Міністерство освіти та науки України

Національний технічний університет України “КПІ ім. Ігоря Сікорського”

Факультет інформатики та обчислювальної техніки

Кафедра автоматизованих систем обробки інформації і управління

**ЗВІТ**

про виконання

Лабораторної роботи №10

З дисципліни:

«**Мультипарадигменне програмування**»

**Тема: «НАПИСАННЯ ПРОСТОЇ ПРОГРАМИ НА МОВІ**

**GNU-PROLOG»**

**Варіант 1**

**Виконав:** студент групи ІП-51

Булатов Дмитро Эгорович

**Перевірив**: викл.

Баклан Ігор Всеволодович

Київ 2017

**Мета роботи:**

отримання практичних навичок складання, доопрацювання

та виконання простої програми, складення правил та використання їх в програмі

в системі програмування GNU-PROLOG.

**Завдання:**

1. Проінсталювати на власному комп'ютері систему програмування

GNUPROLOG та систему редагування текстів программ SciTE (Science Text

Editor).

2. Скласти на мові Prolog дерево родинних відношень, використовуючи

предикат **roditel** з двома параметрами: ім'я одного з батьків та ім'я дитини.

Написати на мові Prolog та запустити наступні запити:

•“Хто є і батьками, і має батьків”

•“Хто не має дітей”

**Наприклад:** для схеми родинних зв'язків

програма буде мати вигляд:

**roditel(`иван`,`нина`).**

**roditel(`иван`,`олег`).**

**roditel(`анна`,`нина`).**

**roditel(`анна`,`олег`).**

**roditel(`олег`,`лариса`).**

**roditel(`олег`,`алла`).**

**roditel(`вера`,`алла`).**

**roditel(`вера`,`лариса`).**

**roditel(`лариса`,`наталья`).**

**roditel(`виктор`,`наталья`).**

3. Програму доповнити новими фактами, що дозволяють побудувати правила

для визначення наступних цілей-предикатів:

• батько

• мати

• син

• дочка

• брат

• сестра

• дядько

• тітка

• дід

• баба

• онук

• онучка

• небіж

• небога

• одружені

• теща

• тесть

• свекруха

• свекор

• зять

• невістка

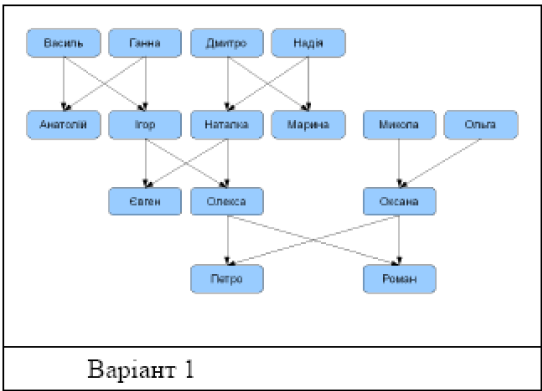
• свояк

• своячка

• дівер

• внучатий небіж

• внучата небога;



**Код програми:**

parent('Василь' ,'Анатолій').

parent('Василь' ,'Ігор').

parent('Ганна' ,'Анатолій').

parent('Ганна' ,'Ігор').

parent('Дмитро' ,'Наталка').

parent('Дмитро' ,'Марина').

parent('Надія' ,'Наталка').

parent('Надія' ,'Марина').

parent('Ігор' ,'Євген').

parent('Ігор' ,'Олекса').

parent('Наталка','Євген').

parent('Наталка','Олекса').

parent('Микола' ,'Оксана').

parent('Ольга' ,'Оксана').

parent('Олекса' ,'Петро').

parent('Олекса' ,'Роман').

parent('Оксана' ,'Петро').

parent('Оксана' ,'Роман').

man('Василь').

man('Анатолій').

man('Ігор').

man('Дмитро').

man('Ігор').

man('Євген').

man('Олекса').

man('Микола').

man('Петро').

man('Роман').

woman('Ганна').

woman('Наталка').

woman('Марина').

woman('Оксана').

woman('Ольга').

woman('Надія').

son(X,Y):-parent(Y,X), man(X).

daughter(X,Y):-parent(Y,X), wonam(X).

father(X,Y):-parent(X,Y), man(X).

mother(X,Y):-parent(X,Y), wonam(X).

brother(X,Y):-parent(Z,X), parent(Z,Y), man(X).

sister(X,Y):-parent(Z,X), parent(Z,Y), woman(X).

sisterorbrother(X,Y):-parent(Z,X), parent(Z,Y).

uncle(X,Y) :- brother(X,Z), parent(Z,Y).

aunt(X,Y):-sister(X,Z), parent(Z,Y).

grandfather(X,Y):-parent(X,Z), parent(Z,Y), man(X).

grandmother(X,Y):-parent(X,Z), parent(Z,Y), woman(X).

grandson(X,Y):-parent(Y,Z), parent(Z,X), man(X).

granddaughter(X,Y):-parent(Y,Z), parent(Z,X), woman(X).

