**Московский авиационный институт**

**(Национальный исследовательский университет)**

Факультет прикладной математики и физики

Кафедра вычислительной математики и программирования

**Курсовая работа**

**по курсу «Логическое программирование»**

Студент: Архипов М. Д.

Группа: М8О-407Б-18

Преподаватель: Левинская М. А.

Сошников Д.В.

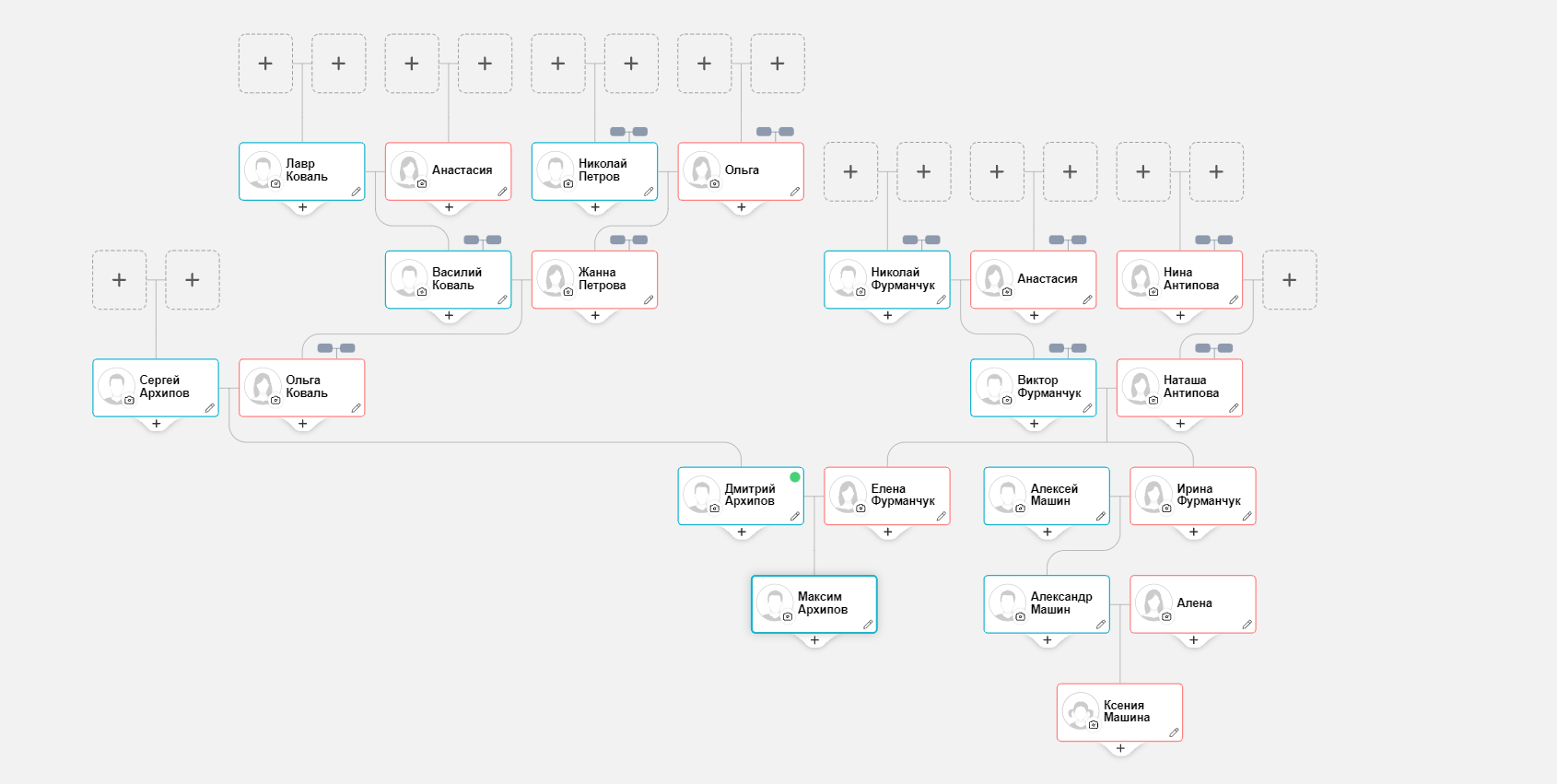
Оценка:

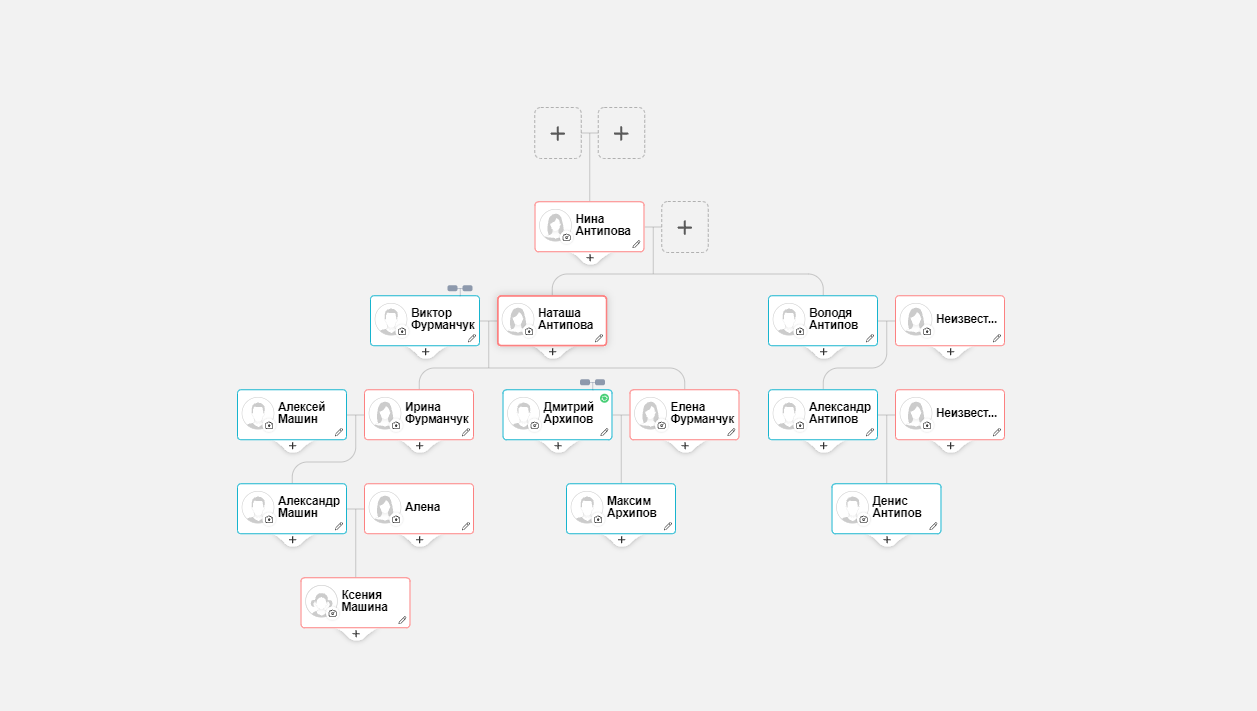
Москва 2021

**1) Формулировка задания:**

1. Создать родословное дерево своего рода на несколько поколений (3-4) назад в стандартном формате GEDCOM с использованием сервиса MyHeritage.com
2. Преобразовать файл в формате GEDCOM в набор утверждений на языке Prolog, используя следующее представление: father, mother
3. Реализовать предикат проверки/поиска (вариант 1) шурин.
4. Реализовать программу на языке Prolog, которая позволит определять степень родства двух произвольных индивидуумов в дереве
5. [На оценки хорошо и отлично] Реализовать естественно-языковый интерфейс к системе, позволяющий задавать вопросы относительно степеней родства, и получать осмысленные ответы.

**2) Алгоритм работы программы:**

**1.** В качестве родословного дерева было выбрано дерево моей семьи

****

**2.** Преобразование файла в формате GEDCOM в набор утверждений на языке Prolog с помощью python

**3**.  Реализация предиката проверки/поиска шурина.

**4.** Реализовать программу на языке Prolog, которая позволит определять степень родства двух произвольных индивидуумов в дереве.

