МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ**

**УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«СЕВАСТОПОЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

Институт информационных технологий

|  |  |
| --- | --- |
| Базовая кафедра | « Информационные технологии и компьютерные системы» |

Лабораторная работа №5

Вариант 2

“ ОБРАБОТКА СПИСКОВ”

Функциональное и логическое программирование

в ФГАОУ ВО «Севастопольский государственный университет

(наименование организации)

|  |  |
| --- | --- |
| Выполнил | Базарный А.Р. |

(Фамилия И.О. обучающегося)

|  |
| --- |
| ФункЛП/б-22-3-о |

(шифр группы)

|  |
| --- |
| Ткаченко Кирилл Станиславович |

(Фамилия И.О. преподавателя)

Севастополь 2024 г.

**Цель работы:** Ознакомиться с реализацией структуры данных типа список в языке Пролог и методами их обработки.

**Постановка задач:**

В ходе выполнения работы требуется:

1. Выполнить контрольные примеры.

Реализовать набор предикатов для обработки списков: ·

Добавление элемента X к списку List. Выдать новый список. Удаление элемента X из списка List. Выдать список без элемента Х.

Конкатенация списков.

Определение длины списка.

Определение принадлежности элемента Х списку List.

2.Создать программу на языке Пролог, которая решает задачи согласно варианту (исходные данные выбираются из приложения Г по формуле I=N(MOD 17)+1, J=(I+8)(MOD 17)+1.) 3. Составить отчет о проделанной работе.

