МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. Шухова»

(БГТУ им. В. Г. Шухова)

Лабораторная работа №7

По дисциплине: «Системы искусственного интеллекта»

На тему: «Модели представления знаний на основе фреймов»

Выполнил: студент группы ВТ41 Ковалёв И. Д.

Проверил: Жихарев А. Г.

г. Белгород

2020 г.

Цель работы: получение знаний, умений и навыков по представлению знаний в указанной предметной области в виде фреймовых моделей.

**Задание к работе**

**Задание 1**

Компания, в которой Вы работаете, получила задание на разработку справочной системы по журналам издательства «Издательство Мечты». Данная компания выпускает различные по целевой аудитории, ценовой категории и объему страниц журналы.

Ваша задача построить фреймовую модель на основе информации, представленной в таблице 1.

Таблица 1.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Название журнала | Основная целевая аудитория | Стоимость одного номера, руб | Объем страниц журнала | Какая информация представлена в журнале | Возможна ли подписка на журнал |
| Тюнинг автомобилей | Мужчины | 140 | 170 | Современные технологии тюнинга автомобилей | нет |
| Мода | Женщины | 90 | 90 | Новейшие тенденции моды | да |
| Компьютерные и видео игры | Мужчины и женщины | 65 | 60 | Все о компьютерных и видео играх | нет |
| Рукоделие | Женщины | 45 | 50 | Эксклюзивные вещи своими рука-ми | да |
| Фотография | Мужчины и женщины | 100 | 95 | Основы и секреты фото-графии | да |
| Кино и музыка | Мужчины и женщины | 30 | 30 | Только актуальная ин-формация и кино и музыке | нет |

В построенной системе фреймов определить:

1. Какой журнал предоставляет информацию о современных технологиях тюнинга автомобилей?

2. Какие журналы предназначены для мужчин?

3. Какие журналы стоят 100 рублей?

4. На какие журналы можно оформить подписку?

5. Найти все журналы издательства.

6. Найти все журналы, которые стоят не более 150 рублей.

**Задание 2**

Вы – офицер штаба армии. Вам поступило задание разработать экспертную систему, предназначенную для выбора десантного средства для тактических операций.

В наличии имеются следующие боевые транспортные аппараты:

Таблица 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование | Способ пере-движения | Вместимость, чел. | Вооружение | Защита |
| БМП-3 | Гусеницы | 7 | пушка 100 мм | броня |
| Ми-24 | Вертолет | 6 | пушка 30 мм | сталь |
| БТР-50Б | Колеса | 8 | пулемет 14.5 мм | броня |
| Ми-8 | Вертолет | 12 | нет | нет |

Необходимо создать систему, позволяющую выбирать боевые средства тактической доставки десанта на основе заданных требований к ним. Таким образом, задачей выполнения лабораторной работы является создание системы фреймов из знаний, представленных в таблице 2.

В созданной системе найти десантные средства, удовлетворяющие следующим требованиям:

1. Вертолет, способный перевозить не менее 6 человек десанта над полем боя без поддержки наземных средств;

2. Наземное средство доставки не менее 6 десантников в тыл противника при активном противодействии бронетанковых войск противника;

3. Средство доставки не менее 6 человек десанта в тыл противника при наличии противодействия противника.