**3) Листинг программы и протоколы:**

**1. Преобразование файла в формате GEDCOM в набор утверждений на языке Prolog**

class Family:

def \_\_init\_\_(self, idx):

self.\_childs = []

self.\_id = idx

self.\_husb\_id = None

self.\_wife\_id = None

def add\_husband(self, idx):

self.\_husb\_id = idx

def add\_wife(self, idx):

self.\_wife\_id = idx

def add\_child(self, idx):

self.\_childs.append(idx)

@property

def childs(self):

return self.\_childs

@property

def husb\_id(self):

return self.\_husb\_id

@property

def wife\_id(self):

return self.\_wife\_id

class Person:

def \_\_init\_\_(self, idx, name, surname, sex):

self.\_id = idx

self.\_name = name

self.\_surname = surname

self.\_sex = sex

def get\_full\_name(self):

return self.\_name + ' ' + self.\_surname

def \_\_str\_\_(self):

return self.\_name

@property

def sex(self):

return self.\_sex

def parse(file\_name):

persons = {}

families = {}

person\_id = None

person\_name = None

person\_surname = None

person\_sex = None

fam\_idx = None

with open(file\_name, encoding='utf8') as f:

for line in f:

if 'INDI' in line:

person\_id = line.split()[1]

elif 'NAME' in line:

arr = line.split()

person\_name = arr[2]

elif 'SURN' in line:

person\_surname = line.split()[2]

elif 'SEX' in line:

person\_sex = line.split()[2]

persons[person\_id] = (Person(

person\_id,

person\_name,

person\_surname,

person\_sex

))

elif 'FAM\n' in line:

families[line.split()[1]] = Family(line.split()[1])

fam\_idx = line.split()[1]

elif 'HUSB' in line:

families[fam\_idx].add\_husband(line.split()[2])

elif 'WIFE' in line:

families[fam\_idx].add\_wife(line.split()[2])

elif 'CHIL' in line:

families[fam\_idx].add\_child(line.split()[2])

return persons, families

def get\_man\_chil(families, persons):

for key in families:

for child in families[key].childs:

if families[key].husb\_id:

print('father(\"{}\", \"{}\").'.format(

persons[families[key].husb\_id].get\_full\_name(),

persons[child].get\_full\_name()))

if families[key].wife\_id:

print('mother(\"{}\", \"{}\").'.format(

persons[families[key].wife\_id].get\_full\_name(),

persons[child].get\_full\_name()))

def main():

persons, families = parse('mytree.ged')

get\_man\_chil(families, persons)

if \_\_name\_\_ == '\_\_main\_\_':

main()

**Результат:**

father("Дмитрий Архипов", "Максим Архипов").

mother("Елена Фурманчук", "Максим Архипов").

father("Виктор Фурманчук", "Ирина Фурманчук").

mother("Наташа Антипова", "Ирина Фурманчук").

father("Виктор Фурманчук", "Елена Фурманчук").

mother("Наташа Антипова", "Елена Фурманчук").

father("Сергей Архипов", "Дмитрий Архипов").

mother("Ольга Коваль", "Дмитрий Архипов").

father("Алексей Машин", "Александр Машин").

mother("Ирина Фурманчук", "Александр Машин").

father("Александр Машин", "Ксения Машина").

mother("Алена Машина", "Ксения Машина").

father("Василий Коваль", "Ольга Коваль").

mother("Жанна Петрова", "Ольга Коваль").

father("Василий Коваль", "Александр Коваль").

mother("Жанна Петрова", "Александр Коваль").

father("Василий Коваль", "Анатолий Коваль").

mother("Жанна Петрова", "Анатолий Коваль").

father("Нина Антипова", "Наташа Антипова").

father("Нина Антипова", "Володя Антипов").

father("Анатолий Коваль", "Настя Коваль").

mother("Аня Коваль", "Настя Коваль").

father("Анатолий Коваль", "Елена Коваль").

mother("Аня Коваль", "Елена Коваль").

father("Николай Фурманчук", "Виктор Фурманчук").

mother("Анастасия Фурманчук", "Виктор Фурманчук").

father("Николай Фурманчук", "Сергей Фурманчук").

mother("Анастасия Фурманчук", "Сергей Фурманчук").

father("Сергей Фурманчук", "Таня Хофбауэр").

mother("Ольга Хофбауэр", "Таня Хофбауэр").

father("Сергей Фурманчук", "Екатерина Хофер").

mother("Ольга Хофбауэр", "Екатерина Хофер").

father("Михаэль Хофбауэр", "Аня Хофбауэр").

mother("Таня Хофбауэр", "Аня Хофбауэр").

father("Михаэль Хофбауэр", "Даня Хофер").

mother("Таня Хофбауэр", "Даня Хофер").

father("Володя Антипов", "Александр Антипов").

mother("// Антипов", "Александр Антипов").

