Отмена/Восстановление, которые можно активизировать меню Edit.

Также меню Edit показывает "горячие клавиши", связанные для этих действий.

Пункт меню Project содержит команды для работы с проектом: создать новый, открыть, запустить и т.д. Запуск проекта на исполнение выполняется нажатием кнопки <R> на панели инструментов (или F9, или с помощью команд меню Project | Run).

Команды меню Options позволяют выполнять настройку проекта, устанавливать необходимые параметры.

Среда Visual Prolog позволяет протестировать программу без создания проекта. Для этого используется утилита Test Goal. Достаточно создать новый файл, набрать текст программы и активизировать Test Goal нажатием кнопки <G> на панели инструментов. Автономно исполняемый файл при этом не создается. Утилита Test Goal компилирует только тот код, который определен в активном окне редактора (код в других открытых окнах или модулях проектов, если они есть, игнорируются). Test Goal находит все возможные решения задачи и автоматически выводит значения всех переменных.

Программа на ПРОЛОГе состоит из предложений, которые могут быть фактами, правилами или запросами. Факт – это утверждение о том, что соблюдается некоторое конкретное соотношение между объектами. Факт используется для того, чтобы показать простую взаимосвязь между данными.

Структура факта:

<имя_отношения>(t1,t2,...,tn)), t1,t2,...,tn– объекты

Примеры фактов:

учится (ира, университет). % Ира учится в университете

родитель(иван, алексей). % Иван является родителем Алексея

язык_программирования (пролог). % Пролог – это язык программирования

Набор фактов составляет базу данных. В виде факта в программе

записываются данные, которые принимаются за истину и не требуют доказательства.

Правила используются для того, чтобы установить отношения между объектами на основе имеющихся фактов.

Структура правила:

<имя_правила> :- <тело правила> или

<имя_правила > if <тело правила>

Левая часть правила вывода называется головой правила, а правая часть - телом. Тело может состоять из нескольких условий, перечисленных через запятую или точку с запятой. Запятая означает операцию «логическое И», точка с запятой – операцию «логическое ИЛИ». В предложениях используются переменные для обобщенной формулировки правил вывода. Переменные действуют только в одном предложении. Имя в разных предложениях указывает на разные объекты. Все предложения обязательно заканчиваются точкой.

Примеры правил:

мать (X, Y) :- родитель (X, Y), женщина(X).

студент (X) :- учится (X, институт); учится (X, университет).

Правило отличается от факта тем, что факт - всегда истина, а правило является истинным, если выполняются все утверждения, составляющие тело правила. Факты и правила образуют базу знаний.

Переменные служат для обозначения объектов, значения которых меняются в ходе выполнения программы. Имена переменных начинаются с заглавных букв или знака «_» Область действия переменной – предложение.

Одноименные переменные в разных предложениях могут иметь разные значения.

Специальным знаком «_» обозначается анонимная переменная, которая используется тогда, когда конкретное значение переменной не существенно для данного предложения. Значение анонимной переменной не выводится на печать.

Если имеется база данных, то можно написать запрос (цель) к ней. Запрос

— это формулировка задачи, которую программа должна решить. Его структура такая же, как у правила или факта. Существуют запросы с константами и запросы с переменными.

Запросы с константами позволяют получить один из двух ответов: “да” или “нет”. Если в запрос входит переменная, то интерпретатор пытается найти такие ее значения, при которых запрос будет истинным. Запросы могут быть составными, т.е. состоять из нескольких простых запросов. Они могут объединяться знаком ‘,’, который понимается как логическая связка И или знаком ‘;’(логическое ИЛИ). Простые запросы называются подцелью, составной запрос принимает истинное значение тогда, когда истинна каждая подцель.

Как правило, программа состоит из четырех разделов.

DOMAINS – секция описания доменов(типов). Секция применяется, если в программе используются нестандартные домены.

PREDICATES – секция описания предикатов. Секция применяется, если в программе используются нестандартные предикаты.

CLAUSES – секция предложений. Именно в этой секции записываются предложения: факты и правила вывода.

GOAL – секция цели. В этой секции записывается запрос.

Пример 1.

Имеется база данных, содержащая следующие факты:

родитель(илья, марина).
родитель(марина, ира).
родитель(елена, иван).
родитель(николай, ира).
родитель(ольга, алексей).
родитель(марина, саша).
родитель(сергей, иван).

