**Лабораторная работа №1**

студента группы ИТ-222

Мокрищева Николая Павловича

Выполнение\_\_\_\_\_\_\_\_\_\_\_\_ Защита\_\_\_\_\_\_\_\_\_\_\_\_

**Продукционные системы. Семантические сети. Prolog**

**Цель работы**: ознакомится с язык и системами логического программирования Prolog.

**Содержание работы**

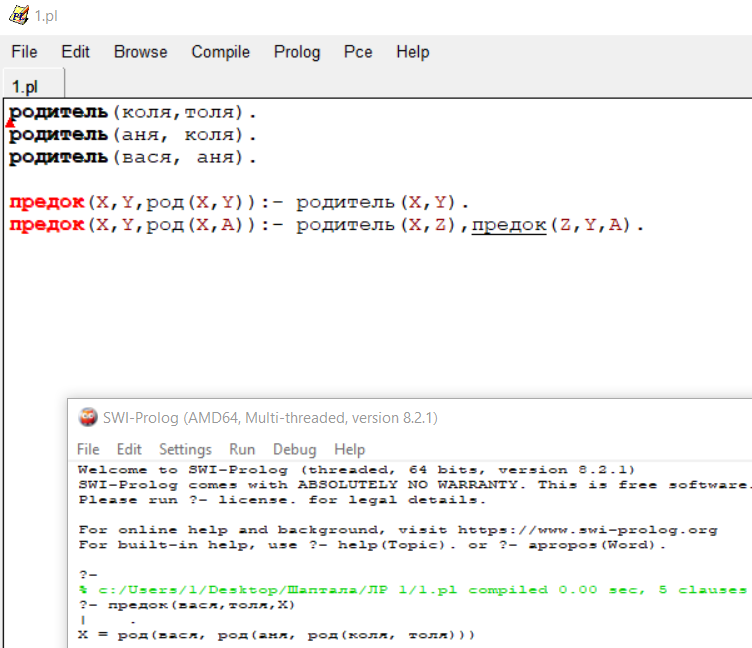
1. Выполните рассмотренные примеры.

2. Постройте и реализуйте средствами SWI Prolog семантическую сеть для какой-либо предметной области.

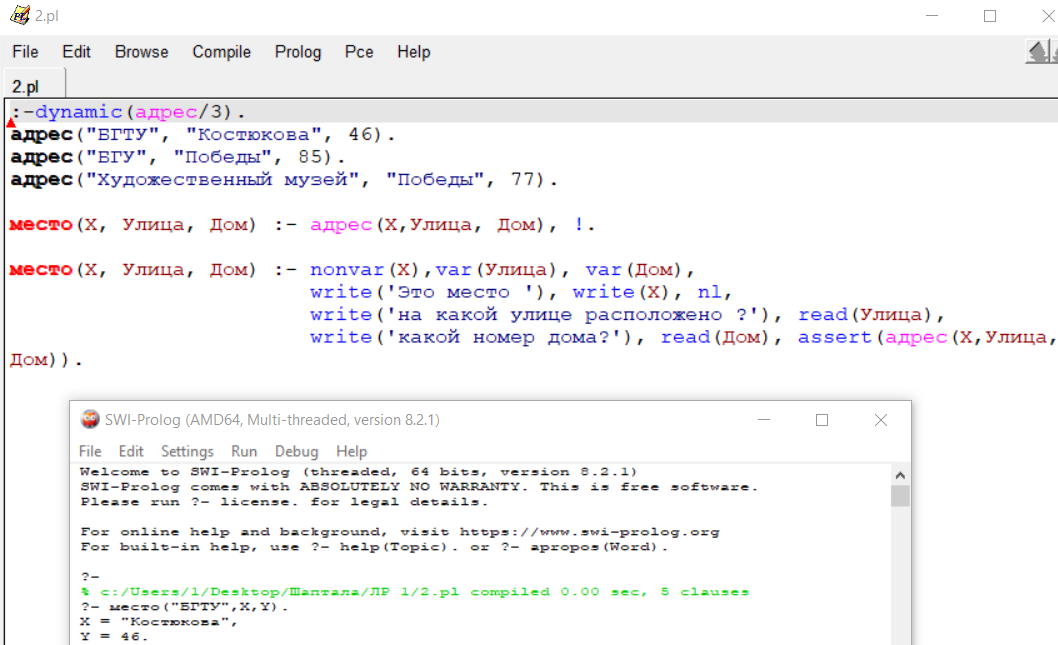
3. Напишите python программу, посылающую запросы к семантической сети.