**Ход работы**

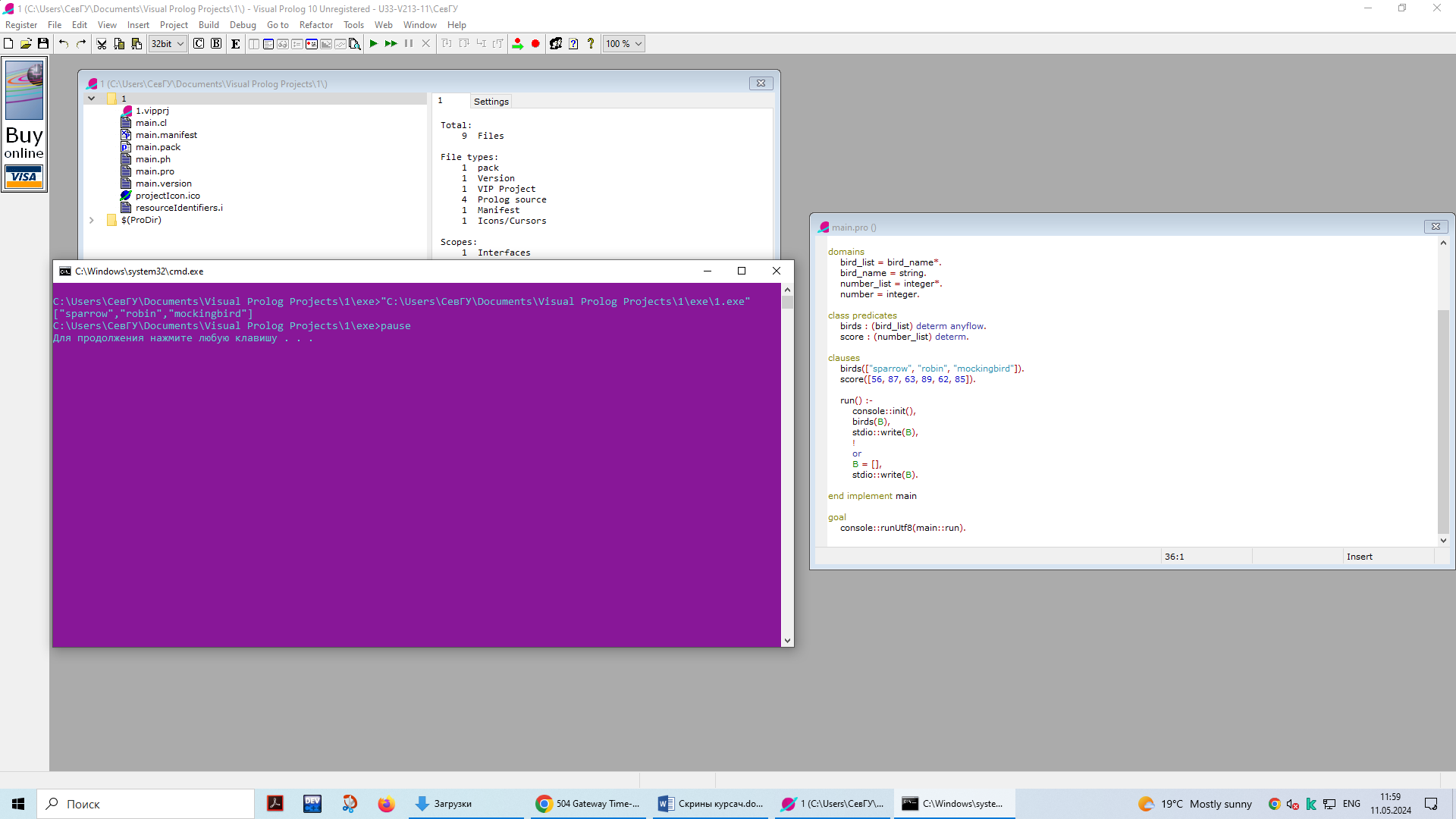


Рисунок 1: Первый пример из методических указаний

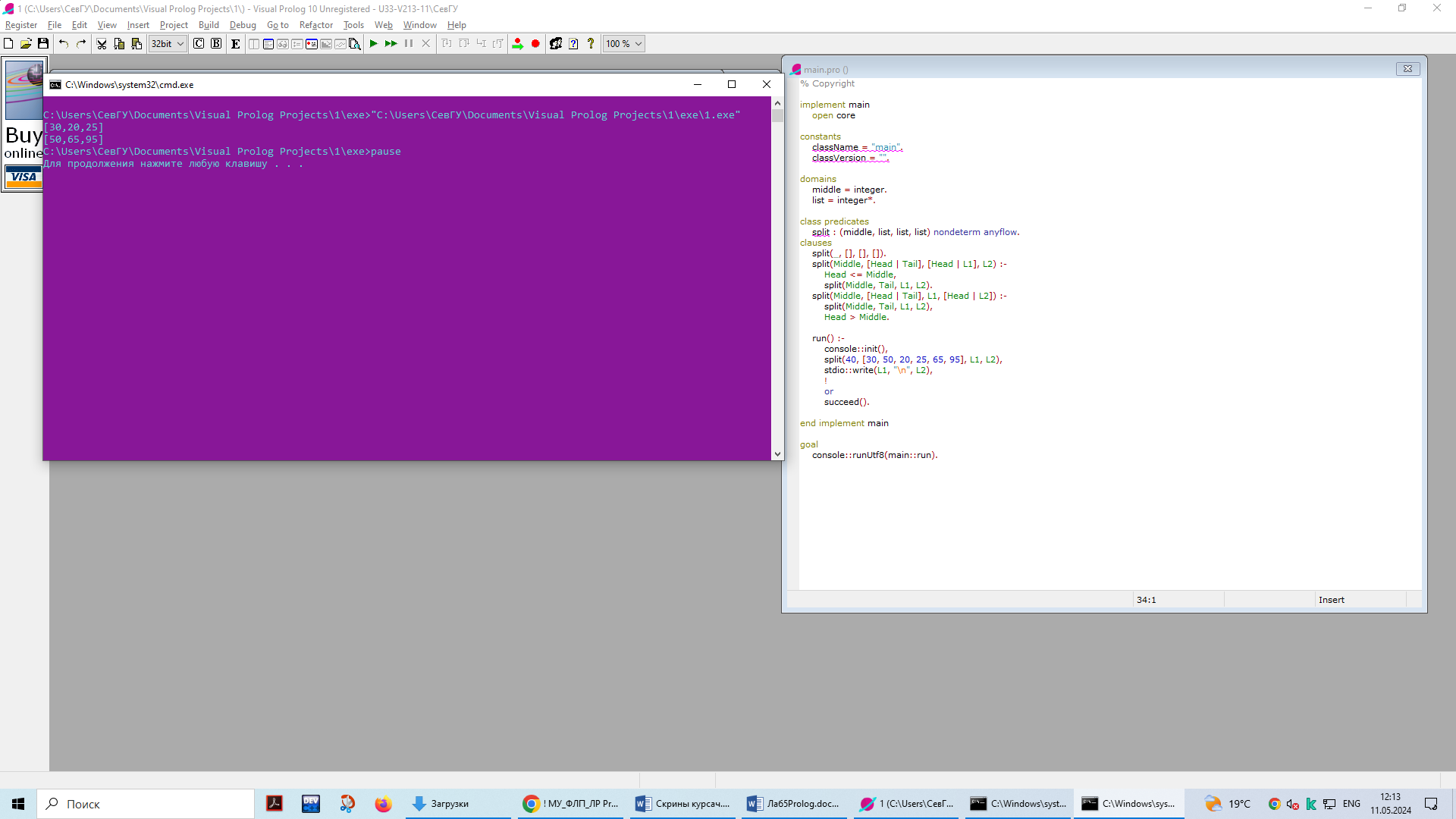


Рисунок 2: Второй пример из методических указаний

