МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра вычислительных технологий**

**ОТЧЁТ**

**ПО ИНДИВИДУАЛЬНОЙ РАБОТЕ №1**

**ДИСЦИПЛИНА : «КОНСТРУИРОВАНИЕ АЛГОРИМОВ И СТРУКТУР ДАННЫХ»**

Работу выполнил\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_А.С. Ларионов

Направление подготовки 02.03.02 Фундаментальная информатика и

информационные технологии

Направленность (профиль) Компьютерные науки

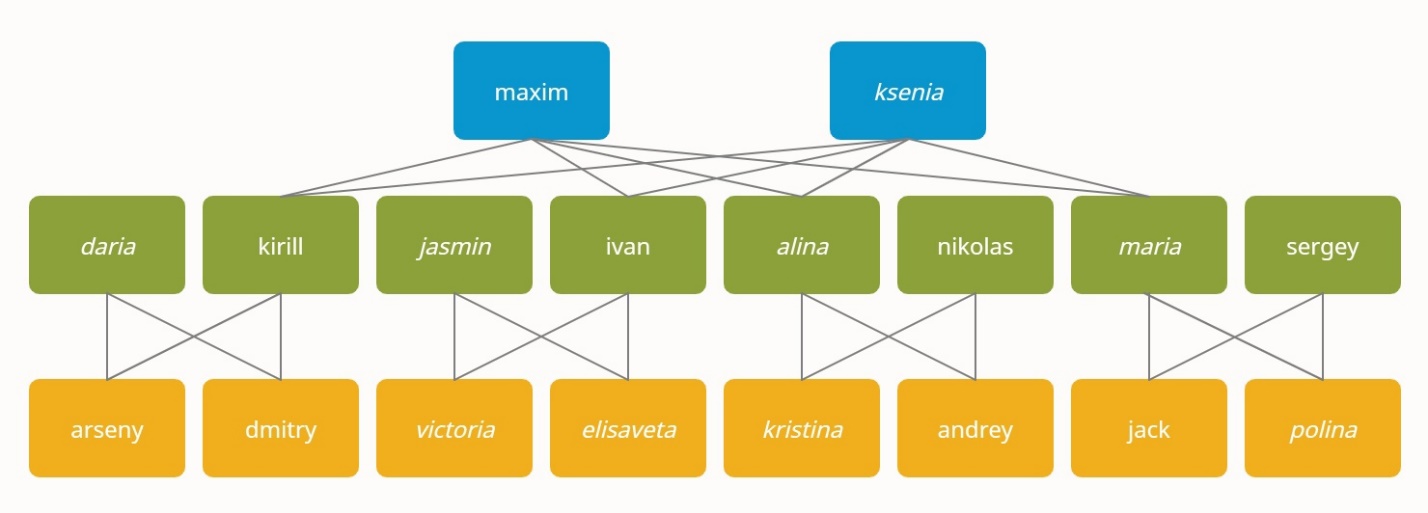
Преподаватель

преподаватель\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_А.А. Климец

Краснодар

2021

Задание 1.

Построил дерево семьи в графическом редакторе.  Рис. 1 - Графически представленное дерево семьи.

Составил базу фактов(мужчин и женщин) Prolog.

Написал два предиката men и women для вывода всех мужчин и женщин соответственно.

men(X):-man(X), write(X), write(", "), fail.

women(X):-woman(X), write(X), write(", "), fail.

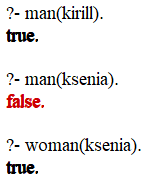


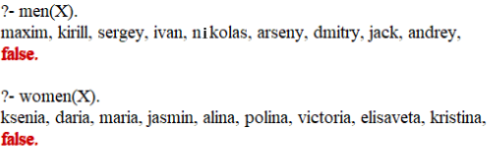
Рис. 2 – Запросы на проверку пола некоторых членов семьи. 

Рис. 3 – Запросы на вывод всех мужчин и всех женщин соответственно.