Определить:

- 1) верно ли, что Марина является родителем Саши;
- 2) верно ли, что Алексей является родителем Ольги;
- 3) кто является ребенком Николая;
- 4) кто родители Ивана;

5) всех родителей и их детей.

Решение.

1. Запустите среду Visual Prolog. Закройте окно проекта (если оно открыто) и откройте новый файл (File|New) (рис.2)

В появившемся окне наберите текст программы, содержащий разделы: PREDICATES (описание предиката родитель), CLAUSES (перечисляются имеющиеся факты) и GOAL (запрос).



рис.2. Рабочее окно редактора

DOMAINS

имя=symbol

PREDICATES

pondeterm родитель(имя, имя)

CLAUSES

родитель(илья, марина).

родитель(марина, ира).

родитель(елена, иван).

родитель(николай, ира).

родитель(ольга, алексей).

родитель(марина, саша).

родитель(сергей, иван).

GOAL

родитель(марина, саша) .

Запустите и протестируйте программу с помощью команды Project | Test Goal (можно использовать кнопку на панели инструментов <G> или сочетание клавиш <Ctrl>+<G>). Результат выполнения программы будет выведен в отдельном окне

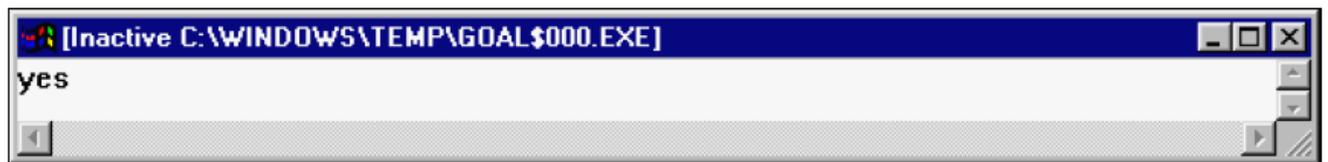


рис3. Окно вывода результата

Указание: перед следующим запуском программы следует закрыть это окно.

Указание: перед следующим запуском программы следует закрыть это окно.

2. Для ответа на вопрос: верно ли, что Алексей является родителем Ольги, измените запрос:

GOAL

родитель (алексей, ольга) .

После запуска программы (Project | Test Goal) будет получен ответ:

no

3. Для ответа на вопрос: кто является ребенком Николая, запишите цель:

GOAL

родитель (николай, X) .

Результат:

X=ира

1 Solution

4. Для ответа на вопрос: кто родители Ивана, укажите запрос:

GOAL

родитель (X, иван) , родитель (Y, иван) , X<>Y.

Результат:

X=елена, Y=сергей

X=сергей, Y=елена

2 Solutions

5. Для определения всех родителей и их детей, запишите:

GOAL

родитель (X, Y) .

Результат:

X=илья, Y=марина

X=марина, Y=ира

X=елена, Y=иван

X=николай, Y=ира

X=ольга, Y=алексей

X=марина, Y=саша

X=сергей, Y=иван

7 Solutions

Пример 2

Имеются факты вида: *родитель(имя, имя)* и *женщина(имя)*.

а) составить правило *мать* и определить, кто мать Маши.

б) составить правило *бабушка* и определить, кто бабушка Ирины.

в) составить правило *внучка* и определить, сколько внучек у Ольги и

как их зовут

Решение:

DOMAINS

имя= symbol

PREDICATES

nondeterm родитель (имя, имя)

женщина (имя)

nondeterm мать (имя, имя)

nondeterm бабушка (имя, имя)

nondeterm внучка (имя, имя)

CLAUSES

родитель (марина, ирина) .

родитель (елена, анна) .

родитель (ольга, марина) .

родитель (ольга, татьяна) .

родитель (татьяна, катя) .

родитель (анна, маша) .

женщина (ольга) .
женщина (маша) .
женщина (ирина) .
женщина (елена) .
женщина (анна) .
женщина (марина) .
женщина (татьяна) .
женщина (катя) .
мать (X, Y) : -родитель (X, Y) , женщина (X) .
бабушка (X, Z) : -мать (X, Y) , родитель (Y, Z) .
внучка (X, Y) : -бабушка (Y, X) , женщина (X) .

A) GOAL

мать (Кто, маша) .

Результат:

Кто=анна
1 Solution

B) GOAL

бабушка (Кто, ирина) .

