

Лабораторна робота 1

Створено системою Doxygen 1.9.1

1 Звіт з лабораторної роботи 1	1
1.1 Постановка задачі	1
1.2 Побудова сімейного дерева	2
1.3 Результат роботи програми	3
1.4 Висновки	6
2 Показчик файлів	6
2.1 Файли	6
3 Файли	7
3.1 Файл main.pl	7
3.2 main.pl	7
3.3 Файл mainpage.dox	9

1 Звіт з лабораторної роботи 1

за дисципліною "Інтелектуальні інформаційні системи"
 студента групи ПА-17-2
 Панасенка Єгора Сергійовича
 Кафедра комп'ютерних технологій
 ФПМ, ДНУ, 2017-2018 навч.р.

Звіт доступний за посиланням

https://gaurapanasenko.github.io/unilab_opt/IIS_Lab1/html/index.html.

Вихідний код доступний за посиланням

https://github.com/gaurapanasenko/unilab/tree/master/08/IIS_Lab1

1.1 Постановка задачі

Використовуючи предикати `parent(string parent, string son-daughter)`, `women(symbol)` і `man(symbol)`, розробити предикати, що визначають такі поняття:

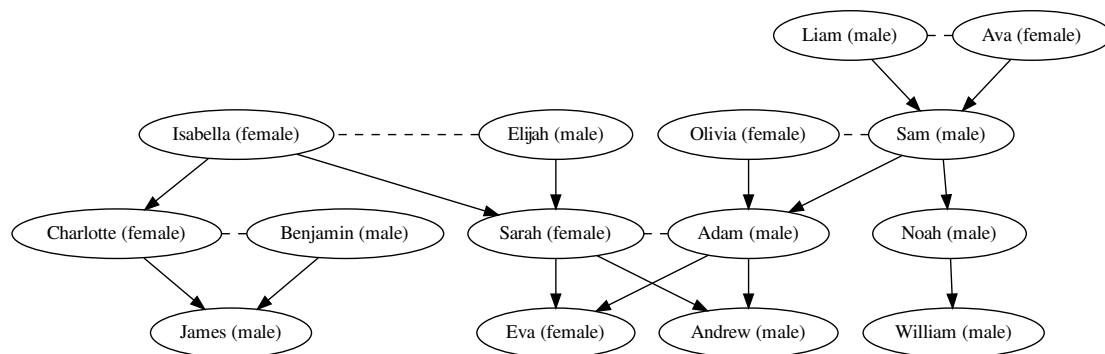
1. Правнучка
2. Двоюрідний брат.
3. Внук
4. Дівер
5. Зять
6. Тесть
7. Свекруха
8. Теща
9. Свекор
10. Прадідусь
11. Племінник

12. Прабабуся
13. Племянница
14. Свояк
15. Тітка
16. Своячка
17. Дядько
18. Дідусь
19. Правнук
20. Внучка
21. Бабуся

Написати програму для пошуку і видачі на екран відповідного родича в базі даних.

1.2 Побудова сімейного дерева

Побудуємо деяке сімейне дерево.



Тепер розглянемо деякі приклади по списку.

Номер	Назва	Код	Запит	Відповідь	Додаткова інформація
1	Правнучка	great_↔ granddaughter	Ava	Eva	-
2	Двоюридний брат	cousin	Andrew	William	-
3	Внук	grandson	Sam	Andrew	-
4	Дівер	brother_in_law	Sarah	Noah	брат чоловіка
5	Зять	son_in_law	Elijah	Adam	чоловік дочки
6	Тесть	father_in_law1	Adam	Elijah	батько дружини
7	Свекруха	mother_in_law1	Sarah	Olivia	мати чоловіка
8	Теща	mother_in_law2	Adam	Isabella	мати дружини
9	Свекор	father_in_law2	Sarah	Sam	батько чоловіка

Номер	Назва	Код	Запит	Відповідь	Додаткова інформація
10	Прадідусь	great_grandfather	Andrew	Liam	-
11	Племінник	nephew	Noah	Andrew	-
12	Прабабуся	great_↔ grandmother	Andrew	Ava	-
13	Племінниця	niece	Noah	Eva	-
14	Свояк	sibling_in_law	Benjamin	Adam	чоловік сестри дружини
15	Тітка	aunt	Eva	Charlotte	-
16	Своячка	sister_in_law	Adam	Charlotte	сестра дружини
17	Дядько	uncle	Eva	Noah	-
18	Дідусь	grandfather	Andrew	Sam	-
19	Правнук	great_grandson	Liam	Andrew	-
20	Внучка	granddaughter	Sam	Eva	-
21	Бабуся	grandmother	Andrew	Olivia	-