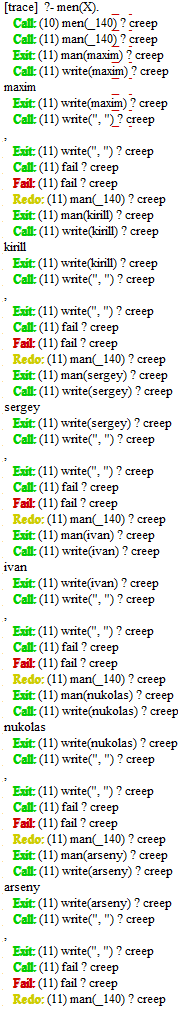
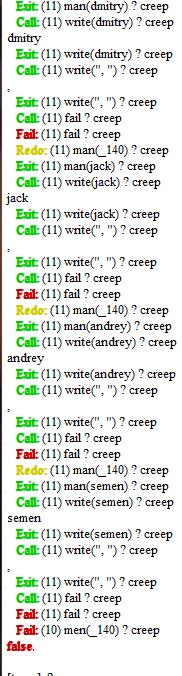
 

Рис. 4 – Трассировка предиката men(X).

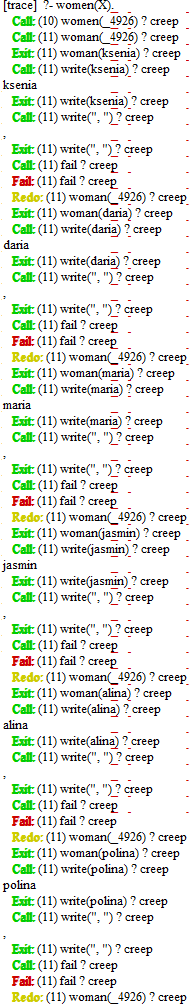
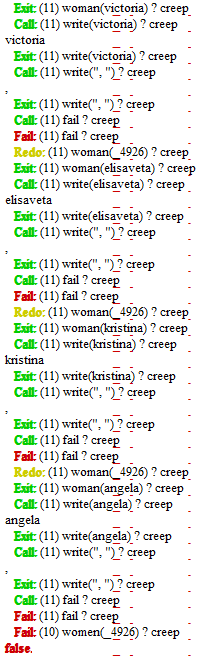
 

Рис. 5 – Трассировка предиката women(X).

Задание 2.

Построил базу данных отношений “являться родителем” в Prolog.

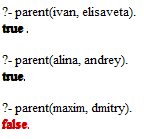


Рис. 6 – Примеры вызовов предиката parent(X,Y), X и Y – имена людей в базе.

Построл предикат children(X), который выводит всех детей X.

children(X):-parent(X,Y), write(Y), write(", "), fail.

Задание 3.

Построил предикаты mother(X,Y) и mother(X), первый из которых проверяет является ли X матерью Y, а последний находит мать X.

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

mother(X):-parent(Y,X), woman(Y), write(Y), fail.

Задание 4.

Построил предикаты daughter(X,Y) и daughter(X), первый из которых проверяет является ли X дочерью Y, а последний находит дочь X.

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

daughter(X):-parent(X,Y), woman(Y), write(Y), write(", "), fail.

Задание 5.

Построил предикат brother(X,Y), который проверяет является ли X братом Y.

brother(X,Y):- not(X=Y), parent(Z,X), parent(Z,Y), man(X).

Построил предикат brothers(X), который выводит всех братьев X.

brothers(X):-parent(Y,X), parent(Y,Z), not(X=Z), man(Z), man(Y), write(Z), write(", "), fail.

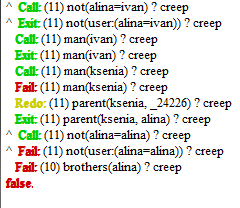
 

Рис. 6 – Трассировка предиката brothers(X), X = alina.

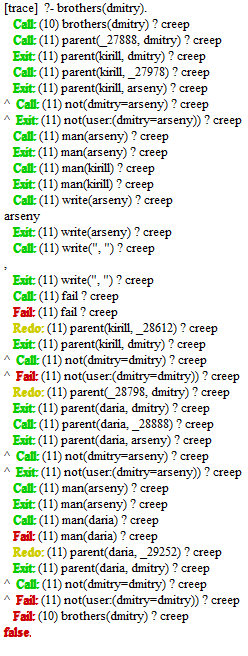


Рис. 7 – Трассировка предиката brothers(X), X = dmitry.

