|  |  |
| --- | --- |
| |  | | --- | | МИНОБРНАУКИ РОССИИ  Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | |

Институт Информационных технологий

Кафедра МОСИТ

**ЛАБОРАТОРНАЯ РАБОТА №1**

по дисциплине «Функциональное и Логическое Программирование»

**Студент группы** ИКБО-12-17 Белоусов И.С. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(подпись студента)*

**Руководитель работы** Смольянинова В.А. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(подпись руководителя)*

Москва, 2019

# Постановка задачи

**Задание к примерам программ на языке Пролог:**

Наберите программу. Запустите на выполнение. Придумайте и задайте 5-6 вопросов к каждой программе.

Примеры:

1. Родственные отношения
   1. Нарисовать схему родства
2. «нравится»
3. «может купить»
4. Машины

**Задания для самостоятельного выполнения:**

1. Составьте программу Родственные отношения, которая кроме родственных отношений parent (родитель) и ancestor (предок) программа должна содержать одно или несколько из следующих отношений: brother (брат); sister (сестра); grand-father (дедушка); grand-mother (бабушка); uncle (дядя).
2. Составьте программу, используя отношения likes «нравится») и can\_buy («может купить»).
3. Составьте собственную программу, состоящую из фактов и правил. Проверьте ее работу.

# Ход выполнения работы

**Примеры. Схема родства.**

Разберём пример «Схема родства».

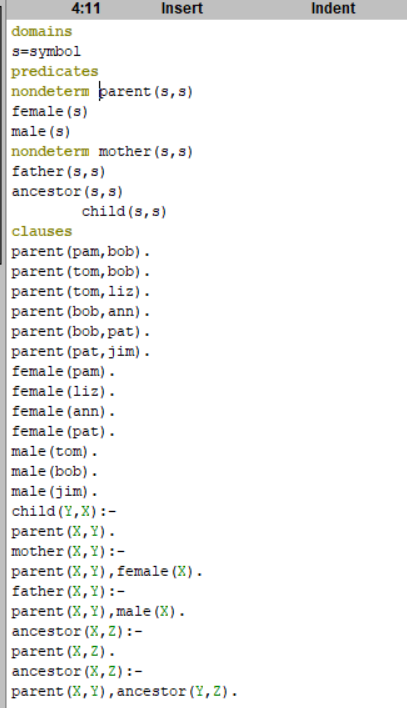
****

Рисунок 1. Код примера 1

Отразим существующие родственные связи на соответствующей схеме.

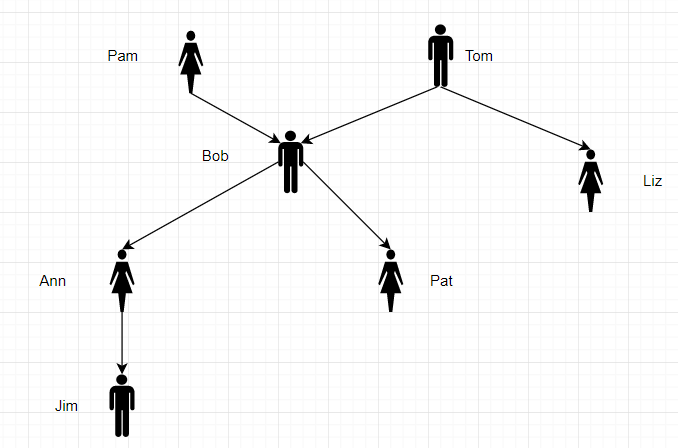


Рисунок 2. Схема родства

Зададим пять вопросов для данного примера.

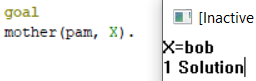


Рисунок 3. Пример 1.1

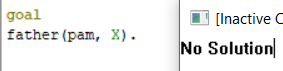


Рисунок 4. Пример 1.2

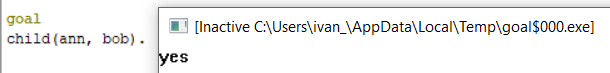


Рисунок 5. Пример 1.3

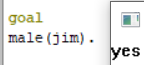


Рисунок 6. Пример 1.4

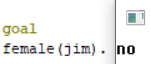


Рисунок 7. Пример 1.5

Перейдем к второму примеру.