1.3 Результат роботи програми

Використаємо компілятор `swipl`, для початку зкомпілюємо нашу програму.

```
[gaura@gauramid IIS_Lab1]$ swipl -c main.pl
% autoloading prolog_codewalk:must_be/2 from /usr/lib64/swipl-7.6.4/library/error
% autoloading prolog_debug:backtrace/1 from /usr/lib64/swipl-7.6.4/library/prolog_stack
% autoloading prolog_codewalk:portray_clause/1 from /usr/lib64/swipl-7.6.4/library/listing
% autoloading prolog_codewalk:clause_info/4 from /usr/lib64/swipl-7.6.4/library/prolog_clause
% autoloading prolog_codewalk:initialization_layout/4 from /usr/lib64/swipl-7.6.4/library/prolog_clause
% autoloading qsave:current_foreign_library/2 from /usr/lib64/swipl-7.6.4/library/shlib
% autoloading oset:reverse/2 from /usr/lib64/swipl-7.6.4/library/lists
% autoloading record:member/2 from /usr/lib64/swipl-7.6.4/library/lists
% autoloading plunit:prolog_stack_frame_property/2 from /usr/lib64/swipl-7.6.4/library/prolog_stack
% autoloading plunit:get_prolog_backtrace/2 from /usr/lib64/swipl-7.6.4/library/prolog_stack
% autoloading error:assertion/1 from /usr/lib64/swipl-7.6.4/library/debug
% autoloading prolog_codewalk:clause_name/2 from /usr/lib64/swipl-7.6.4/library/prolog_clause
% Autoloader: iteration 1 resolved 10 predicates and loaded 12 files in 0.274 seconds. Restarting ...
% Autoloader: loaded 10 files in 2 iterations in 0.389 seconds
```

Тепер запусимо програму та перевіримо відповідність предикатів із прикладами наведеними у таблиці.

```
[gaura@gauramid IIS_Lab1]$ ./a.out
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).
?- run_tests.
% PL-Unit: main
Warning: /home/gaura/git/gaurapanasenko/unilab/08/IIS_Lab1/main.pl:164:
  PL-Unit: Test x: Test succeeded with choicepoint
.
Warning: /home/gaura/git/gaurapanasenko/unilab/08/IIS_Lab1/main.pl:168:
  PL-Unit: Test x: Test succeeded with choicepoint
Warning: /home/gaura/git/gaurapanasenko/unilab/08/IIS_Lab1/main.pl:170:
  PL-Unit: Test x: Test succeeded with choicepoint
Warning: /home/gaura/git/gaurapanasenko/unilab/08/IIS_Lab1/main.pl:172:
  PL-Unit: Test x: Test succeeded with choicepoint
Warning: /home/gaura/git/gaurapanasenko/unilab/08/IIS_Lab1/main.pl:174:
  PL-Unit: Test x: Test succeeded with choicepoint
Warning: /home/gaura/git/gaurapanasenko/unilab/08/IIS_Lab1/main.pl:176:
  PL-Unit: Test x: Test succeeded with choicepoint
Warning: /home/gaura/git/gaurapanasenko/unilab/08/IIS_Lab1/main.pl:178:
  PL-Unit: Test x: Test succeeded with choicepoint
Warning: /home/gaura/git/gaurapanasenko/unilab/08/IIS_Lab1/main.pl:180:
  PL-Unit: Test x: Test succeeded with choicepoint
Warning: /home/gaura/git/gaurapanasenko/unilab/08/IIS_Lab1/main.pl:182:
  PL-Unit: Test x: Test succeeded with choicepoint
Warning: /home/gaura/git/gaurapanasenko/unilab/08/IIS_Lab1/main.pl:184:
  PL-Unit: Test x: Test succeeded with choicepoint
Warning: /home/gaura/git/gaurapanasenko/unilab/08/IIS_Lab1/main.pl:186:
  PL-Unit: Test x: Test succeeded with choicepoint
```

```

Warning: /home/gaura/git/gaurapanasenko/unilab/08/IIS_Lab1/main.pl:188:
  PL-Unit: Test x: Test succeeded with choicepoint
Warning: /home/gaura/git/gaurapanasenko/unilab/08/IIS_Lab1/main.pl:190:
  PL-Unit: Test x: Test succeeded with choicepoint
Warning: /home/gaura/git/gaurapanasenko/unilab/08/IIS_Lab1/main.pl:192:
  PL-Unit: Test x: Test succeeded with choicepoint
Warning: /home/gaura/git/gaurapanasenko/unilab/08/IIS_Lab1/main.pl:194:
  PL-Unit: Test x: Test succeeded with choicepoint
Warning: /home/gaura/git/gaurapanasenko/unilab/08/IIS_Lab1/main.pl:196:
  PL-Unit: Test x: Test succeeded with choicepoint
Warning: /home/gaura/git/gaurapanasenko/unilab/08/IIS_Lab1/main.pl:198:
  PL-Unit: Test x: Test succeeded with choicepoint
.
Warning: /home/gaura/git/gaurapanasenko/unilab/08/IIS_Lab1/main.pl:202:
  PL-Unit: Test x: Test succeeded with choicepoint
. done
% All 21 tests passed
true.
?- halt.