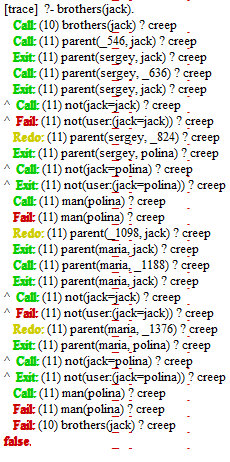


Рис. 8 – Трассировка предиката brothers(X), X = jack.

Задание 6.

Построил предикат wife(X,Y), который проверяет является ли X женой Y.

wife(X,Y):-woman(X), man(Y), parent(X,Z), parent(Y,Z).

Построил предикат wife(X), который выводит жену Y.

wife(X):-man(X), parent(X,Z), parent(Y,Z), woman(Y), write(Y), !.

Задание 7.

Построил предикат b\_s(X,Y), который проверяет являются ли X и Y братьями, сёстрами или братом и сестрой.

b\_s(X,Y):-not(X=Y), parent(Z,Y), parent(Z,X), !.

Построил предикат b\_s(X), который выводит всех братьев и сестёр X.

b\_s(X):-parent(Z,X), parent(Z,Y), man(Z), not(X=Y), write(Y), write(", "), fail.

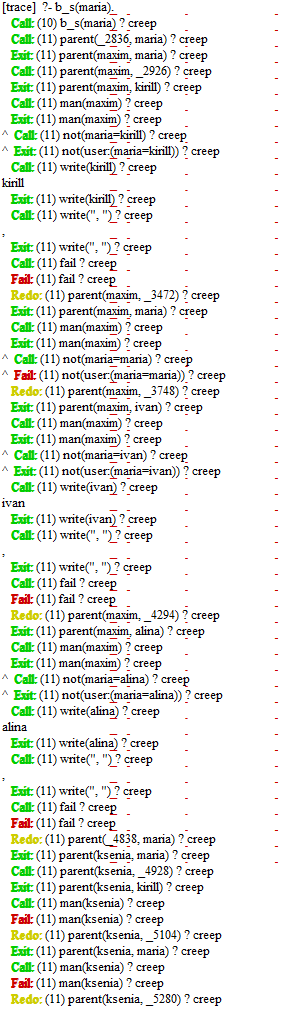
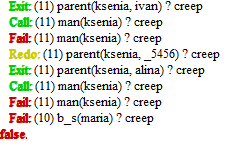
 

Рис. 9 – трассировка предиката b\_s(X), X = maria.

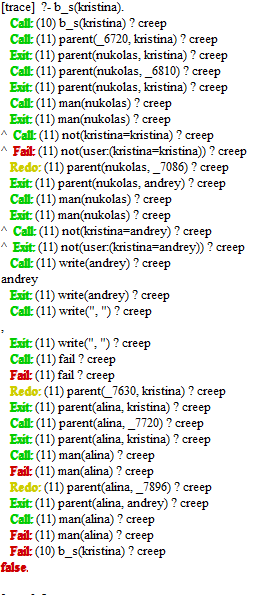


Рис. 10 – трассировка предиката b\_s(X), X = kristina.

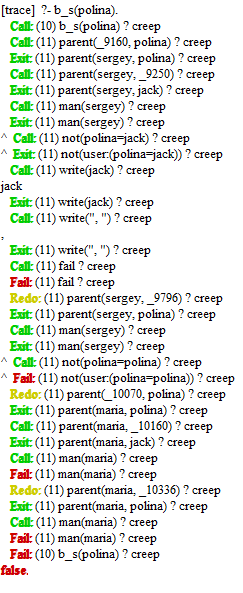


Рис. 11 – трассировка предиката b\_s(X), X = polina.

Задание 8.

Построил предикат grand\_pa(X,Y), который прверяет является ли X дедушкой Y.

grand\_pa(X,Y):-man(X), parent(X,Z), parent(Z,Y).

Дополнил базу фактов таким образом, чтобы у каждого ребёнка было 2 дедушки и 2 бабушки.

