|  |
| --- |
| московский авиационный институт (национальный исследовательский университет)  Кафедра вычислительной математики и программирования  Дисциплина “логическое программирование” |
| Курсовой проект |
|  |
| Группа: 8О-208Б  Студент: Захаров Игорь Сергеевич  Преподаватель: Сошников Дмитрий Валерьевич  Левинская Мария Александровна |

**Москва, 2018**

**Цель работы**

* Применить навыки логического и функционального программирования при решении реалистичной задачи
* Построить и проанализировать родословное дерево своего рода на несколько поколений

**Задание**

1. Создать родословное дерево своего рода на несколько поколений (3-4) назад в стандартном формате GEDCOM. Для этого рекомендуется использовать сервис MyHeritage.com, зарегистрировавшись в нем и подключив к заполнению дерева других родственников (указав их контактный e-mail при регистрации, что приведет к автоматической рассылке приглашения на их адрес). Если Вы по каким-то причинам не хотите использовать свою родословную - вы можете использовать файл [родословной европейской знати](http://www.rusgenealog.ru/gedcom/royal_gen.zip).
2. Преобразовать файл в формате GEDCOM в набор утверждений на языке Prolog, используя одно из следующих представлений в зависимости от номера в группе в соответствии с формулой ((4-1)+3) mod 4 + 1:
   1. с использованием предиката child(ребенок, родитель), male(человек), female(человек)
3. Реализовать предикат проверки/поиска следующих родственников в соответствии с номером варианта ((4-1) mod 7)+1:
   1. Двоюродный брат

**Описание метода решения:**

Для написания парсера к файлу формата .ged я использовал язык программировани Python, написав который я смог извлечь из полученного файла всю необходимую информацию.

**Код парсера:**

f1 = open("family.ged", "r")

f2 = open("answer.txt", "w")

base = {}

for line in f1.readlines():

words = line.split(" ")

if len(words) >= 3:

s1 = words[1]

s2 = words[2]

if s2[0] == "I":

key = words[1]

if s1 == "GIVN":

name = words[2]

if s1 == "SURN":

surn = words[2]

value = (name[:-1], surn[:-1])

newElem = {key:value}

base.update(newElem)

if s1 == "SEX":

if s2[:-1] == "F":

s = "female(\'%s\').\n" % (name[:-1] + " " + surn[:-1])

f2.write(s)

if s2[:-1] == "M":

s = "male(\'%s\').\n" % (name[:-1] + " " + surn[:-1])

f2.write(s)

if s1 == "HUSB":

husb = words[2]

for k, (a, b) in base.items():

if k == husb[:-1]:

father = a + " " + b

if s1 == "WIFE":

wife = words[2]

for k, (a, b) in base.items():

if k == wife[:-1]:

mother = a + " " + b

if s1 == "CHIL":

chil = words[2]

for k, (a, b) in base.items():

if k == chil[:-1]:

child = a + " " + b

s = "child('%s', '%s').\n" % (child, father)

f2.write(s)

s = "child('%s', '%s').\n" % (child, mother)

f2.write(s)

f2.close()

f1.close()

Данная программа проходит построчно файл, разбивает каждую его строчку на слова, заносит информацию о каждом человеке в словарь, а именно его ID, как ключ в словаре, которому сопоставляемая кортеж из имени и фамилии. Мы находим, чьим ребенком он является и составляем предикаты Child, также выводим предикаты Male и Female, определяя какой человек какого пола является.

**Сама программа на прологе**

male('Игорь Захаров').

female('Татьяна Ежикова').

male('Сергей Захаров').

female('Раиса Митякина').

male('Николай Ежиков').

female('Любовь Чуйкова').

male('Петр Захаров').

male('Роман Захаров').

female('Наталья Ежикова').

male('Максим Казаков').

male('Андрей Казаков').

female('Елена Казакова').

male('Василий Митякин').

female('Мария Редкошеева').

child('Игорь Захаров', 'Сергей Захаров').

child('Игорь Захаров', 'Татьяна Ежикова').

child('Роман Захаров', 'Сергей Захаров').

child('Роман Захаров', 'Татьяна Ежикова').

child('Наталья Ежикова', 'Николай Ежиков').

child('Наталья Ежикова', 'Раиса Митякина').

child('Татьяна Ежикова', 'Николай Ежиков').

child('Татьяна Ежикова', 'Раиса Митякина').

child('Сергей Захаров', 'Петр Захаров').

child('Сергей Захаров', 'Любовь Чуйкова').

child('Максим Казаков', 'Андрей Казаков').

child('Максим Казаков', 'Наталья Ежикова').

child('Елена Казакова', 'Андрей Казаков').

child('Елена Казакова', 'Наталья Ежикова').

child('Раиса Митякина', 'Василий Митякин').

child('Раиса Митякина', 'Мария Редкошеева').

cousin(X,Answer):- child(X,P), child(P,PP), child(Y,PP), Y \= P, child(Answer,Y), male(Answer).

**Демонстрация работы**

Welcome to SWI-Prolog (threaded, 64 bits, version 7.6.4)

SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.

Please run ?- license. for legal details.

For online help and background, visit http://www.swi-prolog.org

For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- cousin('Максим Казаков','Игорь Захаров').

true .

?- cousin('Игорь Захаров','Максим Казаков').

true .

?- cousin('Роман Захаров','Максим Казаков').

true .

?- cousin('Максим Казаков','Елена Казакова').

false.

В данной программе я реализовал предикат определения двоюродного брата. Она отвечает на вопрос, является ли X двоюродным братом для Answer.

**Вывод:**

Выполнение данного курсового проекта оказалось для меня очень увлекательным. Данный проект включается в себя очень разносторонние задания: от заполнения семейного дерева и написания программ на языках различных парадигм до написания реферата на увлекательную тему, которая заставляет поразмышлять. Приходилось применять парсинг на языке программирования Python и писать программу на Прологе, обрабатывающую данные. Благодаря этому проекту я узнал о новом интересном для себя сайте, позволяющем связывать между собой родословную историю.