Abgabe: Am Montag, 6.12. 8.00 Uhr durch die Abgabemaske auf der Kursseite

Aufgabe 16

Gegeben sei die folgende Prolog Datenbank mit Fakten der Form

person(dieter,männlich,alter(38)).
person(jack,männlich,alter(32)).
person(peter,männlich,alter(45)).
person(opa,männlich,alter(80)).
person(peter,männlich,alter(80)).

person(katrin,weiblich,alter(28)).
person(katrin,weiblich,alter(25)).
person(katrin,weiblich,alter(23)).
person(katrin,weiblich,alter(79)).

16.1) Schreiben Sie ein Programm mit dem sie eine Liste der weiblichen(männlichen) Personen erzeugen können.
Tipp, verwenden Sie das findall Prädikat. Dieses Prädikat wurde bislang nicht verwendet, finden Sie heraus was es tut mit den Prädikaten listing und help.

```
% findall/3(findall(Term,Ziel,Termliste)) dient dazu, sämtliche Lösungen
% eines Zielaufrufs zu ermitteln und die Lösungen in Form einer
% Ergebnisliste darzustellen.
% Die Aufgabe der einzelnen Argumente:
% 1.Argument(Term): Eine Variable, die im Zielaufruf vorkommt.
% 2.Argument: Ziel
% 3.Argument: Liste aller Bindungen der Variablen, die im ersten
% Argument angegeben sind.
% Beispiel: findall((X,Y), append(X,Y,[1,2,3]),L).
% L=[([],[1,2,3]),([1],[2,3]),([1,2],[3]),([1,2,3],[])].

% Aufgabe 16.1)
weibliche(L):-
        findall(X,person(X,weiblich,_),L).
%?- weibliche(L).
%L = [katrin, eva, isabela, sophie].
maennliche(L):-
        findall(X,person(X,männlich,_),L).
%?- maennlich(L).
%L = [dieter, jack, peter, opa, peter].
```

16.2)
Schreiben Sie jetzt ein Programm mit dem Sie die älteste weibliche bzw. männliche Person mit Ihrem Alter ermitteln können, (aelteste(Genus,X)) und die jüngsten weiblichen bzw. männlichen Personen, (jüngste(Genus,X)) Geben Sie diese in der Form

Person: xxx Alter:yy

aus. Zum finden der ältesten, jüngsten Person in einer Liste verwenden Sie jeweils Minimum. Maximum Prädikate.

```
% Aufgabe 16.2 Lösung

hole_aelteste(Genus,Liste):-
        findall(pers(Name,Alter),person(Name,Genus,alter(Alter)), Gesamtliste),
        baue_liste_aelteste(Gesamtliste,Liste), print_out(Liste).

hole_juengste(Genus,Liste):-
        findall(pers(Name,Alter),person(Name,Genus,alter(Alter)), Gesamtliste),
        baue_liste_juengste(Gesamtliste,Liste), print_out(Liste).

baue_liste_aelteste(Gesamtliste,Liste):- suche_maximum(Gesamtliste,Max), hole_person(Max,Gesamtliste,Liste).

baue_liste_juengste(Gesamtliste,Liste):- suche_minimum(Gesamtliste,Min), hole_person(Min,Gesamtliste,Liste).


hole_person(_,[],[]).
hole_person(Alter, [pers(Name,Alter)|Restliste], [pers(Name,Alter)|RestErgebnisListe]):-
                !,hole_person(Alter,Restliste,RestErgebnisListe).
hole_person(Alter, [_|Restliste], ErgebnisListe):-
                hole_person(Alter,Restliste,ErgebnisListe).

suche_maximum([ ],0).
suche_maximum([pers(_,Alter)|RestListe],Max):- suche_maximum(RestListe,Max1), maximum(Alter,Max1,Max).

suche_minimum([ ],1000).
suche_minimum([pers(_,Alter)|RestListe],Min):- suche_minimum(RestListe,Min1), minimum(Alter,Min1,Min).


maximum(X,Y,Y):- X<Y,!.
maximum(X,_,X).

minimum(X,Y,Y):- Y<X,!.
minimum(X,_,X).

print_out([ ]) :- nl.   %das waren alle Personen
print_out([pers(Name,Alter)|RestListe]) :- write('Person: '), write(Name), write(' Alter: '),
write(Alter),nl,print_out(RestListe).
```

```
% Aufgabe 16.2 Alternative Lösung)

aelteste(weiblich,WeibAlter):-
        write('Die älteste Frau ist '),nl,
        findall(Alter,person(_,weiblich,alter(Alter)),ListeAlter),
        maxListe(ListeAlter,MaxAlter),
        findall((person:X,alter:MaxAlter),person(X,weiblich,alter(MaxAlter)),WeibAlter),!.
%?- aelteste(weiblich,X).
%Die älteste Frau ist
%X = [ (person:sophie, alter:79)].
aelteste(männlich,WeibAlter):-
        write('Der älteste Mann ist '),nl,
        findall(Alter,person(_,männlich,alter(Alter)),ListeAlter),
        maxListe(ListeAlter,MaxAlter),
        findall((person:X,alter:MaxAlter),person(X,männlich,alter(MaxAlter)),WeibAlter),!.
%?- aelteste(männlich,X).
%Der älteste Mann ist
%X = [ (person:opa, alter:80), (person:peter, alter:80)].
juengste(weiblich,WeibAlter):-
        write('Die jüngste Frau ist '),nl,
        findall(Alter,person(_,weiblich,alter(Alter)),ListeAlter),
        minListe(ListeAlter,MaxAlter),
        findall((person:X,alter:MaxAlter),person(X,weiblich,alter(MaxAlter)),WeibAlter),!.
%?- juengste(weiblich,X).
%Die jüngste Frau ist
%X = [ (person:isabela, alter:23)].
juengste(männlich,Alter):-
        write('Der jüngste Mann ist '),nl,
        findall(Alter,person(_,männlich,alter(Alter)),ListeAlter),
        minListe(ListeAlter,MaxAlter),
        findall((person:X,alter:MaxAlter),person(X,männlich,alter(MaxAlter)),Alter),!.
%?- juengste(männlich,X).
%Der jüngste Mann ist
%X = [ (person:jack, alter:32)].
% Das Maximum einer Liste mit max/2
maxListe([X],X).
maxListe([X|R],Erg):-maxListe(R,ZwischErg),Erg is max(X,ZwischErg).
% Das Minimum einer Liste mit min/2
minListe([X],X).
minListe([X|R],Erg):-minListe(R,ZwischErg),Erg is min(X,ZwischErg).
%--------
%alternative Lösungen für max und min
%Um das Maximum einer Liste zu finden, sortiert man zuerst die Liste und
%dann gibt das letzte Element aus.
%maxListe(Liste,Max):-
%  sort(Liste,ZwischListe),
%  letzte_element(ZwischListe,Max).
%letzte_element([X],X).
%letzte_element([_|R],X):-letzte_element(R,X).
%Um das Minimum einer Liste zu finden, sortiert man zuerst die Liste und
%dann gibt das erste Element aus.
%minListe(Liste,Min):-
%  sort(Liste,ZwischListe),
%  erste_element(ZwischListe,Min).
%erste_element([X|_],X).
```