Результат:

Кто=ольга
1 Solution

B) GOAL

внучка (Кто, ольга) .

Результат:

Кто=ирина
Кто=катя
2 Solutions

Замечание: ключевое слово *nondeterm* определяет недетерминированные предикаты, которые могут совершать откат назад и генерировать множественные решения. Таким образом, если задача предполагает возможность получения несколько решений, следует объявлять предикаты как недетерминированные.

Пример 3

Записать по правилам Пролога следующие факты:

Билл играет в теннис, баскетбол и футбол

Майк играет в футбол и хоккей.

Сформулировать запросы, выясняющие:

а) кто играет в футбол

- б) во что играет Майк
- в) во что играют и Билл, и Майк

Решение:

```
DOMAINS
    имя, спорт=symbol
PREDICATES
    play(имя, спорт)
CLAUSES
    play(билл, теннис) .
    play(билл, баскетбол) .
    play(билл, футбол) .
    play(майк, хоккей) .
    play(майк, футбол) .
```

A) GOAL

```
play(X, футбол) .
```

Результат:

```
X=билл
X=майк
2 Solutions
```

B) GOAL

```
play(майк, X) , play(билл, X) .
```

Результат:

```
X=футбол
1 Solution
```

Создание проекта позволяет протестировать пример как автономную исполняемую программу. После запуска проекта на исполнение создается ехе-файл, работа которого завершается после *первого* решения, удовлетворяющего решению задачи. Запуск программы в этом режиме не обеспечивает автоматический вывод значений переменных, поэтому необходимо использовать стандартный предикат вывода **write**.

Пример 4.

Заданы отношения-факты:

- родитель("Иван", "Катя").
- родитель("Анна", "Олег").
- родитель("Олег", "Дима").
- родитель("Игорь", "Ольга").
- родитель("Олег", "Виктор").
- родитель("Игорь", "Иван").
- мужчина("Дима").
- мужчина("Иван").
- мужчина("Игорь").
- мужчина("Олег").
- мужчина("Виктор").
- женщина("Катя").
- женщина("Ольга").
- женщина("Анна").

Составить новое отношение-правило $ded(X,Y)$ и определить, кто является дедушкой Кати. Создать проект и протестировать пример как автономную исполняемую программу.

Решение

1. Запустите среду Visual Prolog и создайте новый проект (Project | New Project), активизируется окно **Application Expert** (эксперт приложения).
2. Определите имя проекта (Primer) и базовый каталог, куда будет сохранен проект (например, D:\VP\Primer)

