МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

«УФИМСКИЙ ГОСУДАРСТВЕННЫЙ АВИАЦИОННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра вычислительной математики и кибернетики

Отчет по лабораторной работе №6

«Функциональное и логическое программирование»

Выполнил: студент гр. МО-321

Шемануев А.Е.

Проверил: Профессор, д/н, доцент

Ризванов Д.А.

Уфа – 2022

# **Цель работы**

Реализовать на Прологе задачу описания родственных отношений с использованием динамических баз данных. Предусмотреть возможность добавления, изменения и удаления данных, а также последующего сохранения внесенных изменений.

# **Ход работы**

Дерево родственных отношений:

База данных:

hum("Пам")

hum("Пат")

hum("Энн")

hum("Лиз")

hum("Том")

hum("Боб")

hum("Джим")

hum("Кек")

wom("Пам")

wom("Пат")

wom("Энн")

wom("Лиз")

man("Том")

man("Боб")

man("Джим")

man("Кек")

par("Пам", "Боб") )

par("Том", "Лиз") )

par("Том", "Боб") ),

par("Боб", "Энн") )

par("Боб", "Пат") ),

par("Пат", "Джим") ),

par("Энн","Кек") ),

Интерфейс программы:



Нажимая на символы мы выбираем соответствующие предикаты.

