

Московский Авиационный Институт

(Национальный исследовательский университет)

Курсовой проект

По курсу:

«Логическое программирование»

Студент: Забарин Н.И.

Группа: 8О – 308Б

№ по списку: 21

Москва, 2016

Цель работы:

- 1) Применить навыки логического и функционального программирования при решении реалистичной задачи
- 2) 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 других родственников (указав их контактный e-mail при регистрации, что приведет к автоматической рассылке приглашения на их адрес).
2. Преобразовать файл в формате GEDCOM в набор утверждений на языке Prolog, используя предиката `parents`(потомок, отец, мать). Для преобразования я буду использовать язык высокого уровня Python3.
3. Реализовать предикат поиска шурина.

Файл [family tree.ged](#):

```
0 HEAD
1 SOUR Reunion
2 VERS V8.0
2 CORP Leister Productions
1 DEST Reunion
1 DATE 11 FEB 2006
1 FILE test
1 GEDC
2 VERS 5.5
1 CHAR MACINTOSH
0 @I500001@ INDI
1 _UPD 12 MAY 2016 10:38:01 GMT -0500
1 NAME Игорь /Забарин/
2 GIVN Игорь
2 SURN Забарин
1 SEX M
1 FAMS @F500001@
1 FAMC @F500005@
1 RIN MH:I500001
1 _UID 56E3D689B4C8CD632484AF04E76FF9FA
0 @I500002@ INDI
1 _UPD 12 MAY 2016 10:34:00 GMT -0500
1 NAME Ирина /Рудакова/
2 GIVN Ирина
2 SURN Рудакова
1 SEX F
1 FAMS @F500001@
1 FAMC @F500002@
1 RIN MH:I500002
1 _UID 56E3D689C3666D642484AF04E76FF9FA
0 @I500003@ INDI
1 _UPD 12 MAR 2016 02:42:49 GMT -0500
1 NAME Никита /Забарин/
2 GIVN Никита
2 SURN Забарин
1 SEX M
1 BIRT
2 DATE 1995
```

1 RESI
2 EMAIL zaberomg@bk.ru
1 FAMC @F500001@
1 RIN MH:I500003
1 _UID 56E3D689D0B66D662484AF04E76FF9FA
0 @I500004@ INDI
1 _UPD 12 MAY 2016 10:36:36 GMT -0500
1 NAME Галина /Кускова/
2 GIVN Галина
2 SURN Кускова
1 SEX F
1 FAMS @F500002@
1 FAMC @F500004@
1 RIN MH:I500004
1 _UID 5734A26858F3964879791D3D6CEED1DE
0 @I500005@ INDI
1 _UPD 12 MAY 2016 10:34:59 GMT -0500
1 NAME Василий /Рудаков/
2 GIVN Василий
2 SURN Рудаков
1 SEX M
1 FAMS @F500002@
1 FAMC @F500003@
1 RIN MH:I500005
1 _UID 5734A282E94EBCAC1207A88D3E0B2F09
0 @I500006@ INDI
1 _UPD 12 MAY 2016 10:34:59 GMT -0500
1 NAME Петр /Рудаков/
2 GIVN Петр
2 SURN Рудаков
1 SEX M
1 FAMS @F500003@
1 RIN MH:I500006
1 _UID 5734A2A31400ED541F6105FDAF634D2F
0 @I500007@ INDI
1 _UPD 12 MAY 2016 10:36:06 GMT -0500
1 NAME Наталья /Захарова/
2 GIVN Наталья
2 SURN Захарова
1 SEX F
1 FAMS @F500003@
1 RIN MH:I500007
1 _UID 5734A2E66D4BD519CC0772A1CD69DCD5
0 @I500008@ INDI
1 _UPD 12 MAY 2016 10:36:36 GMT -0500
1 NAME Алексей /Кусков/
2 GIVN Алексей
2 SURN Кусков
1 SEX M
1 FAMS @F500004@
1 RIN MH:I500008
1 _UID 5734A30462350111EE98F88E0206491F
0 @I500009@ INDI
1 _UPD 12 MAY 2016 10:37:02 GMT -0500
1 NAME Клавдия /Похлебкина/
2 GIVN Клавдия
2 SURN Похлебкина
1 SEX F
1 FAMS @F500004@
1 RIN MH:I500009
1 _UID 5734A31E0157BD5D8FCD2C4BDA13A52C
0 @I500010@ INDI
1 _UPD 12 MAY 2016 10:38:02 GMT -0500
1 NAME Николай /Забарин/
2 GIVN Николай
2 SURN Забарин
1 SEX M
1 FAMS @F500005@
1 RIN MH:I500010
1 _UID 5734A359DB9CD7C06A6A9EF758B9DEE1
0 @I500011@ INDI
1 _UPD 12 MAY 2016 10:38:43 GMT -0500
1 NAME Маргарита //
2 GIVN Маргарита
1 SEX F
1 FAMS @F500005@
1 RIN MH:I500011
1 _UID 5734A38325408F6A91771BA3CABB06D7