**Ход работы**

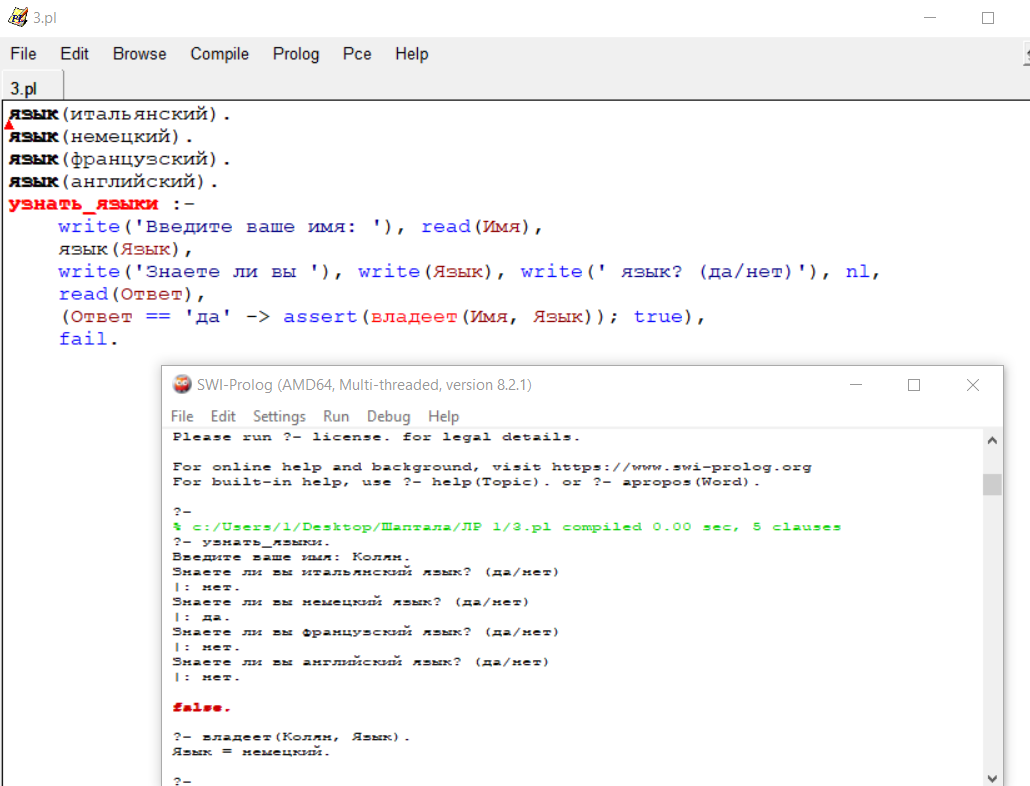
1. Формирование рекурсивных структур. Эта программа на языке Prolog предназначена для построения цепочки родственных связей между людьми с использованием фактов и правил о родителях и предках.



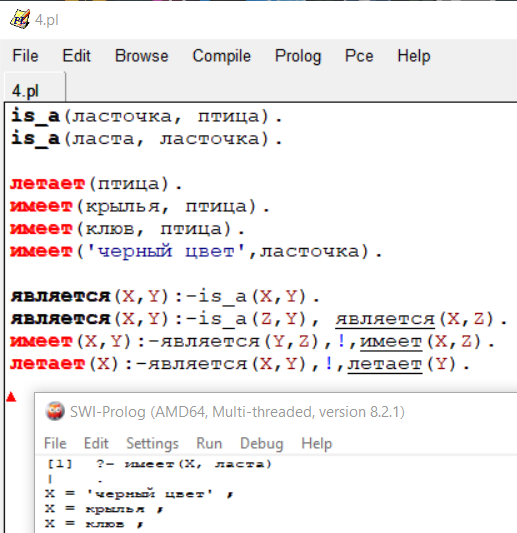
1. Формирование рекурсивных структур
2. Далее рассмотрим процедуру, которая учится у пользователя. Эта программа на Prolog динамически управляет базой данных адресов различных мест и предоставляет возможность пользователю вводить новые адреса, если информация о них отсутствует.