Листинг

|  |
| --- |
| database  par(string, string)  wom(string)  man(string)  hum(string)  domains  str = string  i = integer  predicates  %nondeterm par(str, str).  nondeterm print1.  nondeterm print2.  nondeterm print3.  nondeterm print4.    nondeterm father(str, str).  nondeterm mother(str, str).    nondeterm son(str, str).  nondeterm daughter(str, str).  nondeterm brother(str, str).  nondeterm sister(str, str).    nondeterm grandfather(str, str).  nondeterm grandmother(str, str).    nondeterm uncle(str, str).  nondeterm aunt(str, str).    nondeterm nephew(str, str). %Племяник  nondeterm niece(str, str). %Племяница    nondeterm ancestor(str, str). %предок  nondeterm offspring(str, str) %Отпрыск    nondeterm sameGeneration(str, str). %предок  nondeterm sameGenerationU(str, str). %предок  nondeterm sameGenerationD(str, str). %предок    nondeterm child(str).  nondeterm adult(str).      nondeterm menu().  nondeterm input().  nondeterm action(str).    nondeterm addI().  nondeterm removeСonnectionI().  nondeterm removePersonI().  nondeterm renameI().  nondeterm changeGenderI().    nondeterm rename(str, str).  nondeterm renameGender(str, str).  nondeterm renamePar(str, str).  nondeterm renameChil(str, str).  nondeterm removePerson(str).    nondeterm changeGender(str, str).  clauses  %ft("Пам", "Боб").  print1:- write("4"), nl, par(A,B), write(A," - ",B),nl, fail.  print1.  print2:- write("4"), nl, wom(A), write("wom - ", A),nl, fail.  print2.  print3:- write("4"), nl, man(A), write("man - ", A),nl, fail.  print3.  print4:- write("4"), nl, hum(A),write("hum - ", A),nl, fail.  print4.    father(A,B) :- par(A,B), man(A).  mother(A,B) :- par(A,B), wom(A).    son(A,B) :- par(B,A), man(A).  daughter(A,B) :- par(B,A), wom(A).    brother(A, B) :- man(A), par(C, A), par(C, B), not(A=B).  sister(A, B) :- wom(A), par(C, A), par(C, B), not(A=B).    grandfather(A, B) :- man(A), par(A,C), par(C, B).  grandmother(A, B) :- wom(A), par(A,C), par(C, B).    uncle(A, B) :- man(A), brother(A,C), par(C, B).  uncle(A, B) :- man(A), sister(A,C), par(C, B).    aunt(A, B) :- wom(A), brother(A,C), par(C, B).  aunt(A, B) :- wom(A), sister(A,C), par(C, B).    nephew(A, B) :- man(A), par(C, A), brother(C,B).  nephew(A, B) :- man(A), par(C, A), sister(C,B).    niece(A, B) :- wom(A), par(C, A), brother(C,B).  niece(A, B) :- wom(A), par(C, A), sister(C,B).    ancestor(A,B) :- par(A,B).  ancestor(A,B) :- par(C, B), ancestor(A,C).    offspring(A,B) :- ancestor(B,A).    sameGeneration(A,B) :- sameGenerationU(A,B); sameGenerationD(A,B) .  sameGenerationU(A,B) :- A=B.%; sister(A,B) ; brother(A,B).  sameGenerationU(A,B) :- par(C,A), par(D,B), sameGenerationU(C,D).  sameGenerationD(A,B) :- A=B.%; sister(A,B) ; brother(A,B).  sameGenerationD(A,B) :- par(A,C), par(B,D), sameGenerationD(C,D).    child(A) :- par(A,B), !,fail.  child(A).    adult(A) :- not(child(A)).    %Вывод  %%Вывод  %Добавить  %Удалить  %Переименовать  %Сохранить    %Save("C:/Path")  %Consult("C:/Path")    rename(Old, New):- hum(Old), assert(hum(New)), renameGender(Old, New), renamePar(Old, New), renameChil(Old, New), retract(hum(Old)).  renameGender(Old, New):- man(Old), assert(man(New)), retract(man(Old)).  renameGender(Old, New):- wom(Old), assert(wom(New)), retract(wom(Old)).    renamePar(Old, New):- par(Old, A), assert(par(New, A)), retract(par(Old, A)), renamePar(Old, New).  renamePar(\_, \_).  renameChil(Old, New):- par(A, Old), assert(par(A, New)), retract(par(A, Old)), renameChil(Old, New).  renameChil(\_, \_).  removePerson(Per):- retractall(par(Per, \_)), fail.  removePerson(Per):- retractall(par(\_, Per)), fail.  removePerson(Per):- retract(man(Per)), fail.  removePerson(Per):- retract(wom(Per)), fail.  removePerson(Per):- retract(hum(Per)).      changeGender(Per, G):- not(G="Ж"); not(G="М"), fail, !.  changeGender(Per, G):- retract(man(Per)), fail.  changeGender(Per, G):- retract(wom(Per)), fail.  changeGender(Per, G):- G = "М", assert(man(Per)).  changeGender(Per, G):- G = "Ж", assert(wom(Per)).    addI():- write("ВВедите родителя: "), readln(Par), nl,  write("ВВедите ребёнка : "), readln(Chi), nl,  assert(par(Par, Chi)).  addI():- write("Ошибка").      removeСonnectionI():- write("Введите родителя: "), readln(Par), nl,  write("Введите ребёнка : "), readln(Chi), nl,  retract(par(Par, Chi)).  removeСonnectionI():- write("Ошибка").    removePersonI():-write("кого убрать: "), readln(Per), nl, removePerson(Per).    removePersonI():-write("Ошибка").    renameI():- write("Введите кого переименовать: "), readln(Old), nl,  write("Введите какое имя дать? : "), readln(New), nl,  rename(Old, New).  renameI():- write("Ошибка").  changeGenderI():-write("Введите кого поменять: "), readln(Per), nl,  write("Пол[М\Ж]: "), readln(G), changeGender(Per, G).    menu():-nl,  write("[0] - Вывод"),nl,  write("[1] - Добавить"),nl,  write("[2] - Удалить связь"),nl,  write("[3] - Удалить персону"),nl,  write("[4] - Переименовать"),nl,  write("[5] - Поменять\добавить пол"),nl,  write("[6] - Сохранить "),nl,  input().      input():-readln(I), action(I).  action("0"):-print1, print2, print3, print4, nl, menu().  action("1"):-addI(), menu().  action("2"):-removeСonnectionI(), menu().  action("3"):-removePersonI(), menu().  action("4"):-renameI(), menu().  action("5"):-changeGenderI(), menu().  action("6"):-save("C:/Users/Orkii/Desktop/Prolog/Labs/my.txt"), menu().    goal  consult("C:/Users/Orkii/Desktop/Prolog/Labs/my.txt"),  print1, print2, print3, print4, write(" "),    menu(). |