**Лабораторна робота №1**

**Тема:** Ознайомлення з Visual Prolog

**Мета:** ознайомитися з інтерфейсом Visual Prolog, навчитися створювати проекти та запускати їх.

**Короткі теоретичні відомості**

Prolog є мовою, що основана на програмуванні логіки (PROgramming in LOGic). Замість детальних інструкцій, що визначають, як розв’язувати ту чи іншу задачу, програміст на мові Prolog приділяє основну увагу опису задачі. Інтерфейс Visual Prolog включає: головне меню, панель інструментів, вікно проекту. Якщо під час останнього використання системи Visual Prolog там був відкритий проект, то система автоматично знову відкриє цей проект.

**Хід роботи**

**Завдання 1**

1. Маємо базу даних, яка містить такі факти:

clauses  
likes("Анна", "яблоко").  
likes("Сергей", "банан").  
likes("Андрей", "яблоко").  
likes("Света", "шоколад").  
likes("Вова", "шоколад").  
likes("Анна", "шоколад").  
likes("Света", "апельсин").  
likes("Вова", "банан").  
Скласти програму, що визначає:

а) усіх, хто полюбляє банани (bananas);

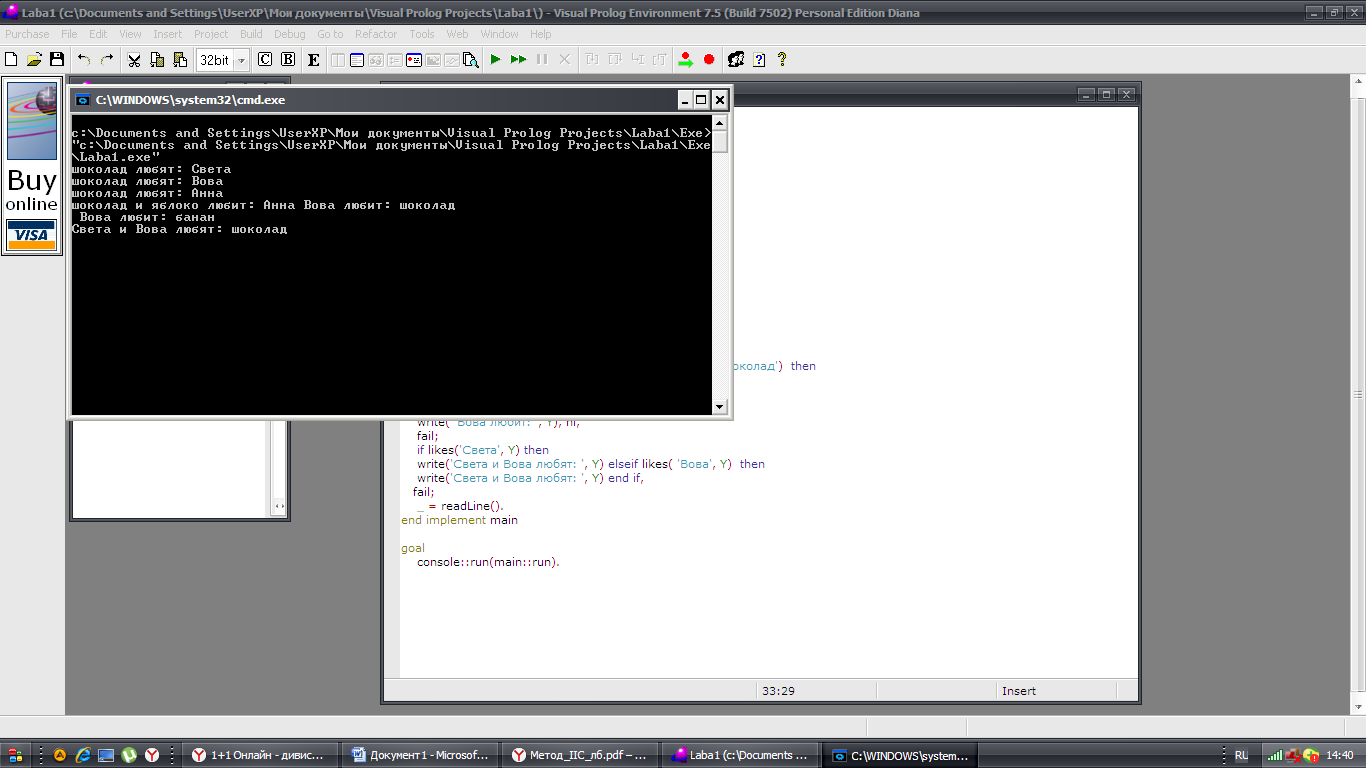
б) хто полюбляє і шоколад (chocolate), і яблука (apples);

в) що полюбляє Вова;

г) що полюбляють і Свєта, і Вова.