**Задание 3**

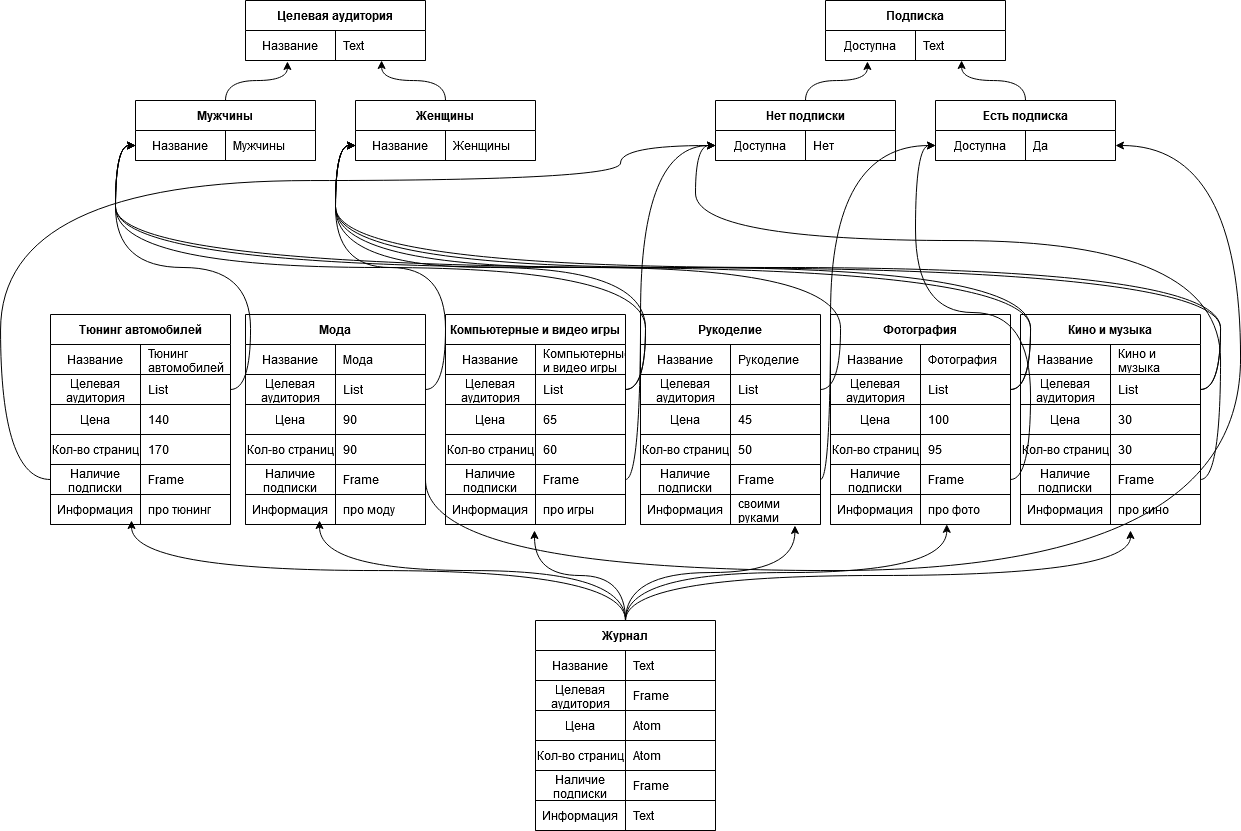
Представить знания в заданной предметной области в виде фреймовой модели. Написать программу, которая отвечала бы на вопрос пользователя, заданный на естественном языке по разработанной фреймовой модели представления знаний. Вопрос и данные выбираются по варианту задания.

4. Получение оценки при составляющих: преподаватель, студент, оценка, получать.