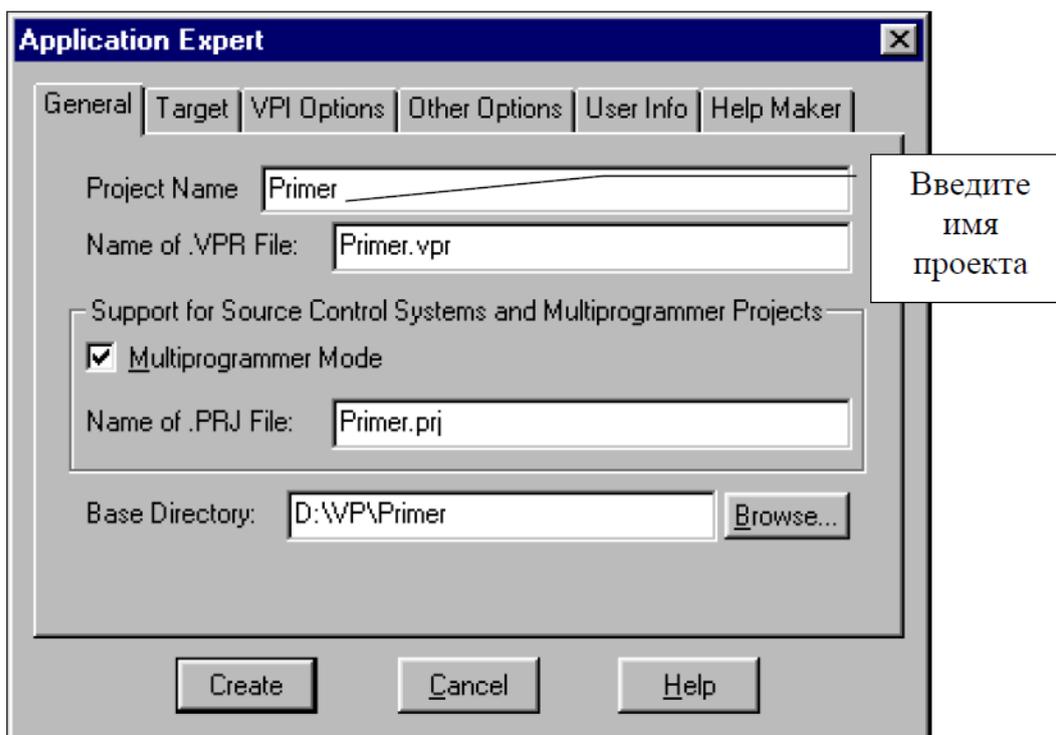


рис.4. Окно **Application Expert**

На вкладке **Target** установите параметры и нажмите кнопку **Create** для создания проекта (рис. 5):

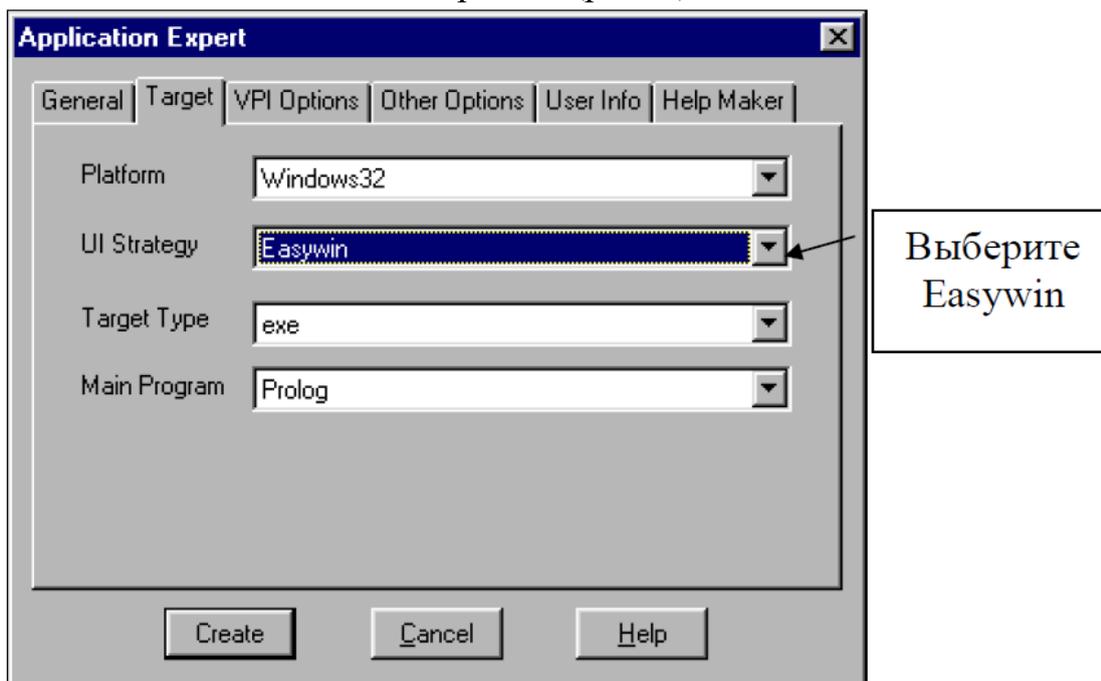


рис.5. Установки на вкладке **Target** окна **Application Expert**

3. Откройте окно **Compiler Options** (**Options** | **Project** | **Compiler Options**), откройте вкладку **Warnings** и установите опции компилятора для созданного проекта (рис.6):

