Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования «Пермский национальный исследовательский политехнический университет» (ПНИПУ)

Кафедра вычислительной математики и механики

**Лабораторная работа № 3**

**по дисциплине: «Семантические сети»**

Выполнил

студент группы ИСТ-19-2б

Кошкаров П.С.

Проверил

ассистент кафедры ВММБ

Нетбай Г.В.

Пермь, 2022

**Машина логического вывода:** Программный компонент, который обеспечивает формирование логического вывода (принимая решение о том, каким правилам удовлетворяют факты или объекты), располагает выполняемые правила по приоритетам и выполняет правило с наивысшим приоритетом.

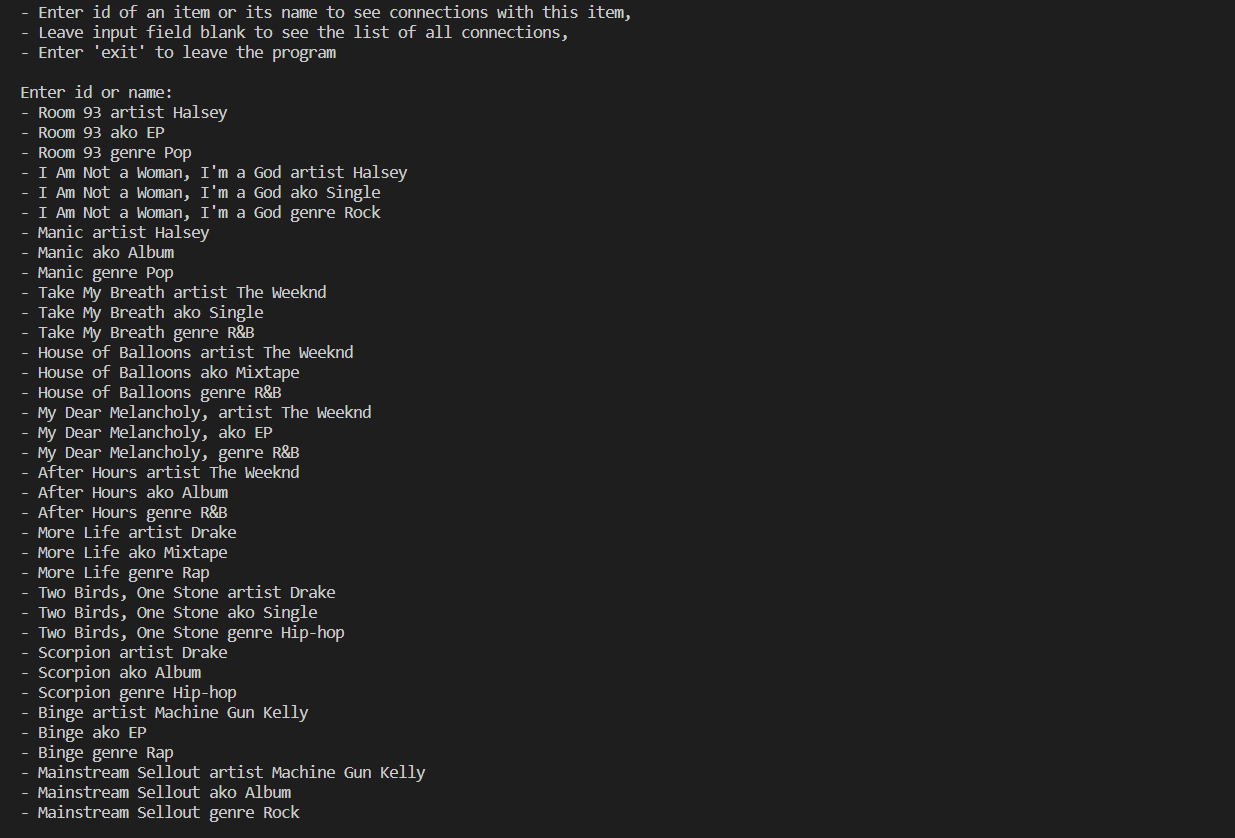
**Цель:** построить прототип ЭС, использующий различные иодели представления знаний.

**Описание:** в рамках данной работы необходимо спроектировать БЗ с представлением знаний в виде фреймов или семантических сетей, а также реализовать оболочку экспертной системы, работающую с данной БЗ.

**Хранение данных в JSON-файле:** для работы программы используется два json-файла, один хранит все данные в виде id и названия, а второй связи этих id по жанрам, артисту и типу выпущенного им материала.

**Пример работы программы:**

1. Вводим пустую строку, получаем список всех связей**:**



1. Вводим id, получаем все связи с этим номером:



3. Вводим название id, получаем все связи с этим названием:



# **Листинг кода Python**

import pathlib

import json

class Network:

    def \_\_init\_\_(self, file\_name\_ids, file\_name\_connections):

        self.Items = self.ReadItems(file\_name\_ids)

        self.Connections = self.ReadConnections(file\_name\_connections)

    def ItemWithName(self, name):

        for item\_id, item in self.Items.items():

                if item == name:

                    return item\_id

        return None

    def ReadItems(self, json\_file\_name):

        items = {}

        with open(json\_file\_name, "r", encoding='utf-8') as read\_entities:

            data = json.load(read\_entities)

            for key, value in data.items():

                items[key] = value

        return items

    def ReadConnections(self, json\_file\_name):

        connections = []

        with open(json\_file\_name, "r", encoding='utf-8') as read\_connections:

            data = json.load(read\_connections)

            for item in data:

                connections += [item]

        return connections

    def ListConnectionsWith(self, user\_input):

        if self.Items.get(user\_input):

            item\_id = user\_input

        else:

            item\_id = self.ItemWithName(user\_input)

        if item\_id == None:

            print("Items with this id or name doesn't exist")

            return

        for connection in self.Connections:

            if item\_id == connection[0] or item\_id == connection[2]:

                print("-", self.Items[connection[0]], connection[1], self.Items[connection[2]])

    def ListAll(self):

        for connection in self.Connections:

            print("-", self.Items[connection[0]], connection[1], self.Items[connection[2]])

folder = str(pathlib.Path(\_\_file\_\_).parent.resolve())

m = Network(folder+"/ids.json", folder+"/connections.json")

print('''\n

- Enter id of an item or its name to see connections with this item,

- Leave input field blank to see the list of all connections,

- Enter \'exit\' to leave the program''')

while True:

    user = input("\nEnter id or name: ")

    if user == "":

        m.ListAll()

    elif user == "exit":

        break

    else:

        print()

        m.ListConnectionsWith(user)

print("\nEnd of the program\n")