**Лабораторна робота № 2**

**Тема:** Пошук з поверненням. Керування пошуком з поверненням: предикати fail та відтинання

**Мета:** ознайомитися з організацією та керуванням пошуку з поверненням, предикатами fail та відтинанням.

**Короткі теоретичні відомості**

Пошук з поверненням (backtracking) – це один з основних прийомів пошуку розв’язків поставленої задачі мовою Пролог. Виконуючи пошук, Пролог може зустрітися з необхідністю вибору між альтернативними шляхами. Тоді він ставить маркер коло місця розвилки (точка відкату) і вибирає першу підмету. Якщо вона не виконується, то Пролог повертається у точку відкату і переходить до наступної підмети. Середовище Visual Prolog дозволяє використовувати настроювальника для покрокового виконання програми. Настроювальник працює з відкомпільованим кодом. У вихідному коді можна ставити точки зупинки і21 виконувати програму по кроках. У режимі покрокового виконання програми можна переглядати значення змінних і вміст затверджених фактів. fail – це тотожно-хибний предикат, що штучно створює ситуацію неуспіху. Після виконання цього предикату керування передається в точку відкату і пошук продовжується. Використання предикату fail дозволяє знайти всі розв’язки задачі. Щоб обмежити простір пошуку та припинити пошук розв’язків при виконанні якої-небудь умови, використовується предикат відсічення (позначається !). Один раз перейшовши через відсічення, неможливо повернутися назад, оскільки цей предикат є тотожно-істинним. Процес може тільки перейти до наступної підмети, якщо така є. Наприклад, p :- p1, p2, !? p3. Якщо досягнуті цілі p1 і p2, то повернення до них для пошуку нових розв’язків є неможливим.

**Хід роботи**

**Завдання 1**

**1.** База даних містить такі факти:

clauses  
interested("Коля","гитара").  
interested("Оля","скрипка").  
interested("Дима","плаванье").  
interested("Таня","тенис").  
sport("плаванье").  
sport("тенис").  
mus\_instr("скрипка").  
mus\_instr("гитара").

а) створити правило sportsman (спортсмен) і визначити, хто захоплюється спортом;

**2.** База даних містить такі факти:

interested("Коля","гитара").  
interested("Оля","скрипка").  
interested("Дима","плаванье").  
interested("Таня","тенис").

а) створити правило musician (музикант) і визначити, хто захоплюється спортом;

**Завдання 2**

1. База даних містить факти виду: rest(name, city), ukraine(city), russia(city), woman(name), man(name).

clauses  
rest("Катя","Харьков").  
rest("Анастасия","Киев").  
rest("Алексей" ,"Днепропетровск").  
rest("Руслан","Хмельницк").  
rest("Владислав","Москва").  
rest("Святослав","Екатеринбург").  
rest("Мария","Оренбург").  
rest("Константин" ,"Уфа").  
rest("Юля","Полтава").  
ukraine("Харьков").  
ukraine("Полтава").  
ukraine("Киев").  
ukraine("Хмельницк").  
ukraine("Днепропетровск").  
russia("Москва").  
russia("Екатеринбург").  
russia("Оренбург").  
russia("Уфа").  
woman("Катя").  
woman("Юля").  
woman("Мария").  
woman("Анастасия").  
man("Алексей").  
man("Руслан").  
man("Владислав").  
man("Святослав").  
man("Константин").

а) вивести список жінок, що відпочивають у Росії;

б) вивести список чоловіків, що відпочивають в Україні;

**2.** База даних містить факти виду: book(author, title, publishing\_house, year), ukraine(city).

clauses  
book("Мастер и Маргарита" , "Михаил Булгаков", "Астрель" , 2010).  
book("Остров Сахалин", "Антон Чехов", "Питер", 1996).  
book("Вишневый сад","Антон Чехов","Питер" , 1998).  
book("Дама с собачкой","Антон Чехов", "Питер", 2001).  
book("Дубровский","Александр Пушкин", "Питер", 1995).  
book("Руслан и Людмила","Александр Пушкин","Питер", 2002).  
title("Дубровский").  
 title("Мастер и Маргарита").  
 title("Остров Сахалин").  
 title("Дама с собачкой").  
 title("Руслан и Людмила").  
 publishing\_house("Питер").  
 publishing\_house("Астрель").  
  year(2010).  
 year(1990).  
  year(1996).  
  year(2001).  
    year(1995).  
        year(2002).  
        autor("Александр Пушкин").  
    autor("Антон Чехов").  
    autor("Михаил Булгаков").