Рисунок 8. Код примера 2

Узнаем, любит ли ellen плаванье

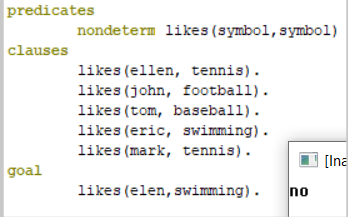


Рисунок 9. Пример 2.1

Узнаем, любит ли ellen тенис.

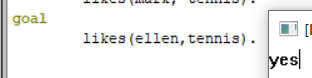


Рисунок 10. Пример 2.2

Найдем тех, кому нравится теннис.

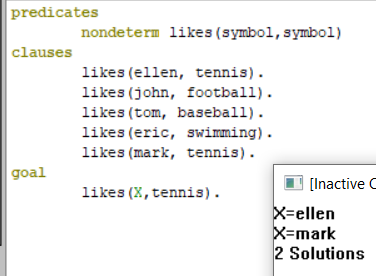


Рисунок 11. Пример 2.3

Найдем тех, кому нравится баскетбол.

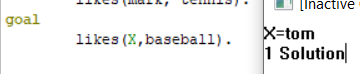


Рисунок 12. Пример 2.4

Найдем тех, кому нравится футбол.

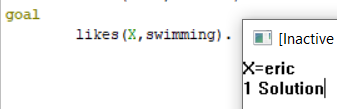


Рисунок 13. Пример 2.5

Перейдем к третьему примеру.

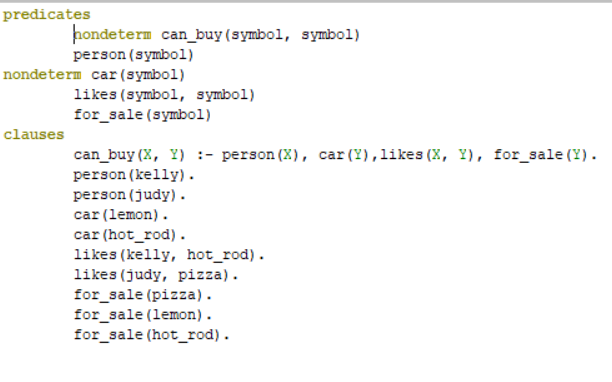


Рисунок 14. Код примера 3

Узнаем, может ли kelly купить автомобиль lemon.



Рисунок 15. Пример 3.1

Узнаем, кто может купить автомобиль lemon.

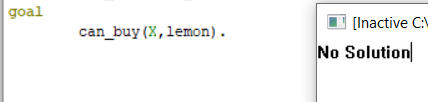


Рисунок 16. Пример 3.2

Узнаем, кому нравится автомобиль pizza.

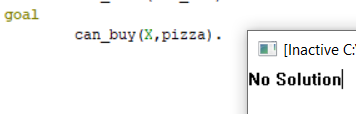


Рисунок 17. Пример 3.3

Найдем всех клиентов

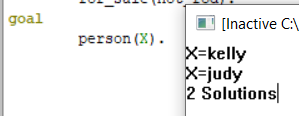


Рисунок 18. Пример 3.4

Наконец, спросим, что может купить judy

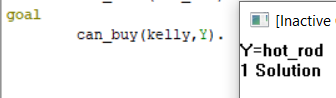


Рисунок 19. Пример 3.5.

Перейдем к примеру 4.

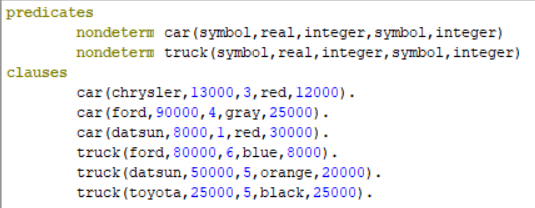


Рисунок 20. Код примера 4

Найдем все машины красного цвета.