```

Тепер запустимо усі предикати.

```

[gaura@gauramid IIS_Lab1]$ ./a.out
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).
?- great_granddaughter(A,B).
A = "Liam",
B = "Eva" ;
A = "Ava",
B = "Eva" ;
false.
?- cousin(A,B).
A = "Eva",
B = "James" ;
A = "Andrew",
B = "James" ;
A = "James",
B = "Andrew" ;
A = "William",
B = "Andrew" ;
A = "Eva",
B = "William" ;
A = "Andrew",
B = "William" ;
false.
?- grandson(A,B).
A = "Liam",
B = "Adam" ;
A = "Liam",
B = "Noah" ;
A = "Ava",
B = "Adam" ;
A = "Ava",
B = "Noah" ;
A = "Isabella",
B = "James" ;
A = "Isabella",
B = "Andrew" ;
A = "Elijah",
B = "Andrew" ;
A = "Olivia",
B = "Andrew" ;
A = "Sam",
B = "Andrew" ;
A = "Sam",
B = "William" ;
false.
?- brother_in_law(A,B).
A = "Sarah",
B = "Noah" ;
A = "Sarah",
B = "Noah" ;
false.
?- son_in_law(A,B).
A = "Isabella",
B = "Benjamin" ;
A = "Isabella",
B = "Adam" ;
A = "Isabella",
B = "Adam" ;
A = "Elijah",
B = "Adam" ;
A = "Elijah",
B = "Adam" ;

```

```

false.
?- father_in_law1(A,B).
A = "Adam" ;
B = "Elijah" ;
A = "Adam" ;
B = "Elijah" ;
false.
?- mother_in_law1(A,B).
A = "Olivia" ;
B = "Ava" ;
A = "Sarah" ;
B = "Olivia" ;
A = "Sarah" ;
B = "Olivia" ;
false.
?- mother_in_law2(A,B).
A = "Benjamin" ;
B = "Isabella" ;
A = "Adam" ;
B = "Isabella" ;
A = "Adam" ;
B = "Isabella" ;
false.
?- father_in_law2(A,B).
A = "Olivia" ;
B = "Liam" ;
A = "Sarah" ;
B = "Sam" ;
A = "Sarah" ;
B = "Sam" ;
false.
?- great_grandfather(A,B).
A = "Eva" ;
B = "Liam" ;
A = "Andrew" ;
B = "Liam" ;
A = "William" ;
B = "Liam" ;
false.
?- nephew(A,B).
A = "Sarah" ;
B = "James" ;
A = "Charlotte" ;
B = "Andrew" ;
A = "Noah" ;
B = "Andrew" ;
A = "Adam" ;
B = "William" ;
false.
?- great_grandmother(A,B).
A = "Eva" ;
B = "Ava" ;
A = "Andrew" ;
B = "Ava" ;
A = "William" ;
B = "Ava" ;
false.
?- niece(A,B).
A = "Charlotte" ;
B = "Eva" ;
A = "Noah" ;
B = "Eva" ;
false.
?- sibling_in_law(A,B).
A = "Adam" ;
B = "Benjamin" ;
A = "Adam" ;
B = "Benjamin" ;
A = "Benjamin" ;
B = "Adam" ;
A = "Benjamin" ;
B = "Adam" ;
false.
?- aunt(A,B).
A = "Eva" ;
B = "Charlotte" ;
A = "Andrew" ;
B = "Charlotte" ;
A = "James" ;
B = "Sarah" ;
false.
?- sister_in_law(A,B).
A = "Adam" ;
B = "Charlotte" ;
A = "Adam" ;
B = "Charlotte" ;
A = "Benjamin" ;