а) вивести весь список книг;

б) вивести список книг авторів Пушкіна та Чехова;

в) вивести список книг, виданих у видавництві «Пітер», не раніше 2000 року.

**3.** Скласти програму, що реалізує авіадовідник.

У довіднику міститься така інформація про кожен рейс: номер рейсу, час вильоту, дні (щодня, парні, непарні).

clauses  
reic(1, 12, "каждый день" ).  
reic(2, 10, "парные" ).  
reic(3, 13, "непарные ").  
reic(4, 9, "каждый день" ).  
reic(3, 10, "каждый день" ).  
reic(5, 6, "каждый день" ).  
num\_reic(1).  
num\_reic(2).  
num\_reic(3).  
num\_reic(4).  
num\_reic(5).  
time(6).  
time(12).  
time(10).  
time(13).  
time(9).  
date("каждый день").  
date("парные").  
date("непарные").

Вивести:

а) усю інформацію із довідника;

б) інформацію про літаки, що вилітають до заданого пункту по парних днях;

в) інформацію про літаки, що вилітають щодня не пізніше вказаного часу.

**4.** Скласти програму, що реалізує географічний довідник. У довіднику міститься така інформація про кожну країну: назва країни, назва столиці, чисельність населення, географічне положення (Європа чи Азія).

clauses  
country("Украина"," Киев", 3296100, "Европа").  
country("Франция","Париж", 9854000 , "Европа").  
country("Германия","Берлин",  3933300, "Европа").  
country("Италия", "Рим", 3550900, "Европа").  
country("Армения", "Ереван", 1462700 , "Азия").  
country("Вьетнам", "Ханой", 2543700, "Азия").  
country("Израиль","Иерусалим", 695500 , "Азия").  
country("Иран", "Тегеран", 7796257 , "Азия").  
name("Иран").  
name("Израиль").  
name("Вьетнам").  
name("Армения").  
name("Италия").  
name("Германия").  
name("Франция").  
name("Украина").  
capital("Тегеран").  
capital("Иерусалим").  
capital("Ереван").  
capital("Ханой").  
capital("Рим").  
capital("Берлин").  
capital("Киев").  
capital("Париж").  
people(3296100).  
people(9854000).  
people(3933300).  
people(3550900).  
people(1462700).  
people(2543700).  
people(695500).  
people(7796257 ).  
geodraph("Европа").  
geodraph("Азия").

Вивести:

а) усю інформацію із довідника;

б) інформацію про країни, чисельність населення яких перевищує задане значення;

в) інформацію про європейські країни, чисельність яких не перевищує задане значення.

**5.** Скласти програму, що реалізує словник. У словнику міститься така інформація: слово та його переклад (українські та англійські слова).

clauses  
slova("yellow", "жолтый").  
slova("black", "чёрный").  
slova("green", "зелёный").  
slova("blue" ,"синий").  
slova("white", "белый").  
slova("red", "красный").  
slovo("yellow").  
slovo("black").  
slovo("blue").  
slovo("white").  
slovo("green").  
translate("красный").  
translate("чёрный").  
translate("зелёный").  
translate("синий").  
translate("жолтый").  
translate("белый").

Реалізувати виведення всього словника, перекладу з української на англійську, з англійської на українську (з декількома значеннями).

**6.** Скласти програму, що реалізує телефонний довідник. У довіднику міститься наступна інформація про кожного абонента: ім’я та телефон.

clauses  
phone("Борис", 51212).  
phone("Володя", 81212).  
phone("Катя", 61142).  
phone("Вера", 71212).  
abonent("Борис").  
abonent("Володя").  
abonent("Катя").  
abonent("Вера").  
num(51212).  
num(81212).  
num(71212).  
num(61142).

Реалізувати виведення всієї інформації із довідника, пошук телефону за іменем, пошук імені за телефоном.