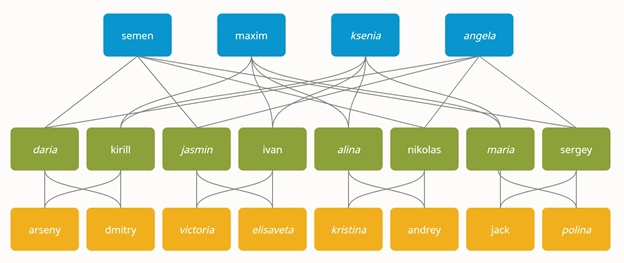


Рис. 12 – Дополненное дерево семьи.

Построил предикат grand\_pas(X), который выводит всех дедушек X.

grand\_pas(X):-parent(Y,X), parent(Z,Y), man(Z), write(Z), write(", "), fail.



Рис. 13 – Трассировка предиката grand\_pas(X), X = andrey.

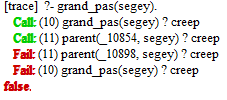


Рис. 14 – Трассировка предиката grand\_pas(X), X = sergey.

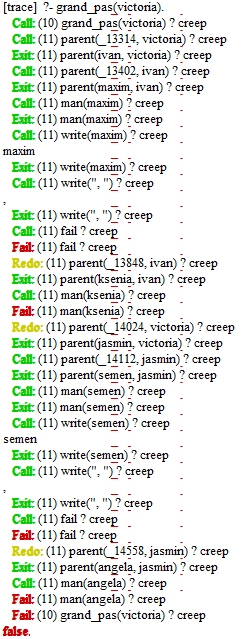


Рис. 15 – Трассировка предиката grand\_pas(X), X = victoria.

Задание 9.

Построил предикат garand\_ma(X,Y), который проверяет является ли X бабушкой Y.

grand\_ma(X,Y):-woman(X), parent(X,Z), parent(Z,Y).

Построил предикат garand\_ma(X), который выводит всех бабушек X.

grand\_mas(X):-parent(Y,X), parent(Z,Y), woman(Z), write(Z), write(", "), fail.

Задание 10.

Построил предикат grand\_pa\_and\_son(X,Y), который проверяет являются ли X и Y дедушкой и внуком или внуком и дедушкой.

grand\_pa\_and\_son(X,Y):-man(X), man(Y), (grand\_pa(X,Y); grand\_pa(Y,X)).

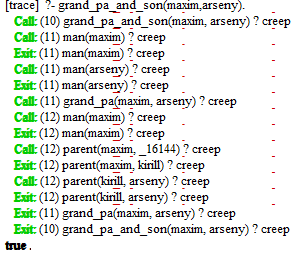


Рис. 16 – трассировка предиката grand\_pa\_and\_son(X,Y), X = maxim, Y = arseny.

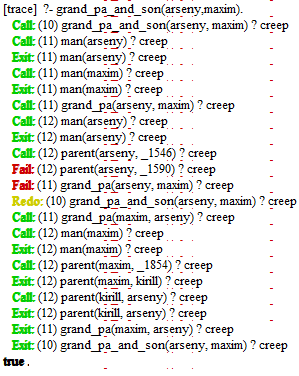


Рис. 17 – трассировка предиката grand\_pa\_and\_son(X,Y), X =arseny, Y = maxim.

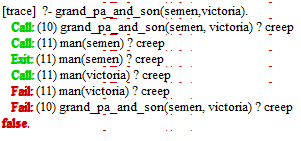


Рис. 18 – трассировка предиката grand\_pa\_and\_son(X,Y), X =semen, Y = victoria.

Задание 11.

Построил предикат grand\_ma\_and\_da(X,Y), который проверяет являются ли X и Y бабушкой и внучкой или внучкой и бабушкой.

grand\_ma\_and\_da(X,Y):-grand\_ma(X,Y); grand\_ma(X,Y).

Задание 12.

Построил предикат uncle(X,Y), который проверяет является ли X дядей Y.

uncle(X,Y):-man(X), parent(Z,X), parent(Z,X1), parent(X1,Y), not(X=X1).

Построил предикат all\_uncles(X), который выводит всех дядей X.

all\_uncles(X):-parent(Y,X), parent(Z,Y), parent(Z,Y1), man(Z), not(Y=Y1), man(Y1), write(Y1), write(", "), fail.