**Ход работы**

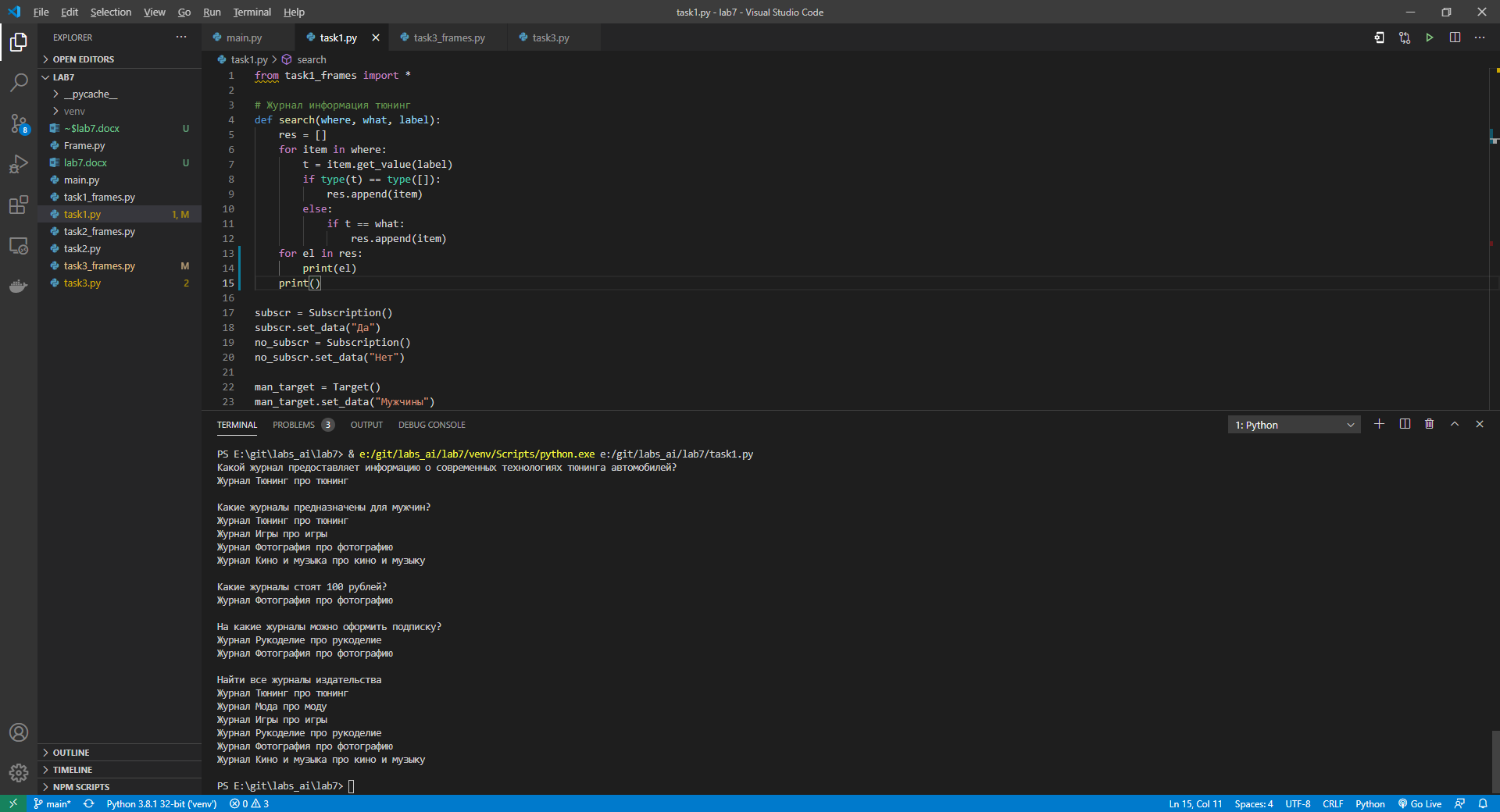
**Задание 1**

Построим граф, представляющий необходимые знания.