**7.** База даних містить факти виду pupil(name, class) і interested(name, hobby).

clauses  
pup("Вася", 2).  
pup("Люся", 3).  
pup("Коля", 2).  
pup("Неля", 5).  
pup("Юля", 3).  
interes("Юля", "квилинг").  
interes("Юля", "футбол").  
interes("Коля", "футбол").  
interes("Неля", "йога").  
interes("Вася", "футбол").  
interes("Люся", "фортопиано").  
child("Вася").  
child("Люся").  
child("Коля").  
child("Неля").  
child("Юля").  
clas(2).  
clas(3).  
clas(5).  
hobby("футбол").  
hobby("квилинг").  
hobby("йога").  
hobby("фортопиано").

Скласти програму, яка виводить:

а) список всіх учнів та їх захоплення;

б) підбирає одному з учнів вказаного класу, що захоплюється кіно, пару з інших класів. Вивести всі можливі пари.

**8.** База даних містить факти виду: pupil(name, class) і play(name, sport).

clauses  
pup("Вася", 2).  
pup("Люся", 3).  
pup("Коля", 2).  
pup("Неля", 5).  
pup("Юля", 3).  
interes("Юля", "квилинг").  
interes("Юля", "футбол").  
interes("Коля", "футбол").  
interes("Неля", "йога").  
interes("Вася", "футбол").  
interes("Люся", "фортопиано").  
child("Вася").  
child("Люся").  
child("Коля").  
child("Неля").  
child("Юля").  
clas(2).  
clas(3).  
clas(5).  
hobby("футбол").  
hobby("квилинг").  
hobby("йога").  
hobby("фортопиано").

Скласти програму, яка:

а) виводить список всіх учнів заданого класу і вид спорту, яким вони захоплюються;

б) підбирає одному з учнів вказаного класу, що грає у бадмінтон, пару з інших класів. Вивести всі можливі пари.

Код:

implement main  
     open core, console  
  
class facts - relatives  
pup:(string Child, integer Clas).  
interes:(string Child, string Hobby).  
  
