# Отчет по курсовому проекту

## по курсу "Логическое программирование"

### студент: Иванов Д.С.

## Результат проверки

| Преподаватель | Дата | Оценка |

|-------------------|--------------|---------------|

## Введение

За время выполнения курсового проекта я изучил обширный пласт логического программирования, применил многие навыки полученные при выполнении лабораторных работ, например методы поиска. Познакомился с таким форматом как GedCom, для построения генеалогических деревьев.

## Задание

1. Создать родословное дерево своего рода на несколько поколений (3-4) назад в стандартном формате GEDCOM.

2. Преобразовать файл в формате GEDCOM в набор утверждений на языке Prolog, используя следующее представление: с использованием предикатов father(отец, потомок) и mother(мать, потомок)

3. Реализовать предикат проверки/поиска золовки

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

5. Реализовать естественно-языковый интерфейс к системе, позволяющий задавать вопросы относительно степеней родства, и получать осмысленные ответы.

## Получение родословного дерева

Моё родословное дерево состоит из 26 человек и имеет 4 случая, в которых у мужа некоторой женщины есть сестра, то есть в дереве присутсвуют 4 золовки, которых нам надо искать.

## Конвертация родословного дерева

Для конвертации родословного дерева я использовал язык Python, поскольку он удобен для работы с текстом. Главным принципом моей программы является построчное считывание файла с запоминание родителя и потомка с дальнейшим выводом.

Определение предикатов father и mother:

```

while (line[1] == 'CHIL'):

if len(husband)>0:

pair = (husband, people[line[2]])

father.append(pair)

if len(wife)>0:

pair = (wife, people[line[2]])

mother.append(pair)

line = f.readline().split()

```

## Предикат поиска родственника

Предикат поиска золовки реализован следующим образом:

1. Ищем мужа

2. Определяем родителей

3. Находим второго ребенка этих родителей

4. Проверяем пол

```

sister\_in\_law(X,C):-

marrieds(X,Y),

father(F,Y),

mother(M,Y),

father(F,C),

mother(M,C),

mother(C).

marrieds(Y,T):-

father(T,C),

mother(Y,C),!.

mother(M):-

mother(M,\_),!.

```

Проверяем:

```

| ?- sister\_in\_law('Попова Мария',R).

R = 'Краснова Ксения' ? ;

(16 ms) no

| ?- sister\_in\_law('Сидорова Наталья',R).

R = 'Иванова Анастасия' ? ;

(16 ms) no

| ?- sister\_in\_law('Иванова Татьяна',R).

R = 'Сидорова Наталья' ? ;

(16 ms) no

```

## Выводы

Курсовой проект познакомил меня с логической парадигмой программирования, файлами формата GEDCOM и обработкой этих файлов. Я очень рад что смог, в некоторой степени, освоить Prolog, отойдя от императивных языков программирования. Поначалу Prolog ввел меня в ступор, ведь я привык описывать точный алгоритм выполнения программы, но со временем стало лучше.