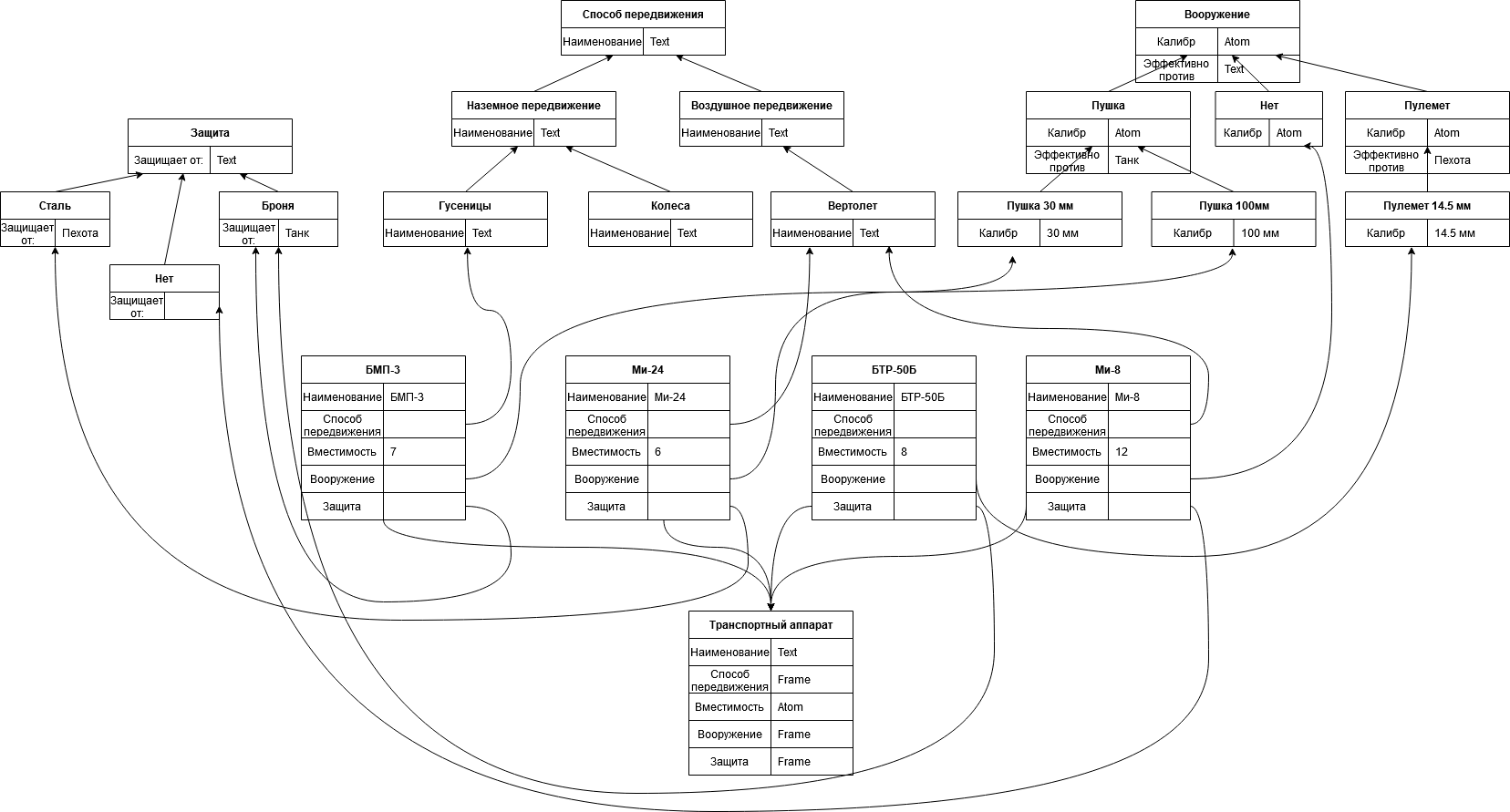
Реализация данного графа представлена в приложении 1.

Результат работы программы:



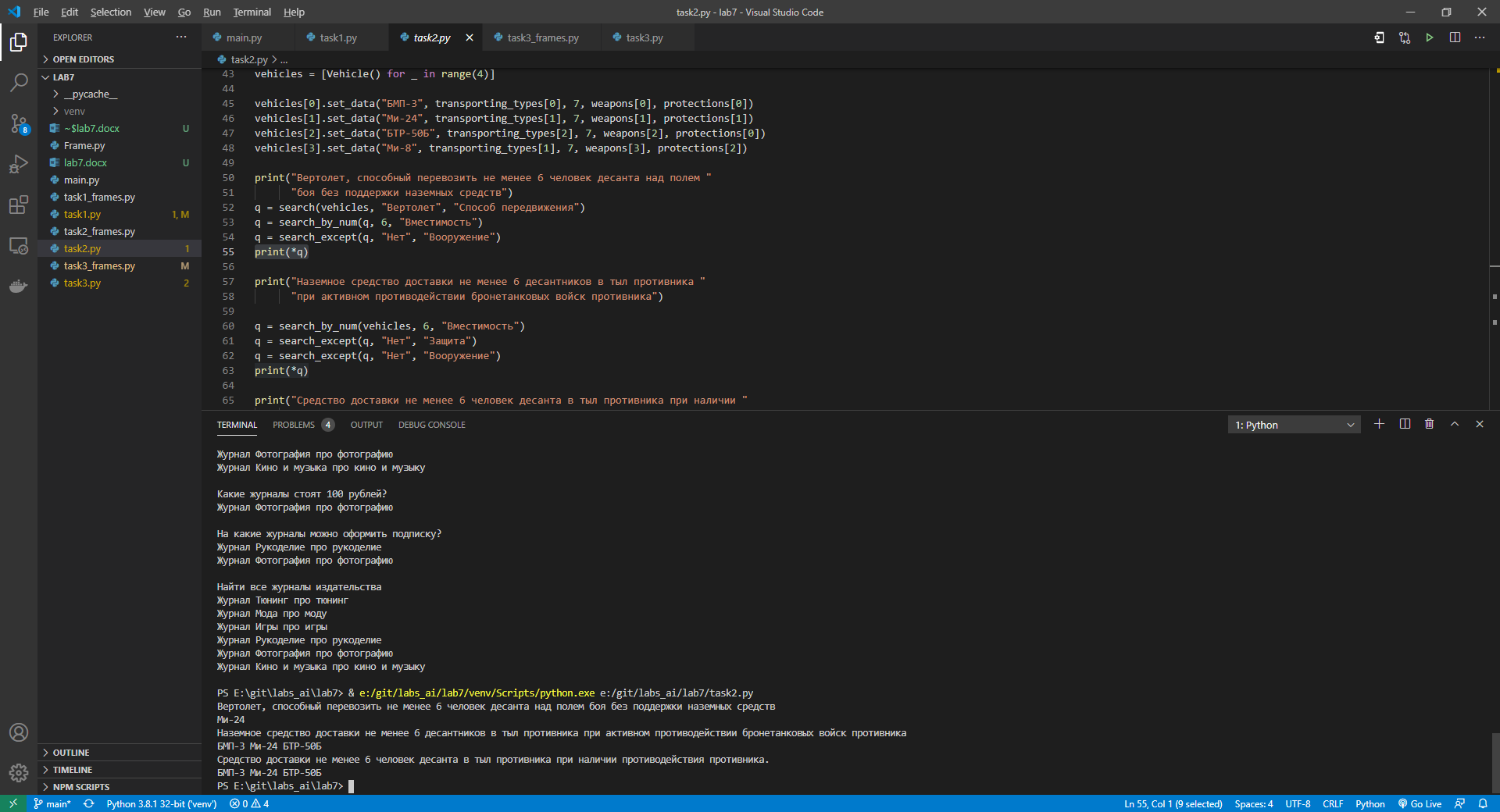
**Задание 2**

Построим граф, представляющий необходимые знания.



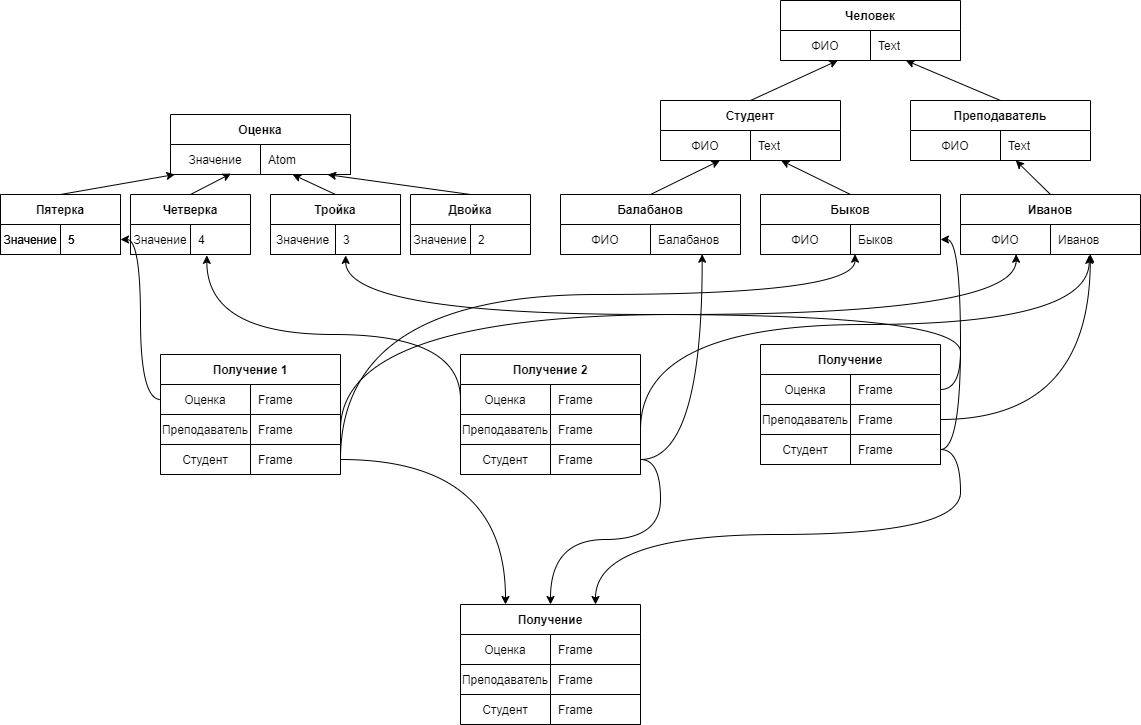
Реализация данного графа представлена в приложении 2.