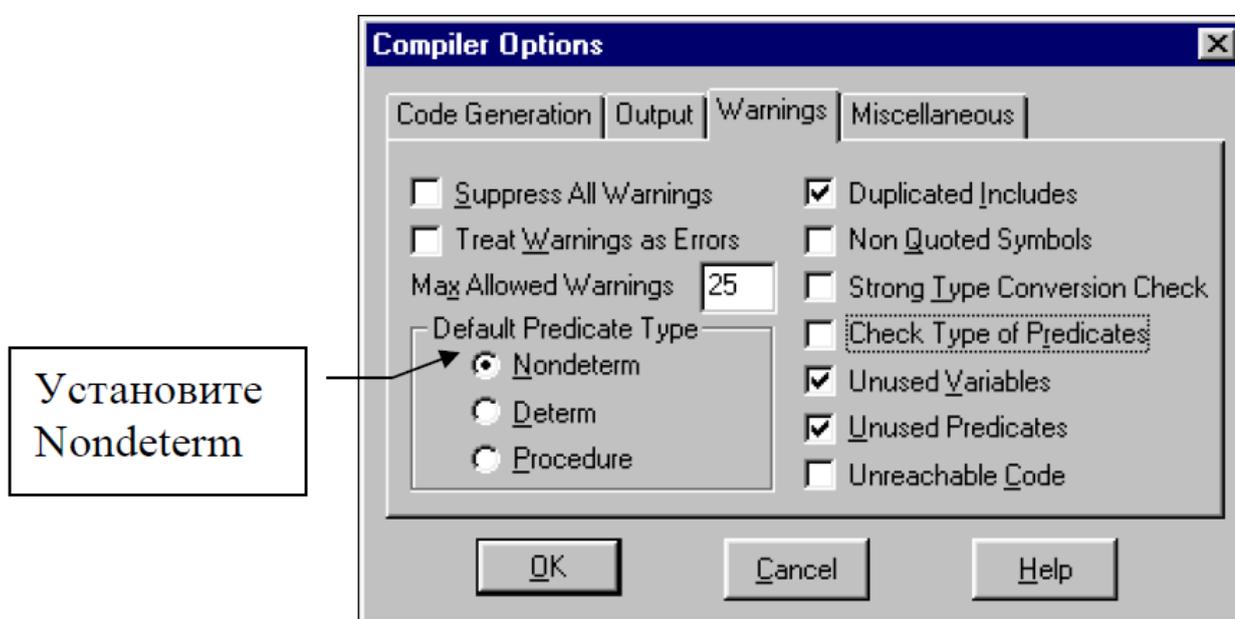


рис.6. Установки опций компилятора

Нажмите **OK**.

4. В окне проекта выделите файл **Primer.pro** и откройте его для редактирования (двойной щелчок или кнопка **Edit**)



рис.7. Окно проекта

Файл с расширением .pro содержит секции PREDICATES, GOAL, CLAUSES. Допишите необходимые определения так, чтобы получилась программа:

DOMAINS

имя=string

PREDICATES

родитель(имя,имя)

женщина(имя)

мужчина(имя)

дед(имя, имя)

CLAUSES

родитель("Иван","Катя").

родитель("Анна","Олег").

родитель("Олег","Дима").

родитель("Игорь","Ольга").

родитель("Олег","Виктор").

родитель("Игорь","Иван").

мужчина("Дима").

мужчина("Иван").

мужчина("Игорь").

мужчина("Олег").

мужчина("Виктор").

женщина("Катя").

женщина("Ольга").

женщина("Анна").

дед(X,Z):-родитель(X,Y), родитель(Y,Z), мужчина(X).

GOAL

дед(X,"Катя"),write(X).

5. Откомпилируйте исходный код примера и запустите его как автономную исполняемую программу. (Project | Run, или клавиша <F9>, или кнопка <R>). Результат выполнения программы должен отобразиться в окне:



рис.8. Окно вывода результата

ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

1. Запишите по правилам Пролога следующие факты:

Ник подарил Тому книгу.

Мэри подарила Тому ручку.

Рик подарил Мэри игрушку.

Боб подарил Пэт игрушку.

Сформулируйте запросы, выясняющие:

- Правда ли, что Рик подарил Мэри игрушку?
- Правда ли, что Ник подарил Тому игрушку?
- Что подарила на день рождения Тому Мэри?
- Что подарили Тому на день рождения?
- Кто подарил Пэт игрушку?

2. Задано дерево родственных отношений:



Запишите данные факты по правилам ПРОЛОГа и сформулируйте запросы, выясняющие:

- a) Является ли Федор родителем Ольги?
- b) Кто является родителем Татьяны?
- c) Кто дети Ивана?
- d) Кто является родителем родителя Ольги?
- e) Кто внуки Ивана?
- f) Есть ли у Федора и Степана общий родитель?
- g) Кто чей родитель?

3. Имеется база данных, содержащая факты вида: любит(имя, продукт), фрукты(продукт), конфеты(продукт).

Составьте программу, определяющую:

- a) всех, кто любит бананы;
- b) кто любит и шоколад, и яблоки;
- c) что любит Вова;
- d) что любят и Света, и Вова.