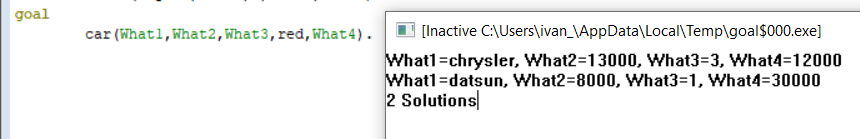


Рисунок 21. Код примера 4.1

Найдем все машины

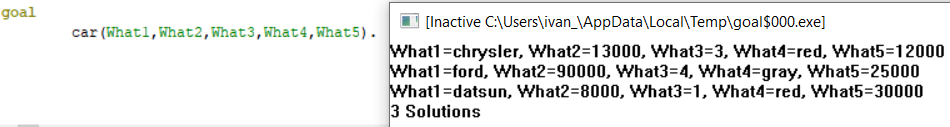


Рисунок 20. Код примера 4.2

Найдем все грузовики

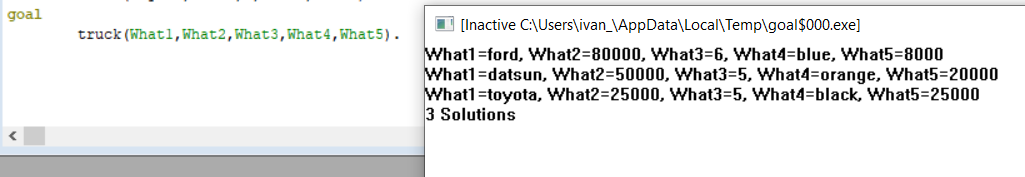


Рисунок 20. Код примера 4.3

Найдем все грузовики, с полем real=50000

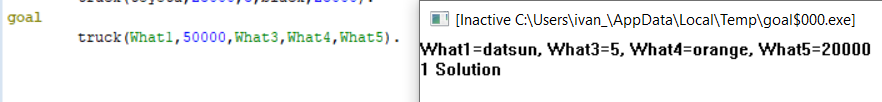


Рисунок 21. Код примера 4.4

Найдем все грузовики, с полем real=50000 и черного цвета.

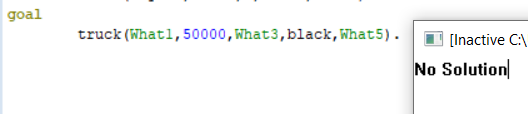


Рисунок 22. Код примера 4.5

**Задание для самостоятельного решения №1**

Реализуем отношения брат и сестра. Для этого создадим новые предикаты: brother и sister.



Рисунок 23. Определение новых предикатов

Затем создадим новых людей и отрегулируем их связи.

Нам понадобятся мать – Мorgana, её детьми будут Ann и Pet, и брат – Richard. Все изменения запишем в классы.

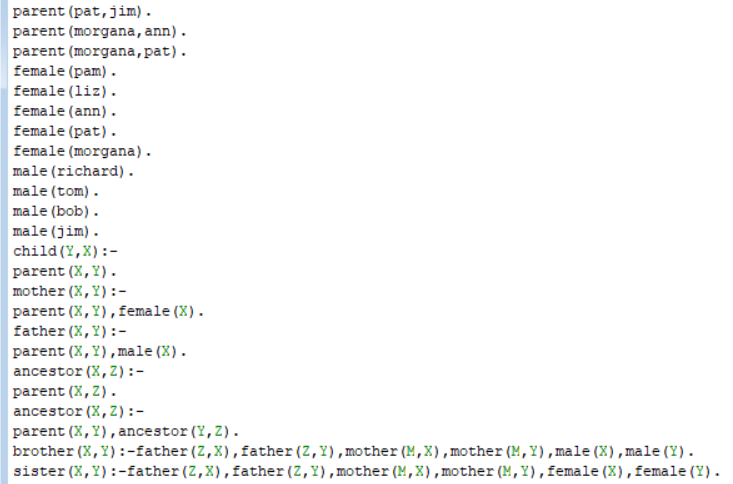


Рисунок 24. Изменения в классах

Продемонстрируем работоспособность введенных отношений.