nephew(X,Y):-son(X,Z),sisterorbrother(Z,Y).

niece(X,Y):-daughter(X,Z),sisterorbrother(Z,Y).

married(X,Y):-parent(X,Z),parent(Y,Z).

hismotherinlaw(X,Y):-married(Z,Y),daughter(Z,X),mother(X,Z).

hisfatherinlaw(X,Y):-married(Z,Y),daughter(Z,X),father(X,Z).

hermotherinlaw(X,Y):-married(Z,Y),son(Z,X),mother(X,Z).

herfatherinlaw(X,Y):-son(Z,X),married(Z,Y),father(X,Z).

soninlaw(X,Y):-married(X,Z),father(Z,Y).

daughterinlaw(X,Y):-married(X,Z),son(Z,Y).

brotherinlaw(X,Y):-married(X,Z),sister(Z,D),married(D,Y).

sisterinlaw(X,Y):-sister(X,Z),married(Z,Y),sister(Z,X).

diver(X,Y):-married(Y,Z),brother(X,Z).

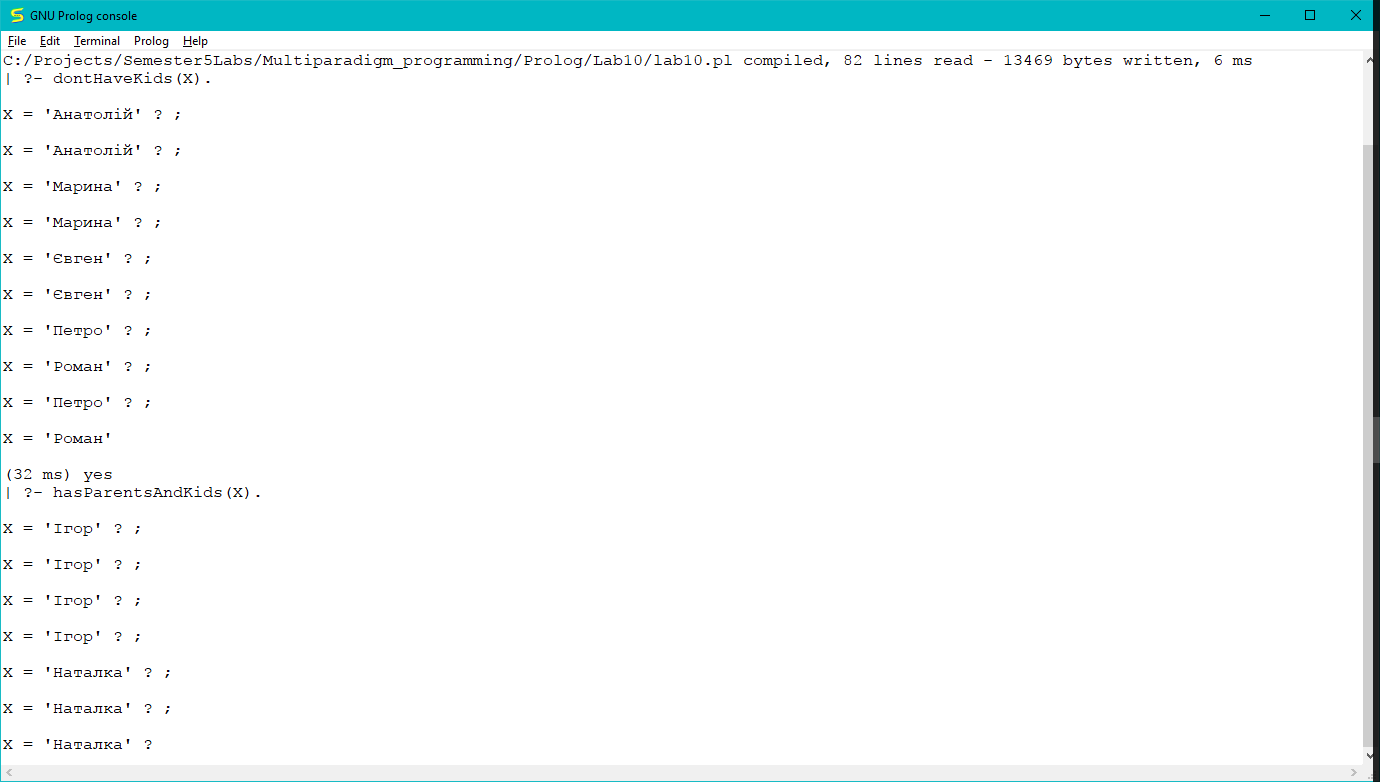
greatnephew(X,Y):-grandson(X,Z),sisterorbrother(Z,Y).

greatniece(X,Y):-granddaughter(X,Z),sisterorbrother(Z,Y).

hasParentsAndKids(X):-parent(X,Y), parent(Z,X).

dontHaveKids(Y):-parent(X,Y), \+ parent(Y,Z).

**Скріншот програми:**



**Висновок:**

В цій лабораторній роботі я опанував використання відношеннь у мові Prolog, та навчився виконувати запити до відношень.