father("Александр Антипов", "Денис Антипов").

mother("// Антипов", "Денис Антипов").

father("Лавр Коваль", "Василий Коваль").

mother("Анастасия Хофер", "Василий Коваль").

father("Николай Петров", "Жанна Петрова").

mother("Ольга Коваль", "Жанна Петрова").

father("Николай Петров", "Юдифь Петрова").

mother("Ольга Коваль", "Юдифь Петрова").

father("// Петрова", "Лена Петрова").

mother("Юдифь Петрова", "Лена Петрова").

father("// Петрова", "Ольга Петрова").

mother("Лена Петрова", "Ольга Петрова").

father("// Петрова", "Иван Петрова").

mother("Ольга Петрова", "Иван Петрова").

**2. Реализация предиката проверки/поиска шурина:**

% Поиск Шурина (брата жены)

%shurin

shurin(Husb, Bro) :- wife(Husb, Wife), brother(Wife, Bro).

% wife

wife(X, T) :- father(X, L), mother(T, L), !.

brother(X, Z) :- (father(Y, X), father(Y, Z); mother(T, X), mother(T, Z)), male(

), X \= Z.

Тест:

14 ?- shurin("Виктор Фурманчук","Володя Антипов").

true ;

**3. Родства двух произвольных индивидуумов в дереве**

what\_relation(Y, X, "brother") :-

    (father(T,X), father(T,Y); mother(T,X), mother(T,Y)),

    male(X),

    Y \= X.

what\_relation(Y, X, "sister") :-

    (father(T,X), father(T,Y); mother(T,X), mother(T,Y)),

    female(X),

    Y \= X.

what\_relation(Y, X, "mother") :-

    mother(X, Y),

    Y \= X.

what\_relation(Y, X, "father") :-

    father(X, Y),

    Y \= X.

what\_relation(X, Y, "husband") :-

    mother(X, L),

    father(Y, L),

    Y \= X.

what\_relation(X, Y, "wife") :-

    father(X, L),

    mother(Y, L),

    Y \= X.

what\_relation(X, Y, "son") :-

    (father(X, Y); mother(X, Y)),

    male(Y),

    Y \= X.

what\_relation(X, Y, "daughter") :-

    (father(X, Y); mother(X, Y)),

    female(Y),

    Y \= X.

% ID-search

prolong\_d([X|T], [Y1|T1], [Y,X|T], T1) :-

    what\_relation(X, Y, Y1),

    not(member(Y, [X|T])).

relative\_d(A, X, Y) :-

    between(1, 4, N),

    dpth([X], Y, \_, L, N),

    reverse(L, A).

dpth([X|T], X, [X|T], [], 0).

dpth(P, F, L, A, N) :-

    N > 0,

    prolong\_d(P, A, P1, A1),

    N1 is N - 1,

    dpth(P1, F, L, A1, N1).

% Поиск в ширину

chain\_to\_relations([], []).

chain\_to\_relations([\_], []).

chain\_to\_relations([C1,C2|T], [R|N]) :-

    what\_relation(C2, C1, R),

    chain\_to\_relations([C2|T], N).

relative\_b(A, X, Y) :-

    bdth([[X]], Y, L),

    chain\_to\_relations(L, A).

prolong\_b([X|T], [Y,X|T]) :-

    what\_relation(Y, X, \_),

    not(member(Y, [X|T])).

bdth([[X|T]|\_], X, [X|T]).

bdth([P|QI], X, R) :-

    findall(Z, prolong\_b(P, Z), T),

    append(QI, T, QO),

    bdth(QO, X, R), !.

bdth([\_|T], Y, L) :-

    bdth(T, Y, L).

Тест

18 ?- relative\_d(R, "Максим Архипов", "Дмитрий Архипов").

R = ["father"] ;

20 ?- what\_relation("Дмитрий Архипов", "Елена Фурманчук", N).

N = "wife" ;

21 ?- what\_relation("Ирина Фурманчук", "Елена Фурманчук", N).

N = "sister"

**Вывод:** во время выполнения данного курсового проекта был закреплен навык работы с языком Prolog, а также освоен метод работы с файлами и парсингом их содержимого. Как оказалось, Prolog достаточно хорошо умеет справляться с подобного рода задачами, а код, получившийся в результате написания проекта, не превосходит пары страниц, что является преимуществом по сравнению с аналогичной программой, написанной на с++.