0 @I500012@ INDI
1 _UPD 12 MAY 2016 10:39:29 GMT -0500
1 NAME Наталья /Забарина/
2 GIVN Наталья
2 SURN Забарина
1 SEX F
1 FAMS @F5000006@
1 FAMC @F5000005@
1 RIN MH:I500012
1 _UID 5734A3B1A091747668E774065EDBE03E
0 @I500013@ INDI
1 _UPD 12 MAY 2016 10:40:11 GMT -0500
1 NAME Светлана /Забарина/
2 GIVN Светлана
2 SURN Забарина
1 SEX F
1 FAMC @F5000001@
1 RIN MH:I500013
1 _UID 5734A3DB0D6A25A386A67DE57DDEA467
0 @I500014@ INDI
1 _UPD 12 MAY 2016 10:40:32 GMT -0500
1 NAME Семен /Забарин/
2 GIVN Семен
2 SURN Забарин
1 SEX M
1 FAMC @F5000001@
1 RIN MH:I500014
1 _UID 5734A3F078E120BADABB8E477C94342C
0 @I500015@ INDI
1 _UPD 12 MAY 2016 10:41:22 GMT -0500
1 NAME Роман /Опенько/
2 GIVN Роман
2 SURN Опенько
1 SEX M
1 FAMS @F5000006@
1 RIN MH:I500015
1 _UID 5734A421E759AE87598906588379BE3A
0 @I500016@ INDI
1 _UPD 12 MAY 2016 10:41:36 GMT -0500
1 NAME Виктория /Опенько/
2 GIVN Виктория
2 SURN Опенько
1 SEX F
1 FAMC @F5000006@
1 RIN MH:I500016
1 _UID 5734A42FE8E90DCDF6B80699096FA805
0 @F500001@ FAM
1 _UPD 12 MAR 2016 02:42:49 GMT -0500
1 HUSB @I500001@
1 WIFE @I500002@
1 CHIL @I500003@
1 CHIL @I500013@
1 CHIL @I500014@
1 RIN MH:F500001
1 _UID 56E3D689CCDC2D652484AF04E76FF9FA
0 @F500002@ FAM
1 _UPD 12 MAY 2016 10:34:27 GMT -0500
1 HUSB @I500005@
1 WIFE @I500004@
1 CHIL @I500002@
1 RIN MH:F500002
1 _UID 5734A2687192A64979791D3D6CEED1DE
0 @F500003@ FAM
1 _UPD 12 MAY 2016 10:36:06 GMT -0500
1 HUSB @I500006@
1 WIFE @I500007@
1 CHIL @I500005@
1 RIN MH:F500003
1 _UID 5734A2A3242A6D551F6105FDAF634D2F
0 @F500004@ FAM
1 _UPD 12 MAY 2016 10:37:02 GMT -0500
1 HUSB @I500008@
1 WIFE @I500009@
1 CHIL @I500004@
1 RIN MH:F500004
1 _UID 5734A3046CBEF112EE98F88E0206491F
0 @F500005@ FAM
1 _UPD 12 MAY 2016 10:38:43 GMT -0500

```

1 HUSB @I500010@
1 WIFE @I500011@
1 CHIL @I500001@
1 CHIL @I500012@
1 RIN MH:F500005
1 _UID 5734A359EAB1C7C16A6A9EF758B9DEE1
0 @F5000006@ FAM
1 _UPD 12 MAY 2016 10:41:21 GMT -0500
1 HUSB @I500015@
1 WIFE @I500012@
1 CHIL @I500016@
1 RIN MH:F500006
1 _UID 5734A421F36A8E88598906588379BE3A
1 MARR
0 TRLR