Код:

implement main  
    open core, console  
class facts - relatives  
likes : (string Person, string Fruit).  
fruit:(string Name).  
person:(string Names).  
clauses  
    run() :-  
    init(),  
    file::consult("like.txt", relatives),  
   likes(X,'шоколад'),  
    write('шоколад любят: ', X), nl,  
    fail;  
    if likes(X,'яблоко') then  
    write('шоколад и яблоко любит: ', X) elseif likes( X,'шоколад')  then  
    write('шоколад и яблоко любит: ', X) end if,  
   fail;  
   likes('Вова', Y),  
    write(' Вова любит: ', Y), nl,  
    fail;  
    if likes('Света', Y) then  
    write('Света и Вова любят: ', Y) elseif likes( 'Вова', Y)  then  
    write('Света и Вова любят: ', Y) end if,  
   fail;  
    \_ = readLine().  
end implement main  
  
goal  
    console::run(main::run).

****

1. Маємо базу даних, що містить такі факти:

clauses  
plays("Саша", "футбол").  
plays("Катя", "тенис").  
plays("Саша", "тенис").  
plays("Андрей", "футбол").  
plays("Олег", "футбол").  
plays("Ольга", "тенис").  
plays("Катя", "волейбол").  
plays("Олег", "волейбол").

Скласти програму, що визначає:

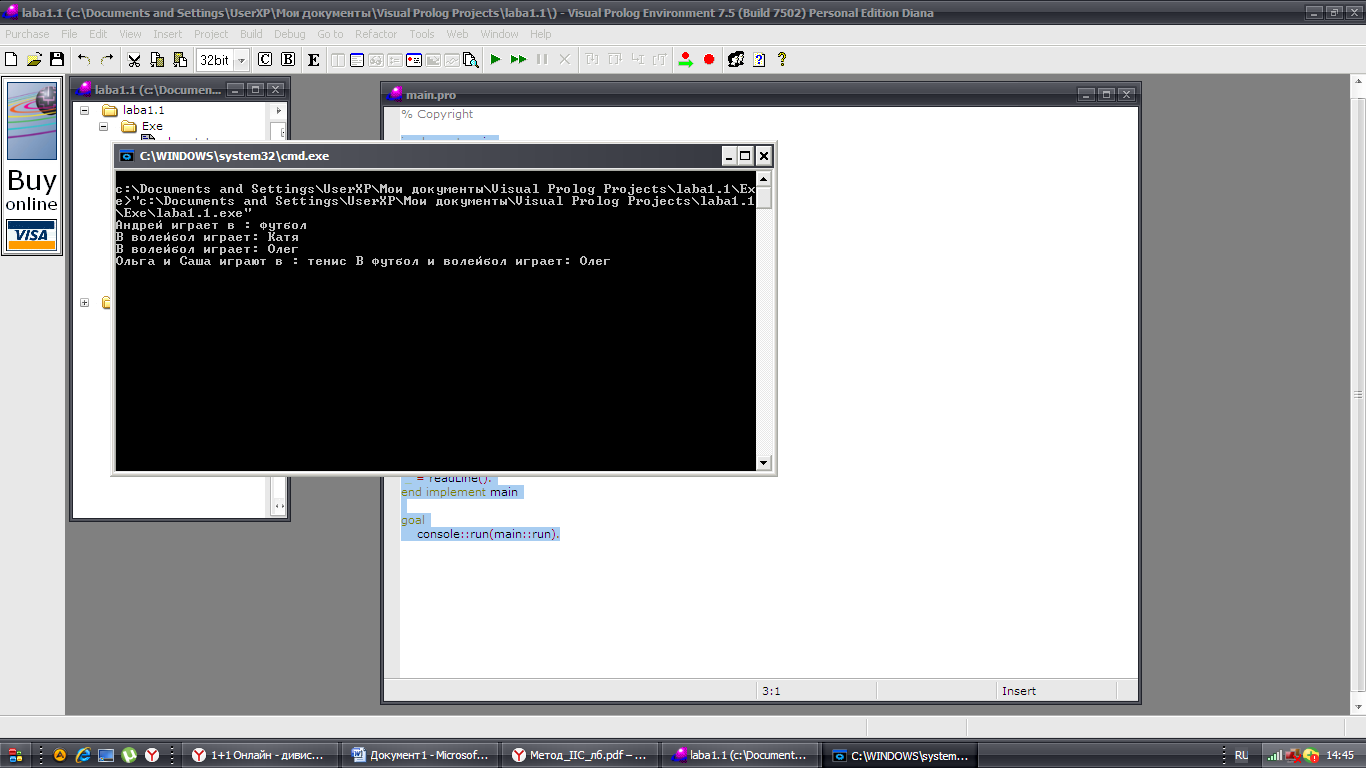
а) яким видом спорту захоплюється Андрій;