Результат работы программы:



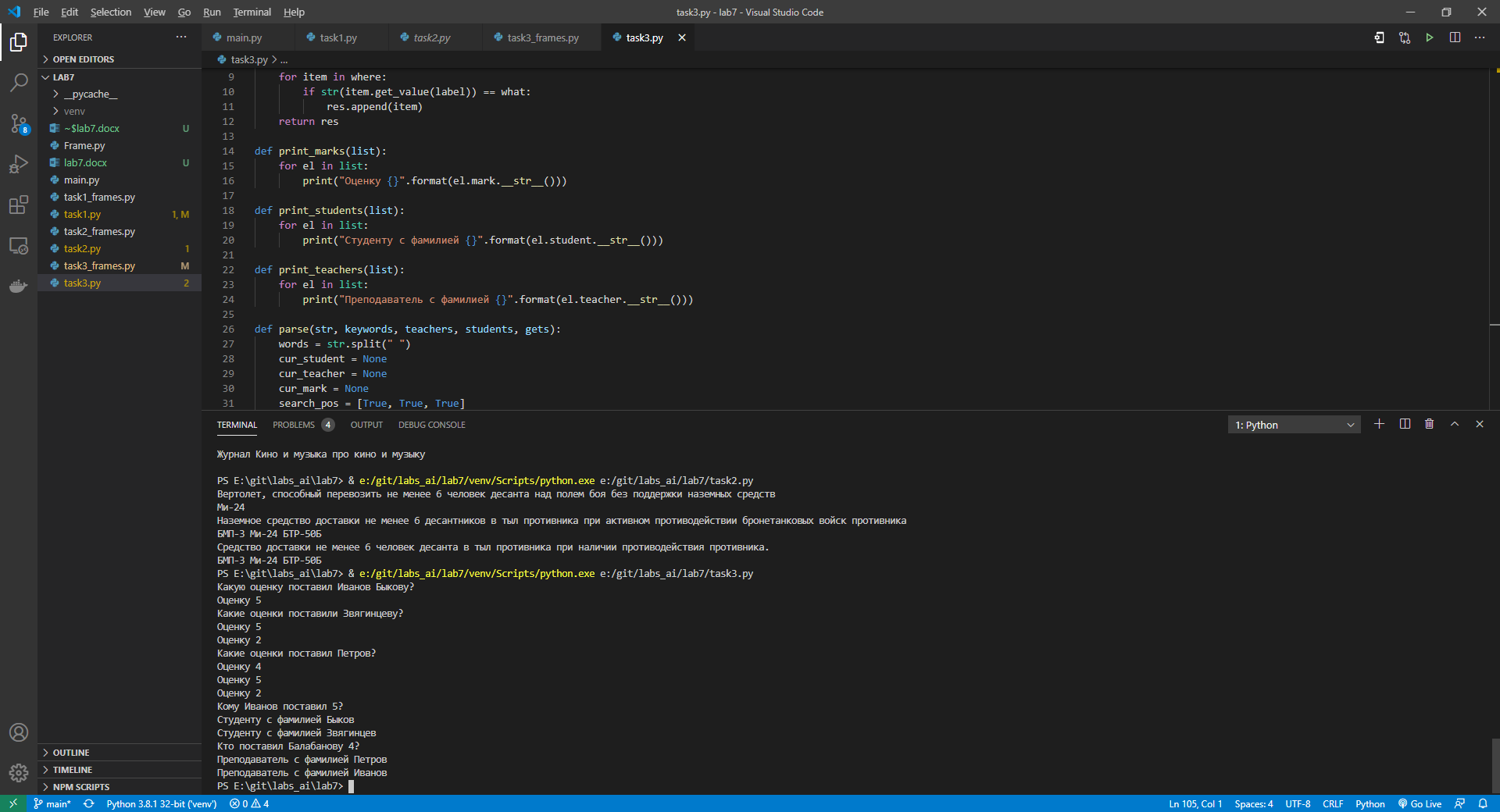
**Задание 3**

Построим граф, представляющий необходимые знания.