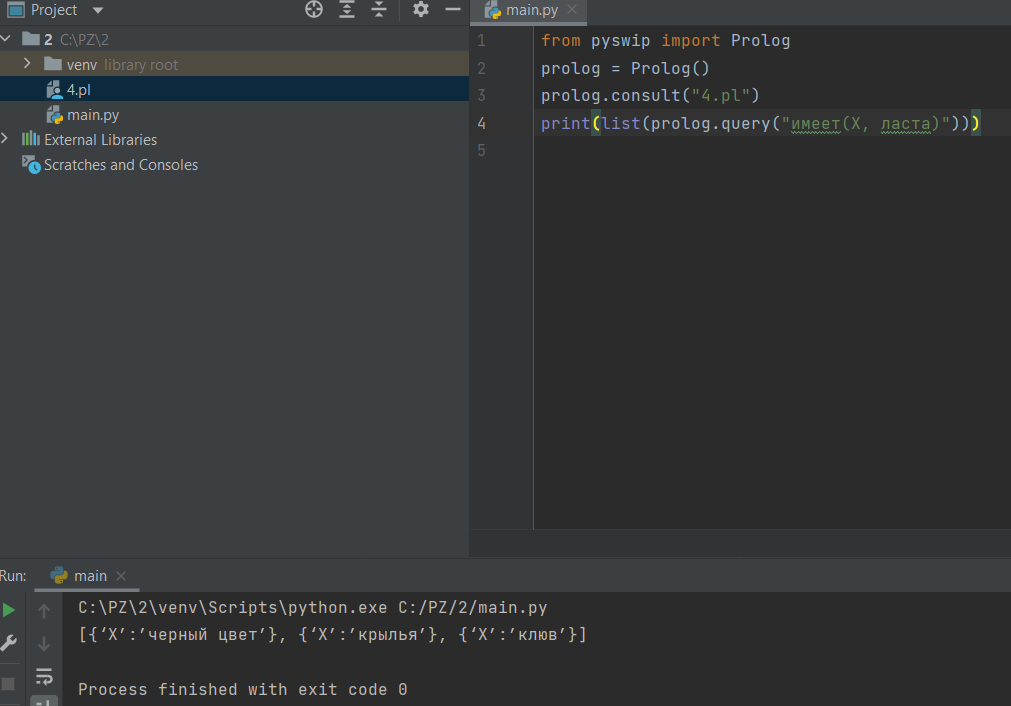
1. Процедура, которая учится у пользователя
2. Алгоритм поиска с возвратом, в котором используется предикат «assert». Эта программа на Prolog взаимодействует с пользователем, запрашивая его имя и проверяя, какими языками он владеет из списка предложенных. Программа последовательно предлагает языки и сохраняет информацию о владении ими.



