**Ministerul Educaţiei al Republicii Moldova**

**Universitatea Tehnică a Moldovei**

**Facultatea Calculatoare Informatică şi Microelectronică**

**Catedra Automatică şi Tehnologii Informaţionale**

**RAPORT**

Lucrare de laborator nr 1

Disciplina: Programarea Funcțională și Inteligența Artificială

Tema: Introducere în limbajul „Prolog”

**A efectuat:**

Vovc Artemie st. TI-133

**A verificat:**

Lazu Victoria lect. super.

**Chișinău 2016**

Cuprins

[1 Scopul lucrării 3](#_Toc462244214)

[2 Sarcina 4](#_Toc462244215)

[3 Noțiuni teoretice 5](#_Toc462244216)

[4 Realizarea 6](#_Toc462244217)

[Concluzia 8](#_Toc462244218)

[Bibliografia 9](#_Toc462244219)

[Anexe A 10](#_Toc462244220)

# 1 Scopul lucrării

Însușirea principiilor fundamentale de programare a limbajului Prolog.

# 2 Sarcina

1. Se va citi breviarul teoretic. Se atrage atenția asupra faptului că toate cunoștințele din această lucrare vor fi necesare și la efectuarea celorlalte lucrări.
2. Se vor studia exemplele propuse, încercînd găsirea altor posibilități de soluționare a acestora. Se vor utiliza și alte scopuri (interogări) pentru a testa difinițiile predicatelor introduse.
3. Se ca elabora un arbore genialogic și o bază de cunoștințe Prolog care ar descrie relațiile existente în familia dumneavoastră proprie care ar perminte cercetare aacestor relații prin utilizarea scopurilor externe. Arborele genealogic elaborat trebuie să conțină cel puțin trei niveluri. Pentru cercetarea relațiilor existente în familie se vor utiliza nu mai puțin de șase scopuri.
4. Se va prezenta darea de seamă.

# 3 Noțiuni teoretice

Prolog este un limbaj logic, descriptiv care permite specificarea spaţiului problemei şi a soluţiei acesteia, operând în termeni de fapte cunoscute despre obiectele universului problemei şi ale relaţiilor existente între aceste obiecte. Execuţia unui program Prolog constă în deducerea implicaţiilor dintre aceste fapte şi relaţii, programul definind astfel o mulţime de consecinţe ce reprezintă înţelesul sau semnificaţia declarative a programului.

Un program Prolog conţine următoarele entităţi:

* *fapte* despre obiecte şi *relaţiile* existente între aceste obiecte;
* *reguli* despre obiecte şi relaţiile dintre acestea care permit deducerea (inferarea) unor fapte noi în baza celor cunoscute;
* întrebări, numite şi *scopuri*, despre obiecte şi relaţiile acestora, la care programul răspunde în baza faptelor şi regulilor existente.

*Faptele* sunt predicate de ordinul întâi de aritate *n* considerate adevărate (reprezintă cea mai simplă forma de predicat din Prolog). Ele stabilesc relaţiile dintre obiectele universului problemei. Numărul de argumente ale faptelor este reprezentat prin aritatea corespunzătoare a predicatelor.

Interpretarea particulară a predicatului şi a argumentelor acestuia depinde de programator. Ordinea argumentelor este importantă şi odată fixată trebuie păstrată pentru orice utilizare ulterioară a faptului cu aceeaşi semnificaţie. Mulţimea faptelor şi regulilor unui program Prolog formează *baza de cunoştinţe* Prolog.

O *regulă* Prolog exprimă un fapt care depinde de alte fapte şi are forma:

**S :- S1, S2, …Sn.**

Fiecare **Si, i =1,n** şi **S** au forma faptelor Prolog, deci sunt predicate cu argumente constante, variabile sau structuri. Faptul **S** care defineşte regula se numeşte *antet de regulă*, iar **S1, S2, …Sn** formează *corpul regulii* şi reprezintă conjuncţia (unificarea) scopurilor care trebuie satisfăcute pentru satisfacerea ulterioară a antetului regulii.

# 4 Realizarea

Prima sarcină e să facem arborele genealogin vezi mai jos in figura 1 arborele genealogic.



Figura 1 – Arborele genealogic

Baza de cunoștințe pentru figura 1 de mai sus.

femeie(ana).

femeie(maria).

femeie(lilia).

femeie(valea).

femeie(tatiana).

femeie(felicia).

femeie(alina).

femeie(anghelina).

femeie(cezarina).

barbat(ilie).

barbat(sergiu).

barbat(valeriu).

barbat(nicolai).

barbat(mihail).

barbat(vergiliu).

barbat(artiom).

barbat(stanis).

parinti(ana,ilie,valeriu).

parinti(ana,ilie,tatiana).

parinti(ana,ilie,valea).

parinti(maria,sergiu,nicolai).

parinti(maria,sergiu,felicia).

parinti(lilia,valeriu,alina).

parinti(lilia,valeriu,vergiliu).

parinti(valea,mihail,anghelina).

parinti(tatiana,nicolai,stanis).

parinti(tatiana,nicolai,artiom).