```

```
B = "Sarah" ;
false.
?- uncle(A,B).
A = "William",
B = "Adam" ;
A = "Eva",
B = "Noah" ;
A = "Andrew",
B = "Noah" ;
false.
?- grandfather(A,B).
A = "Adam",
B = "Liam" ;
A = "Noah",
B = "Liam" ;
A = "Eva",
B = "Elijah" ;
A = "Andrew",
B = "Elijah" ;
A = "Eva",
B = "Sam" ;
A = "Andrew",
B = "Sam" ;
A = "William",
B = "Sam" ;
false.
?- great_grandson(A,B).
A = "Liam",
B = "Andrew" ;
A = "Ava",
B = "Andrew" ;
A = "Liam",
B = "William" ;
A = "Ava",
B = "William".
?- granddaughter(A,B).
A = "Isabella",
B = "Eva" ;
A = "Elijah",
B = "Eva" ;
A = "Olivia",
B = "Eva" ;
A = "Sam",
B = "Eva".
?- grandmother(A,B).
A = "Adam",
B = "Ava" ;
A = "Noah",
B = "Ava" ;
A = "James",
B = "Isabella" ;
A = "Eva",
B = "Isabella" ;
A = "Andrew",
B = "Isabella" ;
A = "Eva",
B = "Olivia" ;
A = "Andrew",
B = "Olivia" ;
false.
?- halt.
```

1.4 Висновки

Було розроблено пролог програму та перевірено деяке сімейне дерево за допомогою цієї програми, також було виведено на екран усі перелічені варіанти сімених зв'язків перелічених у постановці задачі.

2 Показчик файлів

2.1 Файли

Повний список файлів.

main.pl

7

3 Файли

3.1 Файл main.pl

3.2 main.pl

```

00001 in_law(A,B) :-
00002     parent(A,C),
00003     parent(B,C),
00004     A\=B.
00005
00006 sibling(A,B) :-
00007     parent(C,A),
00008     parent(C,B),
00009     A\=B.
00010
00011 great_granddaughter(A,B) :- %1
00012     parent(A,C),
00013     parent(C,D),
00014     parent(D,B),
00015     female(B).
00016
00017 cousin(A,B) :- %2
00018     male(B),
00019     parent(C,A),
00020     parent(D,B),
00021     sibling(C,D).
00022
00023 grandson(A,B) :- %3
00024     parent(A,C),
00025     parent(C,B),
00026     male(B).
00027
00028 brother_in_law(A,B) :- %4
00029     in_law(A,C),
00030     male(C),
00031     sibling(C,B),
00032     male(B).
00033
00034 son_in_law(A,B) :- %5
00035     parent(A,C),
00036     in_law(C,B),
00037     male(B).
00038
00039 father_in_law1(A,B) :- %6
00040     parent(B,C),
00041     in_law(C,A),
00042     female(C),
00043     male(B).
00044
00045 mother_in_law1(A,B) :- %7
00046     parent(B,C),
00047     female(B),
00048     in_law(A,C),
00049     male(C).
00050
00051 mother_in_law2(A,B) :- %8
00052     parent(B,C),
00053     female(B),
00054     in_law(A,C),
00055     female(C).
00056
00057 father_in_law2(A,B) :- %9
00058     parent(B,C),
00059     in_law(C,A),
00060     male(C),
00061     male(B).
00062
00063 great_grandfather(A,B) :- %10
00064     parent(B,C),
00065     parent(C,D),
00066     parent(D,A),
00067     male(B).
00068
00069 nephew(A,B) :- %11
00070     male(B),
00071     parent(C,B),

