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 №3

"Створення простих проектів"

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.

**Факти**, які приведені всередині програми вводять одночасно **об'єкти** та їх класифікацію (**відношення**). Наприклад, факт **закуска ("салат з яйцем")** показує, що салат з яйцем є закускою.

Синтаксично правильно записаний факт має наступну структуру предиката (рис. 1).

На рис. 1 у списку аргументів перераховуються імена об'єктів (не більше 255), пов'язаних даними відношенням. Аргументи в списку відділяються один від одного комами, в деяких випадках можуть бути відсутні. Якщо аргумент являє собою ім'я конкретного об'єкта (властивості) або число, то він є **константою** Прологу.

ФАКТ <імя вітношення> (<список аргументів>).

Рис. 1. Структура предиката факт

Визначаючи за допомогою фактів відношення між об'єктами, необхідно враховувати порядок, в якому перераховуються їх імена всередині круглих дужок. Вибравши один раз який-небудь порядок, ви повинні всюди слідувати йому і далі. Наприклад, факт **батьком ("Петро", "Іван")** означає, що Петро є батьком Івану, а факт **батьком ("Іван","Петро")** говорить вже зовсім про інше, а саме, що Іван є батьком Петрові. Одне і те ж твердження, записане у вигляді факта, може по-різному інтерпретуватися. Тільки автор програми визначає справжню інтерпретацію імен об'єктів i порядок проходження аргументів, і їм він повинен слідувати в процесі написання всієї программи, відображаючи в коментарях сенс записаних ним висловлювань. **Коментар** - це текст, укладений між символами / \* ... \* / або рядок, що починається зі знака %, наприклад:

**/\*Андрію подобається Ольга\*/**

**подобається("Андрій", "Ольга").**

**/\*Спиця является частиною колеса\*/**

**Частина обєкту("спиця", "колесо").**

Постановка задачі:

Задані відносини-факти:

родичі ("Іван", "Катя").

родичі ("Анна", "Олег").

родичі ("Олег", "Діма").

родичі ("Ігор", "Ольга").

родичі ("Олег", "Віктор").

родичі ("Ігор", "Іван").

чоловік ("Діма").

чоловік ("Іван").

чоловік ("Ігор").

чоловік ("Олег").

чоловік ("Віктор").

жінка ("Катя").

жінка ("Ольга").

жінка ("Анна").

Завдання:

3.1 Складіть нове відношення-правило дідусь (X, Y) і визначити, хто є дідусем Каті. Створіть проект і протестувати приклад як автономну виконувану програму. Доопрацюйте вихідний код прикладу наступним чином:

3.2 Додайте нове правило бабуся і визначте, хто є бабусею Діми;

3.3 Додайте нове правило онук і визначте, хто онук Анни;

3.4 Додайте нове правило брат і визначте, хто брат Діми;

3.5 Додайте нове правило сестра і визначте, хто сестра Івана.

Остаточний код задачі:

implement main

open core

domains

name = string.

class predicates

родич : (name, name) nondeterm anyflow.

чоловік: (name) nondeterm anyflow.

жінка: (name) nondeterm anyflow.

батько : (name, name) nondeterm anyflow.

мати : (name, name) nondeterm anyflow.

дідусь : (name, name) nondeterm anyflow.

бабуся : (name, name) nondeterm anyflow.

онук :(name,name) nondeterm anyflow.

брат :(name,name) nondeterm anyflow.

сестра :(name,name) nondeterm anyflow.

clauses

родич ("Іван", "Катя").

родич ("Анна", "Олег").

родич ("Олег", "Діма").

родич ("Ігор", "Ольга").

родич ("Олег", "Віктор").

родич ("Ігор", "Іван").

чоловік ("Діма").

чоловік ("Іван").

чоловік ("Ігор").

чоловік ("Олег").

чоловік ("Віктор").

жінка ("Катя").

жінка ("Ольга").

жінка ("Анна").

батько(X,Y) :- родич(X,Y), чоловік(X).

мати(X,Y) :- родич(X,Y), жінка(X).

дідусь(X,Z) :- родич(X,Y), родич(Y,Z).

бабуся(X,Z) :- родич(X,Y), родич(Y,Z).

онук(X,Z) :- родич (Y,X), родич(Z,Y).

брат(X,Y) :- родич(Z,X), родич(Z,Y), чоловік(X).

сестра(X,Y) :- родич(Z,X), родич(Z,Y), жінка(X).

clauses

run() :-

console::init,

дідусь (X,"Катя"), stdIO::writef("Дідусь Каті = %\n ",X),

fail.

run() :-

console::init,

бабуся (X,"Діма"), stdIO::writef("\nБабуся Діми = %\n ",X),

fail.

run() :-

console::init,

онук (X,"Анна"),

stdIO::writef("\nОнук Анни = %\n ",X),

fail.

run() :-

console::init,

брат (X,Y), Y = "Діма", X<>Y,

stdIO::writef("\n% брат Діми\n ",X),

fail.

run() :-

console::init,

сестра (X,Y), Y = "Іван", X<>Y,

stdIO::writef("\n% сестра Івана\n ",X),

fail.

run() :-

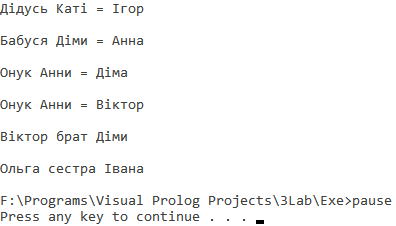
stdIO::writef("").

end implement main

goal

console::runUtf8(main::run).

Результати:



Висновок:

На цій лабораторній роботі, я ознайомився із програмними можливостями середовища Visual Prolog і створив простий проект. Склав правила дідуся, бабусі, онука, брата і сестри. Реалізував код, який виконує необхідні запити.