```

Файл convert.py:

```
from gedcom import Gedcom
```

```

def readgedcom(path):
    root = Gedcom(path)
    people = {}
    families = []
    d = root.element_dict()
    for i in d:
        elem = d[i]
        id = i
        if elem.is_individual():
            name = ""
            for i in elem.children():
                if i.tag() == 'NAME':
                    name = i.value().replace('/', '')
            people[id] = name
        else:
            fam = [[], 0, 0]
            for i in elem.children():
                if i.tag() == 'CHIL':
                    fam[0].append(i.value())
                if i.tag() == 'HUSB':
                    fam[1] = i.value()
                if i.tag() == 'WIFE':
                    fam[2] = i.value()
            families.append(fam)
    return (people, families)

people, families = readgedcom('tree1.ged')
for i in families:
    for child in i[0]:
        print('parents("{} ", "{} ", "{} ").format(people[child], people[i[1]], people[i[2]]))

print()
print('shurin(X, Ans):-')
print('\tparents(_, X, Wife), % X has a Wife')
print('\tparents(Wife, A, B), % Wife has a parents')
print('\tparents(Ans, A, B), % Wife's parents have a son or daughter - Ans')
print('\tparents(_, Ans, _). % Ans is a man')

print()
print('shurin("Роман Опенько", Ans).')

```

Файл tree.pl:

```

parents("Ирина Рудакова", "Василий Рудаков", "Галина Кускова").
parents("Виктория Опенько", "Роман Опенько", "Наталья Забарина").
parents("Никита Забарин", "Игорь Забарин", "Ирина Рудакова").
parents("Светлана Забарина", "Игорь Забарин", "Ирина Рудакова").
parents("Семен Забарин", "Игорь Забарин", "Ирина Рудакова").
parents("Игорь Забарин", "Николай Забарин", "Маргарита ").
parents("Наталья Забарина", "Николай Забарин", "Маргарита ").
parents("Василий Рудаков", "Петр Рудаков", "Наталья Захарова").
parents("Галина Кускова", "Алексей Кусков", "Клавдия Похлебкина").

shurin(X, Ans):-
    parents(_, X, Wife), % X has a Wife
    parents(Wife, A, B), % Wife has a parents
    parents(Ans, A, B), % Wife's parents have a son or daughter - Ans
    parents(_, Ans, _). % Ans is a man

```

shurin("Роман Опенько", Ans).

Тестирование:

```
swipl q.pl
```

```
Warning: /home/zaber/MAI/MAI/LP/kp/q.pl:17:
```

```
Singleton variables: [Ans]
```

```
Welcome to SWI-Prolog (Multi-threaded, 64 bits, Version 7.2.3)
```

```
Copyright (c) 1990-2015 University of Amsterdam, VU Amsterdam
```

```
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software,  
and you are welcome to redistribute it under certain conditions.
```

```
Please visit http://www.swi-prolog.org for details.
```

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

```
?- shurin("Роман Опенько", Ans).
```

```
Ans = "Игорь Забарин" .
```

Выводы:

Язык программирования Пролог — мощный инструмент для решения логических задач. Он предлагает совершенно иные возможности для написания программ, нежели популярные императивные и функциональные языки программирования. Также язык программирования Пролог довольно прост для освоения через процесс решения подобных несложных задач.

Решение данной задачи не просто коротко и красиво, оно обладает неплохой эффективностью. Декларативный язык Пролог предлагает хорошую альтернативу императивным языкам, на нём достаточно легко и быстро можно решить значительно более сложные задачи.