A10: 13P


```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Aufgabenblatt 01 - SE3-LP WiSe 16/17
%
% Finn-Lasse J rgensen 6700628 4joergen@informatik.uni-hamburg.de
% Fabian Behrendt 6534523 fabian.behrendt95@gmail.com
% Daniel Klotzsche 6535732 daniel_klotzsche@hotmail.de
%
% Wir sind bereit folgende Aufgaben zu pr sentieren:
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Vollst ndige Ausgabe von Listen aktivieren
?- set_prolog_fla(answer_write_optio, [quoted(true), spacing(next_argument)]
).
```

A1: 8P

```
%%% A1 %%%

%% 1.1 Betrachten Sie die Dateien zum Feinstaub in der Habichtstra e (Station 6
8HB)
%% Anfang 2015, Ende 2015 und Anfang 2016 (2015-w01, 2015-w50, 2016-w01).
%% Interpretieren Sie die in den CSV-Dateien enthaltene Information und stellen
%% Sie die Verkn pfung her zu den Inhalten der entsprechenden Prolog-Dateien
%% (XY-facts.pl).

% In den CSV-Dateien findet man Informationen zu der Menge an Feinstaub an eine
r
% bestimmten Station (in unserem Fall "Habichtstra e").
% Dabei gibt die zweite Zeile die Art des Feinstaubs an (PM10 und/oder PM2,5),
die
% dritte Zeile die Einheit, in der gemessen wurde, und die vierte Zeile die Mes
szeit.
% Alle nachfolgenden Werte geben den Feinstaubanteil zu einem bestimmten Zeitpu
nkt
% in dem Format "TT.MM.JJJJ HH:MM Anteil" an. Gegebenenfalls kommt noch ein zwe
iter Wert
% f r den Anteil hinzu, falls sowohl PM10 als auch PM2,5 gemessen wurde.
%
% In den "XY-facts.pl"-Dateien gibt es Fakten (pollution_data), welche die Date
n in
% folgender Reihenfolge halten:
% Station, Komponente(n), Messzeit (24hg f r 24-Stundenwerte (gleitend)),
% UNIX-Timestamp vom ersten Tag in der Datei, Abstand der Messungen in Sekunden
,
% Liste mit den gemessenen Daten.

%% 1.2 Implementieren Sie ein Pr dikat, das eine CSV-Datei in das Format  berf
hrt,
%% das in 68HB-2015-w01-facts.pl deklariert ist und verwendet wird.

% Laden des Prolog-Moduls zur Verarbeitung von CSV
:- use_modul(library(csv)).
% Um Encoding-Probleme zu vermeiden
:- set_prolog_fla(encodin, utf8)
% Um Daten aus dem Internet herunterzuladen
:- use_modul(library(http/http_clie)).

% Sollte es sich um eine zweispaltige CSV handeln, so greift diese Regel.
% Nach dem Einlesen der CSV holt sich diese Regel die richtigen Werte aus diese
r
% und unifiziert sie mit den Variablen, die dann weitergegeben werden.
%
% csv_to_pl(+CSV)
csv_to_pl(CSV) :-
  csv_read_fil(CSV, Rows, [match_arity(false),convert(true)]),
```



```
  Rows = [StationRaw, KomponenteRaw, _, _, _ | Werte],
  StationRaw= row('Station', Station),
  % Fallunterscheidung f r PM10 oder PM2,5
  (KomponenteRaw= row('Komponente', Komponente);
  KomponenteRaw= row('Komponente', KomponenteA, KomponenteB),
  concat(KomponenteA, ',', KomponenteTemp),
  concat(KomponenteTemp, KomponenteB, Komponente)),
  Messzeit= '24hg',
  create_data_zweistellig(Werte, DatumR, DatenR),
  reverse(DatumR, DatumAll),
  DatumAll= [Datum | _],
  date_to_timestamp(Datum, Timestamp),
  Abstand= 3600,
  reverse(DatenR, Daten),
  csv_to_pl_zweistellig(Station, Komponente, Messzeit, Timestamp, Abstand, Date
n).

% Sollte es sich um eine dreispaltige CSV handeln, so greift diese Regel.
% Nach dem Einlesen der CSV holt sich diese Regel die richtigen Werte aus diese
r
% und unifiziert sie mit den Variablen, die dann weitergegeben werden.