Используя имеющиеся факты, составить новое правило люб_фрукты(X) и определить всех, кто любит фрукты;

4. Имеется база данных, содержащая факты вида: играет(имя, спорт), мужчина(имя), женщина(имя).

Составьте программу, определяющую:

- a) каким видом спорта увлекается Андрей;

- b) всех, кто играет в волейбол;
- c) каким видом спорта увлекаются и Ольга, и Саша;
- d) кто увлекается и футболом, и волейболом.

Используя имеющиеся факты, составить новое правило волейбол_жен(X) и определить всех женщин, играющих в волейбол.

5. Реализуйте проект примера 4 и доработайте исходный код следующим образом:

- 1) добавьте новое правило бабушка и определите, кто является бабушкой;
- 2) добавьте новое правило внук и определите, кто внук Анны;
- 3) добавьте новое правило брат и определите, кто брат Димы;
- 4) добавьте новое правило сестра и определите, кто сестра Ивана.

Список источников по дисциплине

6. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)				
6.1. Рекомендуемая литература				
6.1.1. Основная литература				
	Авторы, составители	Заглавие	Издательство, год	Количество
Л1.1	Кудинов, Ю.И.	Интеллектуальные системы: учебное пособие	Липецк: Липецкий государственный технический университет, ЭБС АСВ, 2014	ЭБС
Л1.2	Пальмов, С.В.	Интеллектуальные системы и технологии: учебное пособие	Самара: Поволжский государственный университет телекоммуникаций и информатики, 2017	ЭБС
Л1.3	Пальмов, С.В.	Интеллектуальный анализ данных: учебное пособие	Самара: Поволжский государственный университет телекоммуникаций и информатики, 2017	ЭБС
6.1.2. Дополнительная литература				
	Авторы, составители	Заглавие	Издательство, год	Количество
Л2.1	Семенов, А.М., Соловьев, Н.А.	Интеллектуальные системы: учебное пособие	Оренбург: Оренбургский государственный университет, ЭБС АСВ, 2013	ЭБС
Л2.2	Е.Н. Чуйкова	Интеллектуальные информационные системы: учебное пособие	ДГТУ, 2011	ЭБС
6.1.3. Методические разработки				
	Авторы, составители	Заглавие	Издательство, год	Количество
Л3.1		Правила-продукции: метод. указания к практической работе по дисциплине «Интеллектуальные системы управления»	Ростов н/Д.: ДГТУ, 2018	ЭБС
Л3.2		Фреймы: метод. указания к практической работе по дисциплине «Интеллектуальные системы управления»	Ростов н/Д.: ДГТУ, 2018	ЭБС
Л3.3		Представление нечетких знаний: метод. указания к практической работе по дисциплине «Интеллектуальные системы»	Ростов н/Д.: ДГТУ, 2018	ЭБС
Л3.4		Руководство для преподавателей по организации и планированию различных видов занятий и самостоятельной работы обучающихся в Донском государственном техническом университете: метод. указания	Ростов н/Д.: ИЦ ДГТУ, 2018	ЭБС
6.2. Перечень ресурсов информационно-телекоммуникационной сети "Интернет"				
Э1	Электронно-библиотечная система (ЭБС) НТБ ДГТУ (https://ntb.donstu.ru/) через которую (на основании заключенных договоров) осуществляется доступ на порталы: ЭБС «Лань» (http://e.lanbook.com), ЭБС «Университетская библиотека online» (http://www.biblioclub.ru), ЭБС IPRbooks (http://www.iprbookshop.ru/), «Юрайт» (http://www.biblio-online.ru/), ЭБС «ZNANIUM.COM» (www.znanium.com), Электронная библиотека Grebennikon (http://grebennikon.ru), ЭБС «РУКОНТ» (http://lib.rucont.ru)			
Э2	Электронная образовательная среда ДГТУ (http://skif.donstu.ru/)			
6.3 Перечень информационных технологий				
6.3.1 Перечень программного обеспечения				
6.3.1.1	Mathworks (в составе: MATLAB (MathWorks SMS- Software Maintenance Service), Simulink, Control System Toolbox, Neural Network Toolbox, Fuzzy Logic Toolbox, Optimization Toolbox, Partial Differential Equation Toolbox, Signal Processing Toolbox, Simscape Multibody, Simscape, Symbolic Math Toolbox, Statistics and Machine Learning Toolbox, System Identification Toolbox			