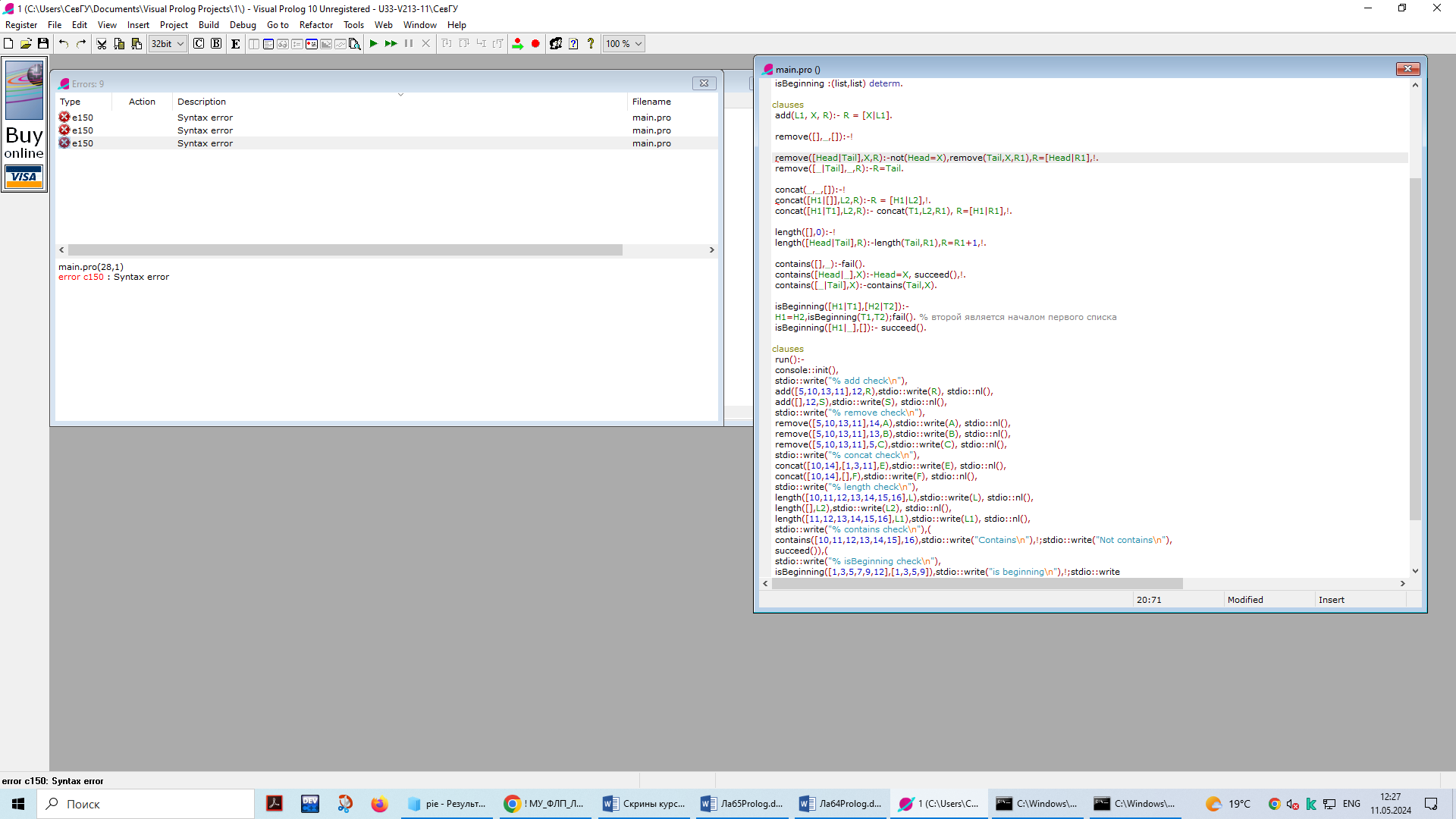


Рисунок 3: Программа из 3 примера не запускается из-за синтаксических ошибок в clauses

2. I = 3, J = 12

1) Определить принадлежность элемента X списку L.