%
% csv_to_pl(+CSV)
csv_to_pl(CSV) :-
  csv_read_fil(CSV, Rows, [match_arity(false),convert(true)]),
  Rows = [StationRaw, _, _, _, _ | Werte],
  StationRaw= row('Station', Station1, Station2),
  Messzeit= '24hg',
  create_data_dreistellig(Werte, DatumR, Daten1R, Daten2R),
  reverse(DatumR, DatumAll),
  DatumAll= [Datum | _],
  date_to_timestamp(Datum, Timestamp),
  Abstand= 3600,
  reverse(Daten1R, Daten1),
  reverse(Daten2R, Daten2),
  csv_to_pl_dreistellig(Station1, Station2, 'PM10', 'PM2,5', Messzeit, Messzeit
,
            Timestamp, Timestamp, Abstand, Abstand, Daten1, Daten2).

% Zur Umwandlung des gegebenen Formats "DD.MM.YYYY HH:MM" in iso_8601 "YYYYMMDD
THHMM00+0000"
% durch Zerlegen des Formats in seine einzelnen Atome, um diese anschlie end in
 der richtigen
% Reihenfolge wieder zusammenzuf gen
%
% date_to_timestamp(+Date, -Timestamp)
date_to_timestamp(Date, Timestamp) :-
  atom_chars(Date, List),
  List = [D1, D2, _, M1, M2, _, Y1, Y2, Y3, Y4, _, H1, H2, _, Min1, Min2],
  concat(Y1, Y2, Year01),
  concat(Y3, Y4, Year02),
  concat(Year01, Year02, Year),
  concat(M1, M2, Month),
  concat(D1, D2, Day),
  concat(H1, H2, Hour),
  concat(Min1, Min2, Minute),
  concat(Year, Month, YM),
  concat(YM, Day, YMD),
  concat(YMD, 'T', YMDT),
  concat(YMDT, Hour, YMDTH),
  concat(YMDTH, Minute, YMDTHM),
  concat(YMDTHM, '00+0000', ISO),
  parse_time(ISO, iso_8601,TimestampTemp),
  Timestamp is round(TimestampTemp).
```

(5) Behrendt, Jörgensen, Klotzsche

```
% Erstellt zwei Listen, wobei List1 die Daten und List2 die Messwerte enth lt.
%
% create_data_zweistellig(+ListeDerDatenUndWerte, -List1, -List2)
create_data_zweistellig([], [], []).
create_data_zweistellig([row(Date, Value) | Tail], List1, List2) :-
  append([Date], ListNew1, List1),
  append([Value], ListNew2, List2),
  create_data_zweistellig(Tail, ListNew1, ListNew2).

% Erstellt drei Listen, wobei List1 die Daten, List2 die ersten Messwerte und L
ist3
% die zweiten Messwerte enth lt.
%
% create_data_dreistellig(+ListeDerDatenUndWerte, -List1, -List2, -List3)
create_data_dreistellig([], [], [], []).
create_data_dreistellig([row(Date, Value1, Value2) | Tail], List1, List2, List3
) :-
  append([Date], ListNew1, List1),
  append([Value1], ListNew2, List2),
  append([Value2], ListNew3, List3),
  create_data_dreistellig(Tail, ListNew1, ListNew2, ListNew3).

% Schreibt den finalen Ausdruck in die Datenbasis und zeigt ihn danach an.
%
% csv_to_pl_zweistellig(+Station, +Komponente, +Messzeit, ...)
csv_to_pl_zweistellig(Station, Komponente, Messzeit, Timestamp, Abstand, Daten)
 :-
  assertz(my_pollution_data(Station, Komponente, Messzeit, Timestamp, Abstand,
Daten)),
  listing(my_pollution_data).

% Schreibt die beiden finalen Ausdr cke in die Datenbasis und zeigt diese danac
h an.
%
% csv_to_pl_dreistellig(+Station1, +Station2, +Komponente1, ...)
csv_to_pl_dreistellig(Station1, Station2, Komponente1, Komponente2, Messzeit1,
Messzeit2,
            Timestamp1, Timestamp2, Abstand1, Abstand2, Daten1, Daten2) :-
  assertz(my_pollution_data(Station1, Komponente1, Messzeit1, Timestamp1, Absta
nd1, Daten1)),
  assertz(my_pollution_data(Station2, Komponente2, Messzeit2, Timestamp2, Absta
nd2, Daten2)),
  listing(my_pollution_data).

% Tests %

% ?- csv_to_pl("68HB-2016-w01.csv").
% :- dynamic my_pollution_data/6.