б) усіх, хто грає у волейбол;

в) яким видом спорту захоплюються і Ольга, і Саша;

г) хто захоплюється і футболом, і волейболом.

implement main  
    open core, console  
  
class facts - relatives  
plays : (string Person, string Geim).  
person: (string Name).  
geim: (string Names).  
clauses  
    run() :-  
    init(),  
    file::consult("plays.txt", relatives),  
     plays('Андрей', Y),  
    write('Андрей играет в : ', Y), nl,  
    fail;  
    plays(X, 'волейбол'),  
    write('В волейбол играет: ', X), nl,  
    fail;  
    plays('Ольга', Y),  if plays( 'Саша', Y)  then  
    write('Ольга и Саша играют в : ', Y) end if,  
   fail;  
    plays(X,'волейбол'), if plays( X,'футбол')  then  
    write(' В футбол и волейбол играет: ', X) end if,  
   fail;  
  
 \_ = readLine().  
end implement main  
  
goal  
    console::run(main::run).



1. Маємо базу даних, що містить такі факти:

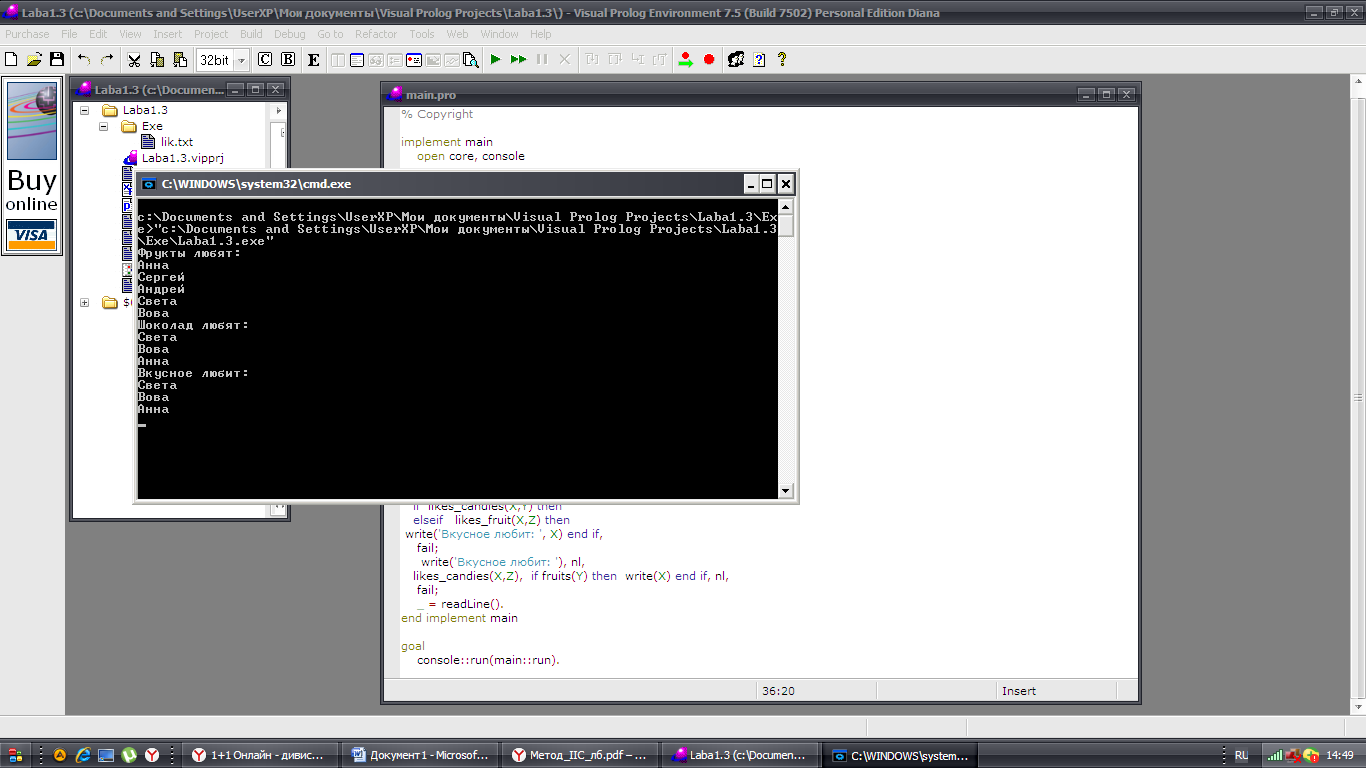
clauses  
likes("Анна", "яблоко").  
likes("Сергей", "банан").  
likes("Андрей", "яблоко").  
likes("Света", "шоколад").  
likes("Вова", "шоколад").  
likes("Анна", "шоколад").  
likes("Света", "апельсин").  
likes("Вова", "банан").

