Міністерство освіти і науки України

Львівський національний університет імені Івана Франка

Факультет електроніки та комп'ютерних технологій

Кафедра радіоелектронних та комп'ютерних систем

Звіт

про виконання лабораторної роботи №2

Виконав студент

групи Феі- 54

Вегера Северин

Викладач:

доц. Бабич О. й.

Львів — 2018

Короткі теоретичні відомості:

Всі твердження програми на Пролозі діляться на **факти, правила і питання**. Факти відображають поточний стан предметної області, містять конкретну інформацію і є істинними предикатами. **Факти** відповідають простим безумовним висловлюванням.

Розглянемо приклад опису меню в ресторані. Об'єкти предметної області - це страви, які можна з'їсти в ресторані, а одним з можливих видів відносин є класифікація всіх страв на закуски, другі м'ясні чи рибні страви та десерти. Меню являє собою невелику **базу знань (БЗ)**, яка записується у вигляді послідовноності фактів наступним чином:

В структурі програми на мові **Visual prolog** використовуються наступні ключові слова і описи:

**Implement [implement]** і **end implement** Серед усіх ключових слів, обговорюваних тут, це єдині ключові слова, використовувані парно. Visual Prolog розглядає код, поміщений між цими ключовими словами, як код, що належить одному класу. За ключовим словом реалізації обов'язково повинні слідувати ім'я класу.

**open** Це ключове слово використовується для розширення області видимості класу. Воно повинно бути поміщено спочатку коду класу, відразу після ключового слова implement [impliment] (з ім'ям класу і, можливо ім'ям інтерфейсу).

**class predicates [predikeit]** Ця секція містить оголошення предикатів, які визначаються в тексті програми в розділі **clauses**. І знову, оголошення предиката - це ім'я, яке присвоюється предикату, і набір аргументів, кожен з яких повинен відповідати або стандартному (зумовленому), або оголошеному домену. Мова Visual Prolog - типізована, тому предикати необхідно оголошувати. В оголошенні предиката вказується його ім'я, ставиться знак двокрапки, а потім в круглих дужках через кому перераховуються імена доменів (типів даних) аргументів.

Предикати закуска (так само як і ряд інших предикатів) оголошені як недетерміновані. Для позначення недетермінірованних предикатів використовується ключове слово **nondeterm**, що позначає, що предикат може виробляти безліч рішень за допомогою бектрекінга (відкату), (так само як і ряд інших предикатів) оголошені як недетерміновані. Кваліфікатор напрямків введення-виведення **anyflow** каже компілятору, що параметри, призначені предикату можуть бути як пов'язаними, так і вільними, без жодних обмежень (розглянемо дальше, коли будуть два параметри в дужках.

**clauses** Серед усіх розділів, існуючих в тексті програм на Visual Prolog, це єдиний розділ, який близько збігається з традиційними програмами на Пролозі. Він містить конкретні визначення оголошених у розділі клас предикатів предикатів, причому синтаксично їм відповідні.

Предикат **console::init()** ініціалізує консоль.

**goal** (мета) Це предикат - саме той предикат, з якого вся програма починає виконуватися на мові системи Visual Prolog. У традиційному Пролозі, як тільки який-небудь предикат визначений у тексті, Пролог-машині може бути наказано виконання програми, починаючи з цього предиката. У Visual Prolog це не так. Будучи компілятором, він відповідає за генерацію ефективного виконуваного коду написаної програми. Цей код виконується не в той же час, коли компілятор породжує його. Тому компілятору треба знати заздалегідь точно, з якого предиката почнеться виконання програми.

Постановка задачі:

Є база даних, що містить наступні факти:

родич ("Марина", "Ірина").

родич ("Олена", "Анна").

родич ("Ольга", "Марина").

родич ("Ольга", "Тетяна").

родич ("Тетяна", "Катя").

родич ("Анна", "Маша").

Завдання:

а) скласти правило мати і визначити, хто мати Маші.

b) скласти правило бабуся і визначити, хто бабуся Маші.

c) скласти правило внучка і визначити, скільки онучок у Ольги і як їх звуть.

Остаточний код задачі:

% Copyright

implement main

open core

domains

name = string.

class predicates

родич : (name, name) nondeterm anyflow.

мати : (name, name) nondeterm anyflow.

бабуся : (name, name) nondeterm anyflow.

внучка :(name,name) nondeterm anyflow.

clauses

родич ("Марина", "Ірина").

родич ("Олена", "Анна").

родич ("Ольга", "Марина").

родич ("Ольга", "Тетяна").

родич ("Тетяна", "Катя").

родич ("Анна", "Маша").

мати(X,Y) :- родич(X,Y).

бабуся(X,Z) :- мати(X,Y), мати (Y,Z).

внучка(X,Z) :- мати (Y,X), мати(Z,Y).

clauses

run() :-

console::init,

мати (X,"Маша"), stdIO::writef("Мама Маші = %\n ",X),

fail.

run() :-

console::init,

бабуся (X,"Маша"), stdIO::writef("\nБабуся Маші = %\n ",X),

fail.

run() :-

console::init,

внучка (X,"Ольга"),

stdIO::writef("\nОнучка Ольги = %\n ",X),

fail.

run() :-

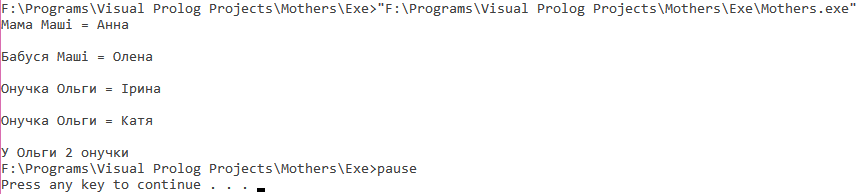
stdIO::writef("\nУ Ольги 2 онучки").

end implement main

goal

console::runUtf8(main::run).

Результати:



Висновок:

На цій лабораторній роботі, я ознайомився із програмними можливостями середовища Visual Prolog. Склав правила мати, бабуся та онучка. Реалізував код, який виконує необхідні запити.