```

```

00072 sibling(A,C).
00073
00074 great_grandmother(A,B) :- %12
00075     parent(B,C),
00076     parent(C,D),
00077     parent(D,A),
00078     female(B).
00079
00080 niece(A,B) :- %13
00081     female(B),
00082     parent(C,B),
00083     sibling(A,C).
00084
00085 sibling_in_law(A,B) :- %14
00086     male(B),
00087     in_law(B,C),
00088     female(C),
00089     sibling(C,D),
00090     female(D),
00091     in_law(D,A).
00092
00093 aunt(A,B) :- %15
00094     female(B),
00095     sibling(B,C),
00096     parent(C,A).
00097
00098 sister_in_law(A,B) :- %16
00099     female(B),
00100     sibling(B,C),
00101     female(C),
00102     in_law(A,C).
00103
00104 uncle(A,B) :- %17
00105     male(B),
00106     sibling(B,C),
00107     parent(C,A).
00108
00109 grandfather(A,B) :- %18
00110     male(B),
00111     parent(B, C),
00112     parent(C, A).
00113
00114 great_grandson(A,B) :- %19
00115     male(B),
00116     parent(C, B),
00117     parent(D, C),
00118     parent(A, D).
00119
00120 granddaughter(A,B) :- %20
00121     female(B),
00122     parent(C,B),
00123     parent(A,C).
00124
00125 grandmother(A,B) :- %21
00126     female(B),
00127     parent(B,C),
00128     parent(C,A).
00129
00130 parent("Liam", "Sam").
00131 parent("Ava", "Sam").
00132 parent("Isabella", "Charlotte").
00133 parent("Isabella", "Sarah").
00134 parent("Elijah", "Sarah").
00135 parent("Olivia", "Adam").
00136 parent("Sam", "Adam").
00137 parent("Sam", "Noah").
00138 parent("Charlotte", "James").
00139 parent("Benjamin", "James").
00140 parent("Sarah", "Eva").
00141 parent("Adam", "Eva").
00142 parent("Sarah", "Andrew").
00143 parent("Adam", "Andrew").
00144 parent("Noah", "William").
00145
00146 male("Liam").
00147 male("Elijah").
00148 male("Sam").
00149 male("Benjamin").
00150 male("Adam").
00151 male("Noah").
00152 male("James").
00153 male("Andrew").
00154 male("William").
00155
00156 female("Ava").
00157 female("Isabella").
00158 female("Olivia").

```

```
00159 female("Charlotte").
00160 female("Sarah").
00161 female("Eva").
00162
00163 :- begin_tests(main).
00164 test(x) :-
00165     great_granddaughter("Ava", "Eva").
00166 test(x) :-
00167     cousin("Andrew", "William").
00168 test(x) :-
00169     grandson("Sam", "Andrew").
00170 test(x) :-
00171     brother_in_law("Sarah", "Noah").
00172 test(x) :-
00173     son_in_law("Elijah", "Adam").
00174 test(x) :-
00175     father_in_law1("Adam", "Elijah").
00176 test(x) :-
00177     mother_in_law1("Sarah", "Olivia").
00178 test(x) :-
00179     mother_in_law2("Adam", "Isabella").
00180 test(x) :-
00181     father_in_law2("Sarah", "Sam").
00182 test(x) :-
00183     great_grandfather("Andrew", "Liam").
00184 test(x) :-
00185     nephew("Noah", "Andrew").
00186 test(x) :-
00187     great_grandmother("Andrew", "Ava").
00188 test(x) :-
00189     niece("Noah", "Eva").
00190 test(x) :-
00191     sibling_in_law("Benjamin", "Adam").
00192 test(x) :-
00193     aunt("Eva", "Charlotte").
00194 test(x) :-
00195     sister_in_law("Adam", "Charlotte").
00196 test(x) :-
00197     uncle("Eva", "Noah").
00198 test(x) :-
00199     grandfather("Andrew", "Sam").
00200 test(x) :-
00201     great_grandson("Liam", "Andrew").
00202 test(x) :-
00203     granddaughter("Sam", "Eva").
00204 test(x) :-
00205     grandmother("Andrew", "Olivia").
00206
00207 :- end_tests(main).
```

3.3 Файл mainpage.dox