а) використовуючи наявні факти, скласти нове правило likes\_fruits(X) ) (полюбляє фрукти) і визначити всіх, хто полюбляє фрукти;

б) використовуючи наявні факти, скласти нове правило likes\_candies(X) ) (полюбляє цукерки) і визначити всіх, хто полюбляє шоколад;

в) використовуючи наявні факти, скласти нове правило likes\_tasty(X) (полюбляє смачне) і визначити всіх, хто полюбляє і фрукти, і цукерки;

implement main  
    open core, console  
class facts - relatives  
likes: (string Person, string Fruits).  
fruits:(string Name).  
person:(string).  
candies: (string).  
class predicates  
likes\_fruit: (string  Likes,  string Fruits) nondeterm (o,o).  
likes\_candies: (string  Likes,  string Candies) nondeterm (o,o).  
clauses  
likes\_fruit(X,Y):-likes(X,Y), fruits(Y).  
likes\_candies(X,Z):-likes(X,Z), candies(Z).  
     run() :-  
         init(),  
file::consult("lik.txt", relatives),  
write('Фрукты любят:'), nl,  
 likes\_fruit(X,Y),  
 write(X),nl,  
    fail;  
    write('Шоколад любят: '), nl,  
 likes\_candies(X,Z),  
 write(X),nl,  
   fail;  
   if  likes\_candies(X,Y) then  
   elseif   likes\_fruit(X,Z) then  
 write('Вкусное любит: ', X) end if,  
    fail;  
     write('Вкусное любит: '), nl,  
   likes\_candies(X,Z),  if fruits(Y) then  write(X) end if, nl,  
    fail;  
    \_ = readLine().  
end implement main  
  
goal  
    console::run(main::run).



1. Маємо базу даних, що містить такі факти:

clauses  
plays("Саша", "футбол").  
plays("Катя", "тенис").  
plays("Саша", "тенис").  
plays("Андрей", "футбол").  
plays("Олег", "футбол").  
plays("Ольга","тенис").  
plays('Катя", "волейбол").

а) використовуючи наявні факти, скласти нове правило volleyball\_woman(X) і визначити всіх жінок, що грають у волейбол.

б) використовуючи наявні факти, скласти нове правило volleyball\_man(X) і визначити всіх чоловіків, що грають у волейбол.

в) використовуючи наявні факти, скласти нове правило tennis\_pair(X,Y), що дозволяє знайти змішану тенісну пару (чоловік + жінка). Визначити всі такі пари.

implement main  
    open core, console  
class facts - relatives  
plays:(string Name, string Gaim).  
woman:(string).  
man:(string).  
class predicates  
plays\_woman:( string Name, string  Gaim) nondeterm anyflow.  
plays\_man:( string Name, string  Gaim) nondeterm anyflow.  
clauses  
plays\_woman(X,Y):-plays(X,Y),woman(X) .  
plays\_man(Z,Y):-plays(Z,Y),man(Z) .  
     run() :-  
         init(),  
file::consult("plays.txt", relatives),  
write('Кто из девушек играет в волейбол '), nl,  
if plays\_woman( X ,'волейбол') then  
write('Играет в волейбол ', X) end if, nl,  
fail;  
write('Кто из мужчин играет в волейбол '), nl,  
if plays\_man( Z ,'волейбол') then  
write('Играет в волейбол ',Z) end if, nl,  
fail;  
if plays\_man (Z,'тенис') and  plays\_woman ( X ,'тенис') then  
write('Пара по тенису ',Z,',', X ) end if, nl,  
fail;  
if plays\_man (Z,'волейбол') and  plays\_woman ( X ,'волейбол') then  
write('Пара по волейболу ',Z,',', X ) end if, nl,  
fail;  
 \_ = readLine().  
end implement main  
goal  
    console::run(main::run).