Și formez interogările:

mama(X,Y):- parinti(X,\_,Y),femeie(X).

tata(X,Y):- parinti(\_,X,Y),barbat(X).

frati(X,Y):- parinti(M,T,X),parinti(M,T,Y),X\=Y.

sora(X,Y):- frati(X,Y),femeie(X).

frate(X,Y):- frati(X,Y),barbat(X).

verisori(X,Y):- parinti(M1,T1,X),parinti(M2,T2,Y),

(

frati(M1,M2);

frati(M1,T2);

frati(T1,M2);

frati(T1,T2)

),X\=Y,not(frati(X,Y)).

verisori(\_,\_).

bunici(X,Y):- parinti(X,Y,Z),(parinti(Z,\_,\_);parinti(\_,Z,\_)).

bunica(X):- bunici(X,\_).

bunel(X):-bunici(\_,X).

nepot(X):-parinti(M,T,X),(parinti(\_,\_,M);parinti(\_,\_,T)).

casatoriti(X,Y):- parinti(X,Y,\_).

unchi(X,Y):-parinti(M,T,Y),

((frate(X,M);frate(X,T));

( parinti(M1,T1,M),parinti(M1,T1,C1),

casatoriti(C1,X),not(parinti(\_,X,Y)));

( parinti(M2,T2,T),parinti(M2,T2,C2),

casatoriti(C2,X),not(parinti(\_,X,Y)))

).

matusa(X,Y):-parinti(M,T,Y),

((sora(X,M);sora(X,T));

( parinti(M1,T1,M),parinti(M1,T1,C1),

casatoriti(X,C1),not(parinti(X,\_,Y)));

( parinti(M2,T2,T),parinti(M2,T2,C2),

casatoriti(X,C2),not(parinti(X,\_,Y)))

).

# Concluzia

Lucrarea dată a avut ca scop să ne facă cunoscuți cu programarea logică la general și programarea în prolog esențial. La lucrare am format o baza de cunoștințe în care formal se constituie arborele genealogic al familiei personale. După ce a fost formată baza s-a trecut la formarea regulilor.

# Bibliografia

1. **Prolog la general :** http://biblioteca.regielive.ro/referate/limbaje-de-programare/prolog-limbaj-de-programare-logica-114586.html
2. **Studierea SWI-Prolog** **:** http://www.swi-prolog.org/pldoc/index.html

## Anexe A

**Baza de cunoștințe**

femeie(ana).

femeie(maria).

femeie(lilia).

femeie(valea).

femeie(tatiana).

femeie(felicia).

femeie(alina).

femeie(anghelina).

femeie(cezarina).

barbat(ilie).

barbat(sergiu).

barbat(valeriu).

barbat(nicolai).

barbat(mihail).

barbat(vergiliu).

barbat(artiom).

barbat(stanis).

parinti(ana,ilie,valeriu).

parinti(ana,ilie,tatiana).

parinti(ana,ilie,valea).

parinti(maria,sergiu,nicolai).

parinti(maria,sergiu,felicia).

parinti(lilia,valeriu,alina).

parinti(lilia,valeriu,vergiliu).

parinti(valea,mihail,anghelina).

parinti(tatiana,nicolai,stanis).

parinti(tatiana,nicolai,artiom).

mama(X,Y):- parinti(X,\_,Y),femeie(X).

tata(X,Y):- parinti(\_,X,Y),barbat(X).

frati(X,Y):- parinti(M,T,X),parinti(M,T,Y),X\=Y.

sora(X,Y):- frati(X,Y),femeie(X).

frate(X,Y):- frati(X,Y),barbat(X).

verisori(X,Y):- parinti(M1,T1,X),parinti(M2,T2,Y),

(

frati(M1,M2);

frati(M1,T2);

frati(T1,M2);

frati(T1,T2)

),X\=Y,not(frati(X,Y)).

verisori(\_,\_).

bunici(X,Y):- parinti(X,Y,Z),(parinti(Z,\_,\_);parinti(\_,Z,\_)).

bunica(X):- bunici(X,\_).

bunel(X):-bunici(\_,X).

nepot(X):-parinti(M,T,X),(parinti(\_,\_,M);parinti(\_,\_,T)).

casatoriti(X,Y):- parinti(X,Y,\_).

unchi(X,Y):-parinti(M,T,Y),

((frate(X,M);frate(X,T));

( parinti(M1,T1,M),parinti(M1,T1,C1),

casatoriti(C1,X),not(parinti(\_,X,Y)));

( parinti(M2,T2,T),parinti(M2,T2,C2),

casatoriti(C2,X),not(parinti(\_,X,Y)))

).

matusa(X,Y):-parinti(M,T,Y),

((sora(X,M);sora(X,T));

( parinti(M1,T1,M),parinti(M1,T1,C1),

casatoriti(X,C1),not(parinti(X,\_,Y)));

( parinti(M2,T2,T),parinti(M2,T2,C2),

casatoriti(X,C2),not(parinti(X,\_,Y)))

).