child:(string).  
clas:(integer).  
hobby:(string).  
phone:(string Abonent, integer Num).  
abonent:(string).  
num:(integer).  
interested: (string Name, string Interes).  
sport: (string Name).  
mus\_instr: (string Name).  
rest:(string Name ,string Strana).  
ukraine:(string Name).  
russia:(string Name).  
woman:(string Name).  
man:(string Name).  
book:(string Title, string Autor, string Publishing\_house, integer Year).  
 title:(string Name).  
 publishing\_house:(string Name).  
year:(integer).  
autor:(string Name).  
reic:(integer Num\_reic, integer Time, string Date ).  
num\_reic:(integer).  
time:(integer).  
date:(string).  
country:(string Name, string Capital, integer People , string Geograph).  
name:(string).  
capital:(string).  
people:(integer).  
geodraph:(string).  
slova:(string Slovo, string Translate).  
slovo:(string).  
translate:(string).  
pupil:(string Name, integer Clas).  
 play:(string Name, string Sport).  
class predicates  
sportsman: (string Interested, string Sport) nondeterm anyflow.  
musician:(string Interested, string Mus\_instr) nondeterm anyflow.  
rus\_otd:(string Rest, string Russia) nondeterm anyflow.  
rus\_otd\_woman:(string Rus\_otd, string Woman) nondeterm anyflow.  
ukr\_otd:(string Rest, string Ukraine) nondeterm anyflow.  
ukr\_otd\_man:(string Rest, string Man) nondeterm anyflow.  
int:(string Child, integer Clas, string Hobby) nondeterm anyflow.  
int\_sp:(string Child, integer Clas, string Sport) nondeterm anyflow.  
clauses  
sportsman(X,Y):- interested(X,Y), sport(Y) .  
musician(X,Y):- interested(X,Y), mus\_instr(Y) .  
rus\_otd(X,Y):- rest(X,Y), russia(Y).  
rus\_otd\_woman(X,Y):- rus\_otd(X,Y), woman(X) .  
ukr\_otd(X,Y):- rest(X,Y), ukraine(Y).  
ukr\_otd\_man(X,Y):- ukr\_otd(X,Y), man(X) .  
int(X,Y,Z):-pup(X,Y), interes(X,Z).  
int\_sp(X,Y,Z):-pupil(X,Y), play(X,Z).  
run() :-  
    init(),  
file::consult("interested.txt", relatives),  
write("Задание 1. Хто захоплюється спортом?"), nl,  
write("Спортсмены: "), nl,  
sportsman(X,Y),  
write(X),nl,  
fail;  
write("Задание 1. Хто захоплюється музикою?"), nl,  
write("Музыканты: "), nl,  
musician(X,Y),  
write(X),nl,  
fail;  
file::consult("Отдых.txt", relatives),  
write("Женщины отдыхающие в Росии"), nl,  
rus\_otd\_woman(X,Y),  
write (X), nl,  
fail;  
write("Мужчины отдыхающие в Украине"), nl,  
ukr\_otd\_man(X,Y),  
write(X), nl,  
fail;  
file::consult("Книги.txt", relatives),  
write("Список книг"), nl,  
book(X, Y, Z, H),  
write(X), nl,  
fail;  
write("Список книг Пушкина"), nl,  
book(X, "Александр Пушкин", Z, H),  
write(X), nl,  
fail;  
write("Список книг Чехова"), nl,  
book(X, "Антон Чехов", Z, H),  
write(X), nl,  
fail;  
write("Список книг издательства Питер не раньше 2000 года"), nl,  
book(X,Y,"Питер", H), if H<2000 then  
write(X),nl  
end if,  
fail;  
file::consult("Авиасправочник.txt", relatives),  
write("Авиасправочник: "),nl,  
reic(X,Y,Z),nl,  
write("Номер рейса: ", X, " Вылет: ", Y, " Дни: ", Z),nl,  
fail;  
write("Вылет по парным дням: "),nl,  
reic(X,Y,"парные"),  
write("Номер рейса: ", X, " Вылет: ", Y),nl,  
fail;  
write("Вылет каждый день не позже 12: "),nl,  
reic(X,Y,"каждый день"), if Y<12 then  
write("Номер рейса: ", X, " Вылет: ", Y),nl  
end if,  
fail;  
file::consult("Геогсправочник.txt", relatives),  
write("Географический спарвочник"),nl,  
country(A,B,C,D), nl,  
write("Страна: ",A, " Столица: ", B, " Численость: ", C, " Положение: ", D),nl,  
fail;  
write("Страны количество населения в которых превышает  1600000"),nl,  
country(A,B,C,D), if C>1600000 then  
write("Страна: ",A, " Столица: ", B, " Численость: ", C, " Положение: ", D),nl  
end if,  
fail;  
write("Страны европы количество населения в которых превышает  3300000"),nl,  
country(A,B,C, "Европа"), if C>3500000 then  
write("Страна: ",A, " Столица: ", B, " Численость: ", C ),nl  
end if,  
fail;  
file::consult("Словарь.txt", relatives),  
write("Все слова в словаре"),nl,  
slova(X,Y),nl,  
write("Слово: ",X," Перевод: ",Y),nl,  
fail;  
write("Перевод слова жолтый"),nl,  
slova(X,"жолтый"),nl,  
write("Перевод: ",X),nl,  
fail;  
file::consult("Телефсправочник.txt", relatives),  
write("Телефонный справочник"),nl,  
phone(X,Y),nl,  
write("Имя:", X," Номер: " ,Y),nl,  
fail;  
write("Чей номер 51212"),nl,  
phone(X, 51212),nl,  
write("Номер:" ,X),nl,  
fail;  
write("Номер Веры"),nl,  
phone("Вера", Y),nl,  
write("Номер:" ,Y),nl,  
fail;  
file::consult("Увлечения.txt", relatives),  
write("Список учеников  и хобби"),nl,  
interes(X,Y),nl,  
write("Ученик: ", X, " Хобби:" ,Y),nl,  
fail;  
write(" Пары по хобби"),nl,  
if int(X, 3 ,Z), int(A, 2 ,C) and Z=C then  
write("C 3 класса: " ,X," С 2 класса:  ", A," Вид спорта: ",C),nl  
end if,  
fail;  
file::consult("sport.txt", relatives),  
write("Список учеников  и видов спорта 2 класса"),nl,  
int\_sp(X,2,Y),nl,  
write("Ученик: ", X, " Вид спорта:" ,Y),nl,  
fail;  
write("Пара учеников  и видов спорта 2 класса"),nl,  
if int\_sp(X, 3 ,Z), int\_sp(C, 5 ,B) and Z=B then  
write("C 2 класса: " ,X," С 3 класса:  ", C," Вид спорта: ",Z),nl  
end if,  
fail;  
 \_ = readLine().  
end implement main  
  
goal  
    console::run(main::run).