%
% my_pollution_data('Habichtstra e', 'PM10', '24hg', 1451865600, 3600, [36, 37,
 37, 38, 39, 39, 40, 41, 41, 42, 42, 43, 43, 44, 44, 45, 45, 46, 47, 47, 48, 48
, 49, 49, 49, 49, 50, 50, 50, 51, 51, 52, 52, 53, 53, 54, 54, 54, 54, 54, 54, 5
3, 53, 53, 52, 52, 51, 50, 50, 50, 50, 50, 49, 50, 50, 50, 50, 49, 49, 48, 48,
47, 47, 47, 47, 47, 46, 46, 46, 46, 46, 47, 48, 49, 50, 51, 53, 54, 55, 56, 57,
 59, 61, 63, 66, 69, 72, 74, 77, 79, 81, 83, 86, 88, 90, 91, 91, 91, 90, 89, 87
, 84, 81, 78, 74, 71, 67, 63, 59, 54, 50, 46, 43, 40, 37, 34, 31, 28, 26, 24, 2
2, 20, 19, 17, 16, 16, 17, 17, 18, 18, 19, 19, 20, 20, 20, 20, 20, 20, 20, 20,
21, 21, 21, 21, 21, 21, 22, 22, 23, 23, 23, 24, 24, 25, 25, 26, 26, 26, 26, 27,
 27, 27, 27, 27, 26, 26, 26, 25, 25]).
% my_pollution_data('Habichtstra e', 'PM2,5', '24hg', 1451865600, 3600, [31, 32
, 33, 34, 34, 35, 36, 36, 37, 38, 38, 39, 39, 40, 40, 41, 41, 42, 42, 42, 43, 4
3, 43, 44, 44, 44, 44, 44, 45, 45, 45, 46, 46, 46, 46, 47, 47, 47, 47, 47, 47,
46, 46, 46, 46, 45, 45, 44, 44, 44, 44, 44, 44, 44, 44, 44, 44, 44, 43, 43, 42,
 42, 41, 41, 41, 40, 40, 39, 39, 39, 39, 39, 40, 40, 41, 41, 42, 43, 44, 45, 46
, 47, 49, 52, 54, 57, 59, 61, 64, 66, 68, 70, 72, 74, 76, 77, 78, 78, 78, 77, 7
```



```
6, 73, 70, 67, 64, 61, 57, 54, 50, 46, 42, 39, 36, 33, 31, 28, 25, 23, 20, 18,
16, 14, 13, 11, 10, 10, 10, 10, 10, 11, 11, 11, 12, 12, 12, 12, 12, 11, 11, 11,
 12, 12, 12, 12, 12, 12, 13, 13, 14, 14, 14, 15, 16, 16, 17, 17, 18, 18, 19, 19
, 19, 19, 19, 19, 19, 19, 19, 19, 18]).
%
% true.

%% 1.Bonus Erstellen Sie ein Pr dikat, welches die Daten direkt aus dem
%% Internet herunterl dt. Nutzen Sie hierf r das http_client-Modul.

% Erzeugt eine CSV-Datei, in der die Messwerte der Kieler Stra e innerhalb eine
s
% bestimmten Zeitraums gespeichert werden.
% Angabe von Start und Ende in 'TT.MM.JJJJ'
%
% http_to_csv(+Start, +Ende)
http_to_csv(Start, Ende) :-
  % Formatieren der URL
  URL_Anfang = 'http://hamburg.luftmessnetz.de/station/64KS.csv?componentgroup=
pollution&componentperiod=24hg&searchperiod=custom&searchfrom=',
  URL_Ende = '&searchuntil=',
  concat(URL_Anfang, Start, URL_Temp1),
  concat(URL_Temp1, '+00:00', URL_Temp2),
  concat(URL_Temp2, URL_Ende, URL_Temp3),
  concat(URL_Temp3, Ende, URL_Temp4),
  concat(URL_Temp4, '+00:00', URL),
  % Formatieren des Dateinamens
  FILE_Name = 'http_to_csv_',
  get_time(FILE_TimestampTemp),
  FILE_Timestamp is floor(FILE_TimestampTemp),
  concat(FILE_Name, FILE_Timestamp, FILE_NameTemp),
  concat(FILE_NameTemp, '.csv', FILE),
  % CSV-Datei laden
  http_get(URL, Data, []),
  % Datei zum Schreiben  ffnen bzw. neu erstellen
  open(FILE, write, Stream),
  write(Stream, Data),
  close(Stream).

% Tests %

% ?- http_to_csv('01.01.2017', '02.01.2017').
% true.
% [Datei wurde erzeugt und liegt im Verzeichnis, aus dem die *.pl-Datei ge ffne
t wurde.]
```

+2P

A2: 0P

```
%%% A2 %%%
```

A3: 5P

```
%%% A3 %%%
:-['68HB-2016-w01-facts.pl'].

