

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение высшего образования
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

ИНСТИТУТ НЕПРЕРЫВНОГО И ДИСТАНЦИОННОГО ОБРАЗОВАНИЯ

КАФЕДРА КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ И ПРОГРАММНОЙ ИНЖЕНЕРИИ

ОЦЕНКА

ПРЕПОДАВАТЕЛЬ

доцент, к.т.н.
должность, уч. степень, звание

подпись, дата

П.А.Степанов
инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ

Обработка списков в языке Prolog

по дисциплине: Функциональное и логическое программирование

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ гр. № _____
номер группы

подпись, дата

М.Д.Быстров
инициалы, фамилия

Студенческий билет № _____
2021/3572

Санкт-Петербург 2025

Оглавление

Оглавление	2
Цель работы	3
Задание.....	4
Исходный код	5
Результат выполнения программы	6
Вывод.....	7

Цель работы

Цель работы – ознакомление с принципами обработки списков на языке Prolog.

Задание

4. Разработать функцию, находящую теоретико-множественное пересечение двух списков.

Например:

Вход: (1 2 3 4 5), (4 5 6 7).

Выход: (4 5).

Исходный код

Файл "lab3.pl"

```
% вспомогательный предикат - есть ли в списке элемент
mymember([H|_], H):- !.
mymember(_|T], H):- mymember(T, H).

% вспомогательный предикат-присвоение
myassign(X, X).

% предикат, делающий список из уникальных элементов
myunique([], []):- !.

% решил написать так, подобие ветвления, чтобы не писать два предиката
% и ради проверки можно ли так написать
myunique([H1|T1], T3):-
    myunique(T1,T2),
    ((mymember(T2,H1),
    myassign(T2,T3));
    (not(mymember(T2,H1)),
    myassign([H1|T2],T3))).

% внутренние предикаты вычисления
% теоретико-множественного пересечения
% 1. корневой предикат для пустого первого списка
myintersect_service([],_,[]):- !.

% 2. предикат для случая, когда головы первого списка нет во втором списке
myintersect_service([H1|T1],[H2|T2],T3):-
    not(mymember([H2|T2], H1)),
    myintersect_service(T1, [H2|T2], T3).

% 3. предикат для случая, когда голова списка есть во втором списке
myintersect_service([H1|T1], [H2|T2], [H1|T3]):-
    mymember([H2|T2], H1),
    myintersect_service(T1, [H2|T2], T3).

% главный предикат "фронт-енд" - вычисление пересечения,
% удаление дубликатов
% и отключение дальнейших вычислений
myintersect(X,Y,Z):-
    myintersect_service(X,Y,R),
    myunique(R,Z),!.
```

Результат выполнения программы

```
[debug] ?- myintersect([2,2,3], [4,1,2,3,3,3], X).  
X = [2, 3].
```

```
[debug] ?- myintersect([2,2,3,3,3,3], [4,1,2,3,3,3], X).  
X = [2, 3].
```

```
[debug] ?- myintersect([10,2,2,3,3,3,3], [4,1,2,3,3,3,10], X).  
X = [10, 2, 3].
```

Рисунок 1 Результат выполнения программы

Вывод

В ходе выполнения первой лабораторной работы создан предикат на языке Prolog для выполнения поставленной задачи – поиск теоретико-множественного пересечения двух списков.

Приобретены навыки обработки списков на языке программирования Prolog.