Реализация данного графа представлена в приложении 3.

Результат работы программы:



**Приложение 1. Содержимое task1.py:**

from task1\_frames import \*

def search(where, what, label):

    res = []

    for item in where:

        t = item.get\_value(label)

        if type(t) == type([]):

            res.append(item)

        else:

            if t == what:

                res.append(item)

    for el in res:

        print(el)

    print()

subscr = Subscription()

subscr.set\_data("Да")

no\_subscr = Subscription()

no\_subscr.set\_data("Нет")

man\_target = Target()

man\_target.set\_data("Мужчины")

antiman\_target = Target()

antiman\_target.set\_data("Женщины")

both = [man\_target, antiman\_target]

journals = [Journal() for \_ in range(6)]

journals[0].set\_data("Тюнинг", man\_target, 140, 170, "про тюнинг", no\_subscr)

journals[1].set\_data("Мода", antiman\_target, 90, 90, "про моду", no\_subscr)

journals[2].set\_data("Игры", both, 65, 60, "про игры", no\_subscr)

journals[3].set\_data("Рукоделие", antiman\_target, 45, 50, "про рукоделие", subscr)

journals[4].set\_data("Фотография", both, 100, 95, "про фотографию", subscr)

journals[5].set\_data("Кино и музыка", both, 30, 30, "про кино и музыку", no\_subscr)

print("Какой журнал предоставляет информацию о современных технологиях тюнинга автомобилей?")

search(journals, "про тюнинг", "Информация")

print("Какие журналы предназначены для мужчин?")

search(journals, man\_target, "Целевая аудитория")

print("Какие журналы стоят 100 рублей?")

search(journals, 100, "Стоимость")

print("На какие журналы можно оформить подписку?")

search(journals, subscr, "Подписка")

print("Найти все журналы издательства")

search(journals, "Журнал", "Журнал")

**Приложение 2. Содержимое task1.py:**

from task2\_frames import \*

def search(where, what, label):

    res = []

    for item in where:

        if str(item.get\_value(label)) == what:

            res.append(item)

    return res

def search\_by\_num(where, what, label):

    res = []

    for item in where:

        if item.get\_value(label) > what:

            res.append(item)

    return res

def search\_except(where, what, label):

    res = []

    for item in where:

        if not str(item.get\_value(label)) == what:

            res.append(item)

    return res

transporting\_types = [TransporingType() for \_ in range(3)]

transporting\_types[0].set\_data("Гусеницы")

transporting\_types[1].set\_data("Вертолет")

transporting\_types[2].set\_data("Колеса")

weapons = [Weapon() for \_ in range(4)]

weapons[0].set\_data("Пушка", "100 мм")

weapons[1].set\_data("Пушка", "30 мм")

weapons[2].set\_data("Пулемет", "14,5 мм")

weapons[3].set\_data("Нет", "")

protections = [Protection() for \_ in range(3)]

protections[0].set\_data("Броня")

protections[1].set\_data("Сталь")

protections[2].set\_data("Нет")

vehicles = [Vehicle() for \_ in range(4)]

vehicles[0].set\_data("БМП-3", transporting\_types[0], 7, weapons[0], protections[0])

vehicles[1].set\_data("Ми-24", transporting\_types[1], 7, weapons[1], protections[1])

vehicles[2].set\_data("БТР-50Б", transporting\_types[2], 7, weapons[2], protections[0])

vehicles[3].set\_data("Ми-8", transporting\_types[1], 7, weapons[3], protections[2])

print("Вертолет, способный перевозить не менее 6 человек десанта над полем "

      "боя без поддержки наземных средств")

q = search(vehicles, "Вертолет", "Способ передвижения")

q = search\_by\_num(q, 6, "Вместимость")

q = search\_except(q, "Нет", "Вооружение")

print(\*q)

print("Наземное средство доставки не менее 6 десантников в тыл противника "