korrelationskoeffzient(Zeitreihe1,Zeitreihe2,E):-
  length(Zeitreihe1,Laenge),
  N is Laenge,
  sum_list(Zeitreihe1,Xi),
  sum_list(Zeitreihe2,Yi),
  multipliziereListen(Zeitreihe1,Zeitreihe2,Zwischenliste),
  sum_list(Zwischenliste,Zwischenergebnis),
  quadriereListen(Zeitreihe1,XiQuadrat),
  quadriereListen(Zeitreihe2,YiQuadrat),
  sum_list(XiQuadrat,QuadrierteListe1),
  sum_list(YiQuadrat,QuadrierteListe2),
  E is ((N * Zwischenergebnis) - (Xi * Yi))/(sqrt(N * QuadrierteListe1 - (Xi^2)) *
 sqrt(N * QuadrierteListe2 - (Yi^2))).
```

```
% Dies ist ein Hilfspr dikat, welches die Elemente von zwei Listen miteinander
multipliziert.
multipliziereListen([],[],[]).
multipliziereListen([Ergebnis1| Restliste1],[Ergebnis2| Restliste2],[Ergebnis
 | Restliste]):-
  Ergebnis is Ergebnis1* Ergebnis2,
  multipliziereListen(Restliste1,Restliste2,Restliste).

% Dies ist ein Hilfspr dikat, welches die Elemente von zwei Listen miteinander
quadriert.
quadriereListen([],[]).
quadriereListen([Ergebnis1|Rest1],[Ergebnis2|Rest2]):-
  Ergebnis2 is Ergebnis1*Ergebnis1,
  quadriereListen(Rest1,Rest2).

%Test
%?- korrelationskoeffzient([1,2,3],[1,2,3],E).
%E = 1.0000000000000002.
%?- korrelationskoeffzient([1,2,3],[2,3,8],E).
%E = 0.9332565252573828.
%?- korrelationskoeffzient([1,2,3],[2,6000,8],E).
%E = 0.0008667473679564494.
%?- korrelationskoeffzient([1,2,3],[10,3,5],E).
%E = -0.6933752452815364.

%pollution_data(_,'PM10',_,_,_,L1), pollution_data(_,'PM2,5',_,_,_,L2),korrelat
ionskoeffzient(L1,L2,E).
%L1 = [36, 37, 37, 38, 39, 39, 40, 41, 41, 42, 42, 43, 43, 44, 44, 45, 45, 46,
47, 47, 48, 48, 49, 49, 49, 49,
%50, 50, 50, 51, 51, 52, 52, 53, 53, 54, 54, 54, 54, 54, 54, 53, 53, 53, 52, 52
, 51, 50, 50, 50, 50, 50,
% 49, 50, 50, 50, 50, 49, 49, 48, 48, 47, 47, 47, 47, 47, 46, 46, 46, 46, 46, 4
7, 48, 49, 50, 51, 53, 54,
% 55, 56, 57, 59, 61, 63, 66, 69, 72, 74, 77, 79, 81, 83, 86, 88, 90, 91, 91, 9
1, 90, 89, 87, 84, 81, 78,
% 74, 71, 67, 63, 59, 54, 50, 46, 43, 40, 37, 34, 31, 28, 26, 24, 22, 20, 19, 1
7, 16, 16, 17, 17, 18, 18,
% 19, 19, 20, 20, 20, 20, 20, 20, 20, 20, 21, 21, 21, 21, 21, 21, 22, 22, 23, 2
3, 23, 24, 24, 25, 25, 26,
% 26, 26, 26, 27, 27, 27, 27, 27, 26, 26, 26, 25, 25],
%L2 = [31, 32, 33, 34, 34, 35, 36, 36, 37, 38, 38, 39, 39, 40, 40, 41, 41, 42,
42, 42, 43, 43, 43, 44,
%44, 44, 44, 44, 45, 45, 45, 46, 46, 46, 46, 47, 47, 47, 47, 47, 47, 46, 46, 46
, 46, 45, 45, 44, 44,
%44, 44, 44, 44, 44, 44, 44, 44, 44, 43, 43, 42, 42, 41, 41, 41, 40, 40, 39, 39
, 39, 39, 39, 40, 40,
%41, 41, 42, 43, 44, 45, 46, 47, 49, 52, 54, 57, 59, 61, 64, 66, 68, 70, 72, 74
, 76, 77, 78, 78, 78, 77,
%76, 73, 70, 67, 64, 61, 57, 54, 50, 46, 42, 39, 36, 33, 31, 28, 25, 23, 20, 18
, 16, 14, 13, 11, 10, 10,
%10, 10, 10, 11, 11, 11, 12, 12, 12, 12, 12, 11, 11, 11, 12, 12, 12, 12, 12, 12
, 13, 13, 14, 14, 14, 15,
%16, 16, 17, 17, 18, 18, 19, 19, 19, 19, 19, 19, 19, 19, 19, 19, 18],
%E = 0.9925835437213651.
```