2) Удалить повторяющиеся элементы в списке.

1) Код программы:  
implement main  
    open core  
  
domains  
    list = integer\*.  
  
class predicates  
    checkInList : (list, integer) determ.  
    removeFirst : (list, list [out]) determ.  
  
clauses  
    removeFirst([\_Head | Tail], R) :-  
        R = Tail.  
  
    checkInList([Head | Tail], ING) :-  
        if ING = Head then  
            stdio::write(ING)  
        else  
            removeFirst([Head | Tail], R),  
            checkInList(R, ING)  
        end if.  
  
clauses  
    run() :-  
        console::init(),  
        stdio::write("% checkInList check\n"),  
        stdio::write("15,16,17,18 || 16"),  
        (checkInList([15, 16, 17, 18], 19) and stdio::write("15,16,17,18 || 16") and ! or stdio::write(" InList") and succeed()),  
        stdio::nl(),  
        (checkInList([15, 16, 17, 18], 16) and stdio::write("15,16,17,18 || 19") and ! or stdio::write(" InList") and succeed()),  
        stdio::nl().  
  
end implement main  
  
goal  
    console::runUtf8(main::run).

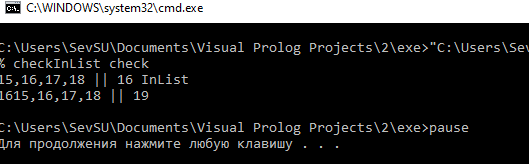


Рисунок 4: Пример работы программы

2)

Код программы

implement main  
    open core  
  
domains  
    list = integer\*.  
  
class predicates  
    checkInList : (list, integer) determ.  
    removeFirst : (list, list [out]) determ.  
    selectNonUnique : (list, integer) determ.  
    allToSelect : (list) determ.  
  
clauses  
    allToSelect([Head | Tail]) :-  
        selectNonUnique([Head | Tail], Head),  
        removeFirst([Head | Tail], R).  
  
    selectNonUnique([Head | Tail], INT) :-  
        if INT = Head then  
            stdio::write("Not Unique is "),  
            stdio::write(Head)  
        else  
            removeFirst([Head | Tail], R),  
            selectNonUnique(R, INT)  
        end if.  
  
    removeFirst([\_Head | Tail], R) :-  
        R = Tail.  
  
    checkInList([Head | Tail], ING) :-  
        if ING = Head then  
            stdio::write(ING)  
        else  
            removeFirst([Head | Tail], R),  
            checkInList(R, ING)  
        end if.  
  
clauses  
    run() :-  
        console::init(),  
        stdio::write("% SelectNonUnique check\n"),  
        (allToSelect([15, 16, 15, 15, 17, 18]) and stdio::write(" from 15, 16, 15, 15, 17, 18") and ! or stdio::write(" ...") and succeed()).  
  
end implement main  
  
goal  
    console::runUtf8(main::run).

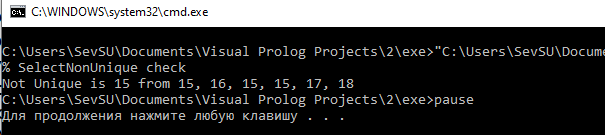


Рисунок 5: Пример работы программы

Деревья правил

allToSelect(list)

selectNonUnique([Head | Tail], Head) removeFirst([Head | Tail], R)

  selectNonUnique([Head | Tail], INT)

INT = Head

True: write head;

False: removeFirst([Head | Tail], R), selectNonUnique(R, INT)

  removeFirst([\_Head | Tail], R)

R = Tail

  checkInList([Head | Tail], ING)

ING = Head

True: write ING

False: removeFirst([Head | Tail], R), checkInList(R, ING)

**Вывод:** В ходе лабораторной работы были исследованы основные принципы работы со списками, реализованы 2 программы по варианту.