МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИМЕНИ АКАДЕМИКА С.П. КОРОЛЕВА»

ФАКУЛЬТЕТ ИНФОРМАТИКИ

кафедра программных систем

по лабораторному практикуму по дисциплине

«Логическое программирование»

Лабораторная работа № 3

Вариант №9

Выполнила:

Гижевская В.Д.

гр. 6313

Проверил:

Оплачко Д.С.

Самара 2020

Задание

Предметная область – расписание движения самолётов. Каждый рейс может быть описан структурой: название авиакомпании, номер рейса, пункт отлёта, пункт прилёта, время отлёта (часы), время прилёта (часы), список пунктов промежуточных посадок, список тарифов. Тариф может быть описан структурой: тип класса салона, цена. Реализовать следующие типы запросов:

1. Найти авиакомпанию, у которой максимальное число тарифов на заданный маршрут;

2. Подсчитать число рейсов, улетающих из заданного города до указанного времени;

3. Найти все авиакомпании, у которых есть билеты класса люкс;

4. Найти все рейсы, у которых цена билета ниже заданной;

5. Найти все авиакомпании, у которых время полёта меньше заданного. Листинг программы

domains

list\_station=symbol\*

list\_tariff=tariff\*

list=integer\*

name\_company, station\_start, station\_finish, type\_class = symbol

time\_start, time\_finish, price =real

number\_flight=integer

tariff= tar(type\_class, price)

facts

dbfind1(name\_company,station\_start)

dq2(station\_start,time\_start,integer)

dq3(type\_class, name\_company)

dq4(price, number\_flight)

dq5(name\_company)

nondeterm n(integer)

max(integer,symbol)

predicates

%count(list\_tariff,integer)

%nondeterm q1(station\_start,station\_finish)

% nondeterm q2 (station\_start,time\_start,integer)

nondeterm flight (name\_company, number\_flight, station\_start, station\_finish, time\_start, time\_finish, list\_station, list\_tariff)

%q3(type\_class)

%nondeterm q31(type\_class,list\_tariff)

%nondeterm count\_flight(symbol,real,integer)

%nondeterm length(list\_tariff,integer)

%nondeterm q4 (price)

%nondeterm chislo\_number\_flight(string,real,integer)

%nondeterm chislo\_number\_flight(symbol,time\_start,integer)

nondeterm length(list,integer)

%q5(real)

%nondeterm number\_flight\_by\_time(name\_company, number\_flight, station\_start, station\_finish, time\_start, time\_finish, list\_station, list\_tariff)

nondeterm count(symbol,real).

nondeterm count(symbol,real,integer).

clauses

/\*length([], 0).

length([\_|T], L) :-

length(T, L\_T),

L = L\_T + 1. \*/

/\*count([],0).

count([Head|Tail],N):-count(Tail,N1), N=N1+1.\*/

length([],0).

length([\_|Tail],N):-length(Tail,N1),N=N1+1.

flight ("First aircompany",1,"Samara","Erevan",18.00,19.30,[],[tar(econom,3000),tar(lux,8000),tar(business,4000)]).

flight ("Second aircompany", 4, "Samara", "Erevan", 5.30, 8.30, [], [tar(econom,5000), tar(business,7500)]).

flight ("Second aircompany", 74, "Samara", "San Francisco", 15.30, 22.00, ["Moscow", "London"], [tar(econom,15000),tar(business,17500)]).

flight ("Third aircompany", 56, "Samara", "Dubai", 3.30, 8.20, ["Moscow", "Minsk"], [tar(lux,15000),tar(econom,10000),tar(business,12000)]).

flight ("First aircompany", 8, "Samara", "Oslo", 17.30, 23.10, ["Moscow", "Paris"], [tar(econom,5000),tar(business,7500)]).

flight ("First aircompany", 87, "Samara", "Milan", 17.30, 23.10, ["Moscow", "Paris"], [tar(econom,5000)]).

flight ("Fifth aircompany", 666, "Samara", "Rome", 15.30, 22.00, ["Moscow", "London"], [tar(econom,15000),tar(lux,25000),tar(business,17500)]).

%1

/\*max(0,a).

q1(Station\_start,Station\_finish) :- flight (Name,\_,Station\_start,Station\_finish,\_,\_,\_,L),

count(L,N),

max(N1,Name1),

N>N1,

retract(max(N1,Name1)),

assert(max(N,Name)),

fail.

goal

q1("Samara","Rome");

max(A,B). \*/

%2

count(Station\_start,Time):-flight(\_,\_,Station\_start,\_,Time\_start,\_,\_,\_),

Time\_start<Time,

n(J),

retract(n(J)),

J1 = J+1,

assert(n(J1)),

fail.

count(\_,\_).

count(Station\_start,Time,N):-retractall(n(\_)),

assert(n(0)),

count(Station\_start,Time),

n(N).

%3

/\*q3(P):-dq3(P,A),

write("\*",P," ",A),

nl,

fail.

q3(P):-not(dq3(P,\_)),

flight (N,\_,\_,\_,\_,\_,\_,L),

q31(P,L),

write(N),

nl,

assert(dq3(P,N)),

fail.

q31(H,[H1|\_]):-H1=tar(H,\_).

q31(H,[\_|T]):-q31(H,T).

goal

q3(lux), q3(lux).\*/

%4

/\*q4(F):-dq4(F,A),

write("\*",F," ",A),

nl.

q4(F):-not(dq4(F,\_)),

flight(\_,A,\_,\_,\_,\_,\_,[tar(\_,Price)]),

abs(Price)<F,

write(A),

nl,

assert(dq4(Price,A)).

goal

q4(8000),q4(8000).

\*/

%5

/\*q5(\_):-dq5(Nameavia),

write('\*', Nameavia),

nl,

fail.

q5(Time):-not(dq5(\_)),

flight (Name,\_,\_,\_,Time1, Time2,\_,\_),

abs(Time2-Time1)<Time,

write(Name),

nl,

assert(dq5(Name)),

fail.

goal

q5(4.00), q5(4.00).\*/

goal

count("Samara",17.00,N).

Результат работы

На рисунке представлен пример работы программы: число рейсов, улетающих из Самары до 17 часов.

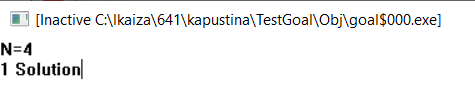


Рисунок 1 – Результат работы программы.