16.3) Geben Sie nun Listen der ältesten, bzw. jüngsten Person aus. D.h. sollten mehrere Personen das maximale Alter erreicht haben, so sollen alle in der Form

Person: xxx Alter:yy
Person: yyy Alter:zz

ausgegeben werden.

```
% Aufgabe 16.3
aelteste(Alter):-
        write('Die älteste Person ist '),nl,
        findall(Alter,person(_,_,alter(Alter)),ListeAlter),
        maxListe(ListeAlter,MaxAlter),
        findall((person:X,geschlecht:Y,alter:MaxAlter),person(X,Y,alter(MaxAlter)),Alter),!.
%?- aelteste(L).
%Das älteste Person ist
%L = [ (person:opa, alter:80), (person:peter, alter:80)].
juenste(WeibAlter):-
        write('Die jünste Person ist '),nl,
        findall(Alter,person(_,_,alter(Alter)),ListeAlter),
        minListe(ListeAlter,MaxAlter),
        findall((person:X,geschlecht:Y,alter:MaxAlter),person(X,Y,alter(MaxAlter)),WeibAlter),!.
%?- juenste(X).
%Die jüngste Person ist
%X = [ (person:isabela, geschlecht:weiblich, alter:23)].
```

Aufgabe 17

Schreiben Sie ein Programm, das eine Liste L1 als Eingabe akzeptiert und als Ausgabe L2 eine Liste von Listen hat, wo gleiche aufeinanderfolgende Einträge jeweils als Teilliste zusammengefasst sind.
Liste [1,2,2,3,a,a,a,b] soll also als [[1],[2,2],[3],[a,a,a][b]] ausgegeben werden.

```
%Aufgabe17 Alternative Lösung auch für unsortierte Listen
list([],[]).
list(X,Y):- msort(X,XSorted), orderedlists(XSorted,Y).
orderedlists([],[]).
orderedlists([X|Xs],L):-orderedlists(Xs,L1),
                        add(X,L1,L),!.
add(X,[],[[X]]).
add(X,[Y|Ys],L):-member(X,Y),!,append([X],Y,L1),L=[L1|Ys].% mit cut
```

```
add(X,[Y|Ys],L):-L=[[X],Y|Ys].

%add(X,[Y|Ys],L):-member(X,Y),append([X],Y,L1),L=[L1|Ys].%ohne cut
%add(X,[Y|Ys],L):-not(member(X,Y)),L=[[X],Y|Ys].
%?- list([1,2,2,3,a,a,a,b],L).
%L = [[1], [2, 2], [3], [a, a, a], [b]].

%?- list([2,2,a,a,2,2a,a],L).
%L=[[2,2,2,2][a,a,a,a]].


%Aufgabe17 Elegante aber undurchsichtige Lösung für sortierte Listen

zusammen([],[]).
zusammen([X],[[X]]).
zusammen([X,X|Rest],[[X|XListe]|Restliste]):- !, zusammen([X|Rest],[XListe|Restliste]).
zusammen([X,Y|Rest],[[X],YListe|Restliste]):- zusammen([Y|Rest],[YListe|Restliste]).
```

Aufgabe 18

Schreiben Sie ein Prädikat ungerade/1, das yes liefert, wenn eine Liste eine ungerade Zahl von Elementen enthält.

```
%Aufgabe18
ungerade([_]).
ungerade([_,_|X]):-ungerade(X).
%alternative Lösung
ungerade([_|L]):-gerade(L).
gerade([_,_|R]):-gerade(R).
gerade([ ]).
```