      "при активном противодействии бронетанковых войск противника")

q = search\_by\_num(vehicles, 6, "Вместимость")

q = search\_except(q, "Нет", "Защита")

q = search\_except(q, "Нет", "Вооружение")

print(\*q)

print("Средство доставки не менее 6 человек десанта в тыл противника при наличии "

      "противодействия противника.")

q = search\_by\_num(vehicles, 6, "Вместимость")

q = search\_except(q, "Нет", "Защита")

print(\*q)

**Приложение 3. Содержимое task1.py:**

from task3\_frames import \*

studwords = ["Балабанов", "Быков", "Звягинцев"]

teachwords = ["Иванов", "Петров"]

def search(where, what, label):

    res = []

    for item in where:

        if str(item.get\_value(label)) == what:

            res.append(item)

    return res

def print\_marks(list):

    for el in list:

        print("Оценку {}".format(el.mark.\_\_str\_\_()))

def print\_students(list):

    for el in list:

        print("Студенту с фамилией {}".format(el.student.\_\_str\_\_()))

def print\_teachers(list):

    for el in list:

        print("Преподаватель с фамилией {}".format(el.teacher.\_\_str\_\_()))

def parse(str, keywords, teachers, students, gets):

    words = str.split(" ")

    cur\_student = None

    cur\_teacher = None

    cur\_mark = None

    search\_pos = [True, True, True]

    for word in words:

        if word.\_\_contains\_\_("?"):

            word = word[:len(word) - 1]

        for sw in studwords:

            if word.startswith(sw):

                cur\_student = sw

                search\_pos[0] = False

        for tw in teachwords:

            if word.startswith(tw):

                cur\_teacher = tw

                search\_pos[1] = False

        if word.isnumeric():

            cur\_mark = int(word)

            search\_pos[2] = False

    if not search\_pos[0] and not search\_pos[1]:

        s1 = search(gets, cur\_teacher, "Преподаватель")

        s2 = search(gets, cur\_student, "Студент")

        res = list(set(s1) & set(s2))

        print\_marks(res)

        return

    if not search\_pos[0] and not search\_pos[2]:

        s1 = search(gets, "{}".format(cur\_mark), "Оценка")

        s2 = search(gets, cur\_student, "Студент")

        res = list(set(s1) & set(s2))

        print\_teachers(res)

        return

    if not search\_pos[1] and not search\_pos[2]:

        s1 = search(gets, cur\_teacher, "Преподаватель")

        s2 = search(gets, "{}".format(cur\_mark), "Оценка")

        res = list(set(s1) & set(s2))

        print\_students(res)

        return

    if not search\_pos[0]:

        res = search(gets, cur\_student, "Студент")

        print\_marks(res)

    if not search\_pos[1]:

        res = search(gets, cur\_teacher, "Преподаватель")

        print\_marks(res)

teachers = [Teacher() for \_ in range(2)]

teachers[0].set\_data("Иванов")

teachers[1].set\_data("Петров")

students = [Student() for \_ in range(3)]

students[0].set\_data("Балабанов")

students[1].set\_data("Быков")

students[2].set\_data("Звягинцев")

marks = [Mark() for \_ in range(4)]

marks[0].set\_data(2)

marks[1].set\_data(3)

marks[2].set\_data(4)

marks[3].set\_data(5)

gets = [Get() for \_ in range(6)]

gets[0].set\_data(teachers[0], students[0], marks[2])

gets[1].set\_data(teachers[0], students[1], marks[3])

gets[2].set\_data(teachers[0], students[2], marks[3])

gets[3].set\_data(teachers[1], students[0], marks[2])

gets[4].set\_data(teachers[1], students[1], marks[3])

gets[5].set\_data(teachers[1], students[2], marks[0])

print("Какую оценку поставил Иванов Быкову?")

parse("Какую оценку поставил Иванов Быкову?", [], teachers, students, gets)

print("Какие оценки поставили Звягинцеву?")

parse("Какие оценки поставили Звягинцеву?", [], teachers, students, gets)

print("Какие оценки поставил Петров?")

parse("Какие оценки поставил Петров?", [], teachers, students, gets)

print("Кому Иванов поставил 5?")

parse("Кому Иванов поставил 5?", [], teachers, students, gets)

print("Кто поставил Балабанову 4?")

parse("Кто поставил Балабанову 4?", [], teachers, students, gets)