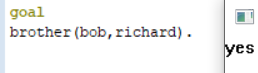


Рисунок 25. Работа отношения brother 1

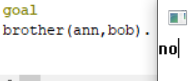


Рисунок 26. Работа отношения brother 2

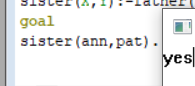


Рисунок 27. Работа отношения sister 1

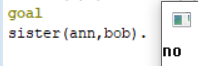


Рисунок 28. Работа отношения sister 2

**Задание для самостоятельного решения №2**

Реализуем программу с описанным в задании функционалом.

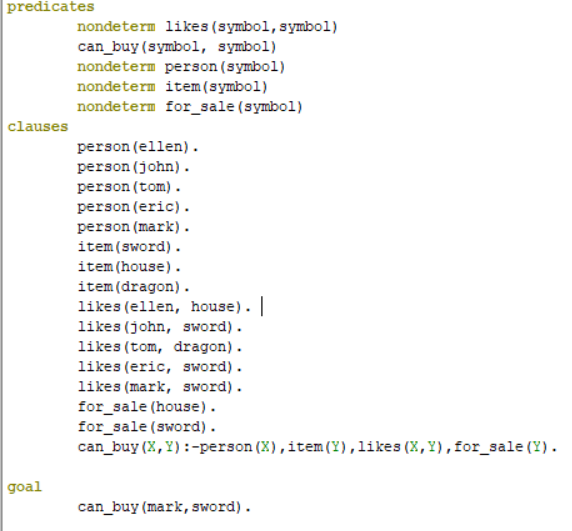


Рисунок 29. Код программы 2

Зададим вопрос программе

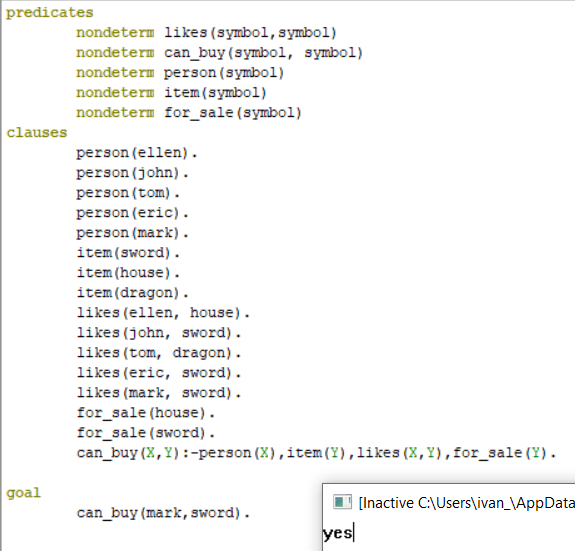


Рисунок 30. Вопрос к программе 1

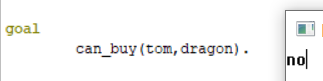


Рисунок 31. Вопрос к программе 1

**Задание для самостоятельного решения №3**

Для выполнения задания была сформирована программа, отражающая события Войны Алой и Белой Роз в Англии.

В программе существуют предикаты:

* Персонаж
* Старый Король
* Мертв
* Родственник Короля
* Битва
* Победил в битве

Также существуют правила:

* Может быть королем– персонаж может быть королем, если старый король мертв и данные персонаж является родственником погибшего короля
* Стал королем – чтобы стать королем, участникам войны роз необходимо было иметь права на трон (иметь возможность быть королем), а также победить в прошедшей битве

На основе этого была создана следующая программа.



Рисунок 32. Код программы «Война роз»

Исходя из вышеописанных правил, королем могут быть все персонажи, кроме Уоррика, так как он не является родственником короля.

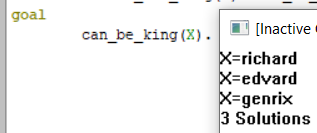


Рисунок 33. Тестирование программы

Стать новым королем при текущих фактах может только Генрих.

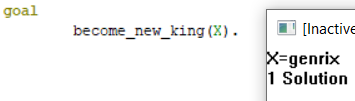


Рисунок 34. Тестирование программы 2

**Выводы**

В результате данной лабораторной работы, был установлен и настроен Visual Prolog v5.2, освоены методы создания простых программ и базовые синтаксические правила этого языка.