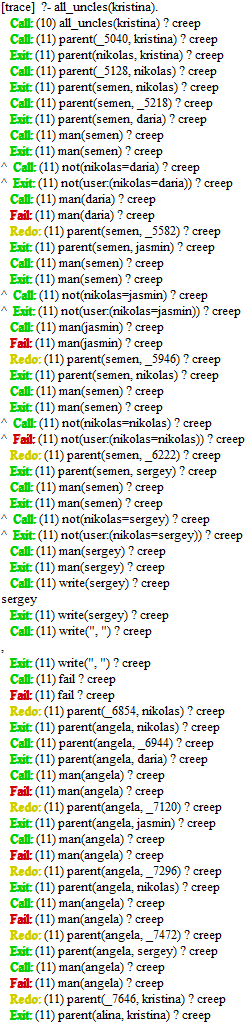
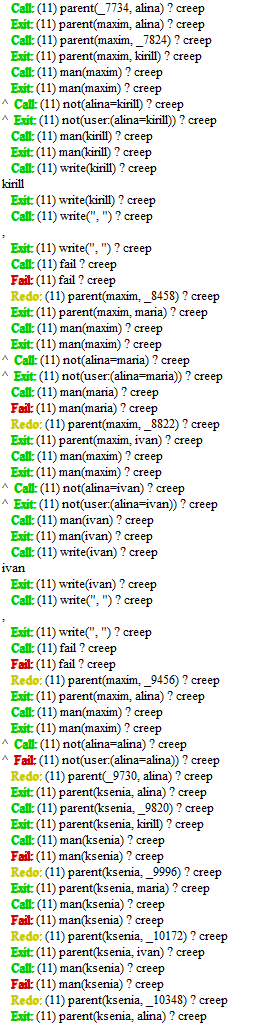
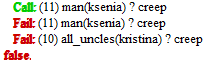
  

Рис. 19 – Трассировка предиката all\_uncles(X), X = kristina.

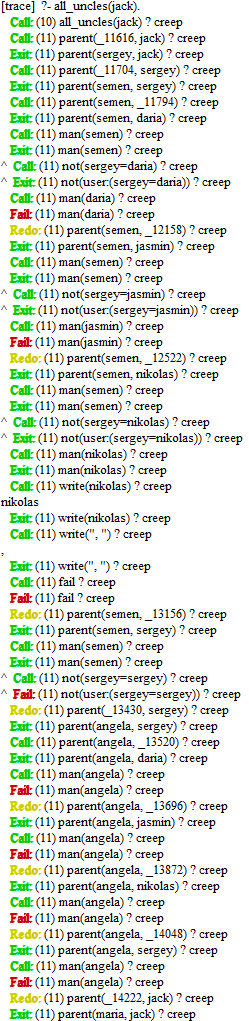
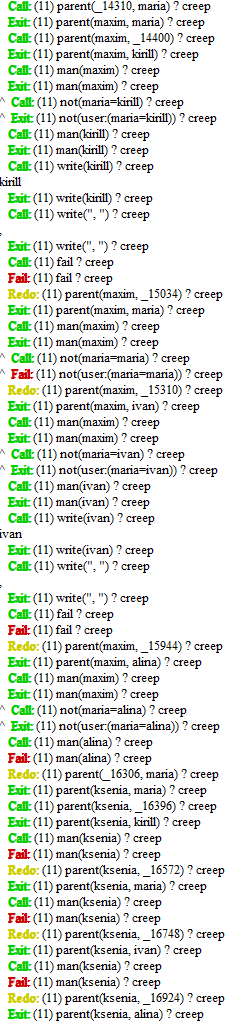
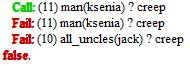
  

Рис. 20 – Трассировка предиката all\_uncles(X), X = jack.

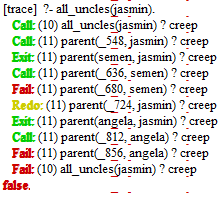


Рис. 21 – Трассировка предиката all\_uncles(X), X = jasmin.

Задание 13.

Построил предикаты nephew(X,Y) и all\_nephews(X), первый из которых проверяет является ли X, племянником Y, а второй выводит всех племянников X.

nephew(X,Y):-man(X), parent(Y1,X), parent(Z,Y1), parent(Z,Y), not(Y=Y1), !.

all\_nephews(X):-parent(Z,X), man(Z), parent(Z,X1), not(X=X1), parent(X1,Y), man(Y), write(Y), write(", "), fail.