**Лабораторна робота № 5**

**Тема:** Списки

**Мета:** набути практичних навичок роботи зі списками.

**Короткі теоретичні відомості**

Список – це об’єкт, який містить скінченну кількість інших об’єктів. Список у Пролозі береться у квадратні дужки, й елементи списку розділяються комами. Список, який не містить жодного елемента, називається пустим списком. Список є рекурсивним об’єктом. Він складається з голови (першого елемента списку) і хвоста (всі наступні елементи). Хвіст також є списком. У пролозі є операція «|», яка дозволяє ділити список на голову і хвіст. Пустий список не можна розділити на голову і хвіст. Тип даних «список» оголошується у програмі на Пролозі таким чином: DOMAINS списковий\_тип = тип\* де «тип» – тип елементів списку; це може бути як стандартний тип, так і нестандартний, заданий користувачем і оголошений у розділі DOMAINS раніше. Основними операціями на списках є: − формування списку; − об’єднання списків; − пошук елемента у списку; − вставка елемента у список та видалення зі списку.

**Хід роботи:**

**Завдання до лабораторної роботи:**

**1.** Сформувати список [2, 4, 6, 8, 10] і видалити з нього введене число.

**2**. Сформувати списки [1, 3, 5, 7, 9] і [2, 4, б, 8, 10] і об'єднати їх в один.

**3.** Сформувати список [3, 6, 9, 12, 15, 18] і вставити у нього введене число.

**4.** Сформувати список з N натуральних чисел, починаючи з 10. Кожне наступне число на 5 більше за попереднє. **5.** Сформувати список [3, 6, 9, 12, 15] і знайти суму його елементів.

**6.** Сформувати список [6, 5, 4, 3, 2] і знайти суму його елементів.

**7.** Сформувати список [7, 5, 3, 1] і знайти добуток його елементів.

**8.** Сформувати список з N послідовних натуральних чисел, починаючи з 10. Знайти суму його елементів.

implement main  
    open core, console  
  
domains  
list = integer\*.  
class predicates  
     spisok:(integer,list,list) nondeterm anyflow.  
    gen:(integer,integer,list) nondeterm anyflow.  
    genp:( integer, integer, list ) nondeterm anyflow.  
    append:(list, list, list) nondeterm anyflow.  
lisp:(integer, integer, list) nondeterm anyflow.  
ins:(integer,list,list) nondeterm anyflow.  
add:(integer,list,list)nondeterm anyflow.  
genl:( integer, integer,integer, list,list ) nondeterm anyflow.  
lisk:(integer, integer, list) nondeterm anyflow.  
sum\_list:(list, integer) nondeterm anyflow.  
lispp:(integer, list) nondeterm anyflow.  
sum\_g:(list, integer) nondeterm anyflow.  
lik:(integer, list) nondeterm anyflow.  
sum\_k:(list, integer) nondeterm anyflow.  
ge:(integer, integer, list)  nondeterm anyflow.  
sum:(list, integer) nondeterm anyflow.  
  
clauses  
    spisok(X,[X|L],L).  
    spisok(X,[Y|L],[Y|L1]):-spisok(X,L,L1).  
        gen(B,B,[B]):-!.  
    gen(A,B,[A|Tail]):-A2 = A+2,gen(A2,B,Tail).  
  
      genp(B,B,[B]):-!.  
    genp(A,B,[A|M]):-A2 = A+2, A2=A+2,genp(A2,B,M).  
    append([],M,M).  
    append([X|M1],M2,[X|M3]):-append(M1,M2,M3).  
   lisp(B,B,[B]):-!.  
    lisp(A,B,[A|L]):-A2 = A+3, lisp(A2,B,L).  
   ins(X,[X|L],L).  
    ins(X,[X|L1],[X|L]):-ins(X,L1,L).  
    add(El,[],[El]):-!.  
add(El,[H|Tail],[H|T1]):-add(El,Tail,T1).  
genl(N2,N2,\_,L,L):-!.  
genl(N1,N2,K,L,B) :-  K1=K+5,N3=N2+1,add(K1,B,B1),genl(N1,N3,K1,L,B1).  
lisk(B,B,[B]):-!.  
    lisk(A,B,[A|K]):-A2 = A+3, lisk(A2,B,K).  
sum\_list([],0).  
sum\_list([X|K],S):-sum\_list(K,S1),S=S1+X.  
lispp(0,[]):-!.  
lispp(N,[N|С]):-N1=N-1, lispp(N1,С).  
sum\_g([],0).  
sum\_g([X|C],S):-sum\_g(C,S1),S=S1+X.  
  
lik(1,[]):-!.  
lik(N,[N|С]):-N1=N-2, lik(N1,С).  
sum\_k([],1).  
sum\_k([X|C],S):-sum\_k(C,S1),S=S1\*X.  
  
ge(B,B,[B]):-!.  
ge(A,B,[A|L]):-A2 = A+1, lisp(A2,B,L).  
sum([],0).  
sum([X|L],S):-sum(L,S1),S=S1+X.  
  
run() :-  
        init(),  
        write("Задание 1. Составить список 2, 4 ..10 и удалить элемент"), nl,  
        gen(2,10,L), write("L=",L),nl,write("X="),X=read(),spisok(X,L,L1), write("L=", L1), nl,  
  
    fail;  
    write("Задание 2. Составить списки [1, 3, 5, 7, 9] и [2, 4, 6, 8, 10] и объеденить их в один."), nl,  
        genp(2,10,M1), write(M1),nl,  
        genp(1,9,M2), write(M2),nl,  
        append(M1,M2,M3),write("L3=",M3),nl,  
 fail;  
 write("Задание 3. Добавить элемент в список."), nl,  
  lisp(3,18,L), write(L),nl,  
  write("X="),X=read(),  
ins(X,L1,L),write("L1=",L1),nl,  
fail;  
write("Задание 4. Составить список из N натуральных чисел, начиная с 10. Каждое следующее на 5 больше ."), nl,  
write("N="),N=read(), genl(N,1,10,L,[10]), nl,  
write(L),nl,  
fail;  
write("Задание 5. Найти сумму [3, 6, 9, 12, 15, 18] "), nl,  
  lisp(3,18,K), write(K),nl,  
sum\_list(K,S), write("S=",S), nl,  
fail;  
write("Задание 6. Найти сумму элементов [6, 5, 4, 3, 2, 1] "), nl,  
  lispp(6,C),write(C),nl,  
sum\_g(C,S), write("S=",S), nl,  
fail;  
write("Задание 7. Найти произведение элементов [7, 5, 3] "), nl,  
  lik(7,C),write(C),nl,  
sum\_k(C,S), write("S=",S), nl,  
fail;  
write("Задание 8. Составить список из N натуральных чисел, начиная с 10 и найти сумму: "), nl,  
ge(10,20,L),write(L),nl,  
sum(L,S), write("S=",S), nl,  
fail;  
\_ = readLine().  
end implement main  
  
goal  
    console::run(main::run).