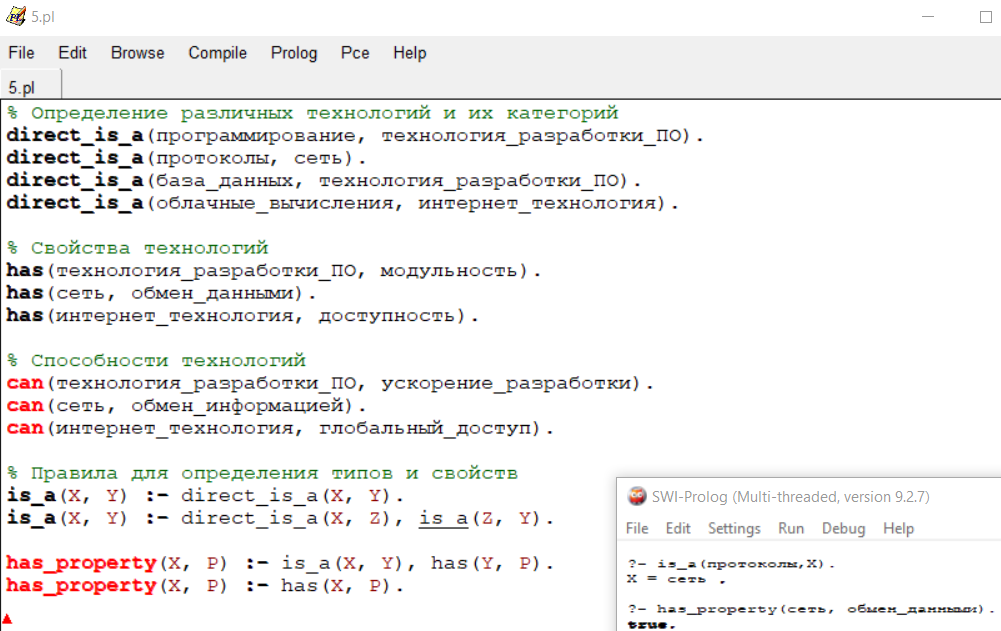
1. Алгоритм поиска с возвратом
2. Представление семантических сетей. Эта программа на языке Prolog определяет иерархию классов объектов, их свойства и методы для запроса этих свойств. Программа также показывает, как свойства передаются по иерархии классов.



1. Представление семантических сетей.
2. PySwip позволяет выполнять запросы SWI-пролог из программ на Python.



1. Запрос на Python
2. Напишем python программу, посылающую запросы к семантической сети.



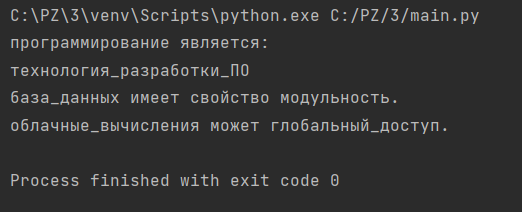
1. Семантическая сеть на Prolog

Код на python:

from pyswip import Prolog  
  
# Создание экземпляра Prolog  
prolog = Prolog()  
  
# Загрузка семантической сети из файла Prolog  
prolog.consult("technologies.pl") # Ваш Prolog файл  
  
# Функция для проверки типа объекта  
def check\_type(entity):  
 query = f"is\_a({entity}, X)"  
 results = list(prolog.query(query))  
 if results:  
 print(f"{entity} является:")  
 for result in results:  
 print(result["X"])  
 else:  
 print(f"Тип {entity} не найден.")  
  
# Функция для проверки свойства объекта  
def check\_property(entity, property):  
 query = f"has\_property({entity}, {property})"  
 results = list(prolog.query(query))  
 if results:  
 print(f"{entity} имеет свойство {property}.")  
 else:  
 print(f"{entity} не имеет свойства {property}.")  
  
# Функция для проверки способности объекта  
def check\_ability(entity, ability):  
 query = f"can({entity}, {ability})"  
 results = list(prolog.query(query))  
 if results:  
 print(f"{entity} может {ability}.")  
 else:  
 print(f"{entity} не может {ability}.")

# Пример запросов

# Пример запросов  
check\_type("программирование") # Проверка, к какому типу относится "программирование"  
check\_property("база\_данных", "модульность") # Проверка свойства "модульность" у "база\_данных"  
check\_ability("облачные\_вычисления", "глобальный\_доступ") # Проверка способности "глобальный доступ" у "облачные вычисления"



1. Результат выполненния

**Вывод**: в процессе работы была построена семантическая сеть на Prolog, охватывающая различные виды транспортных средств, их характеристики и возможности. Реализованные запросы продемонстрировали ключевые функции Prolog, включая обработку динамических данных и использование рекурсии. Кроме того, была разработана программа на Python с применением библиотеки pyswip для взаимодействия с семантической сетью Prolog, что продемонстрировало эффективность интеграции между двумя языками программирования. Работа подтвердила практическое применение логического программирования и его взаимодействие с другими технологиями.