

Московский государственный технический университет имени Н.Э.Баумана

---

Факультет РТ Радиотехнический

Кафедра ИУ5 Системы обработки информации и управления

**Отчет по рубежному контролю №1 по курсу  
Базовые компоненты интернет-технологий.**

**“Изучение объектно-ориентированных возможностей языка Python.”**

4

(количество листов)

Вариант № 26

Исполнитель

студент группы РТ5-31Б \_\_\_\_\_ Таланкина В.Д.

“28” октября 2022 г.

Проверил

Доцент кафедры ИУ5 \_\_\_\_\_ Гапанюк Ю.Е.

“28” октября 2022 г.

## Вариант Е. Предметная область 26.

- 1) «Студенческая группа» и «Учебный курс» связаны соотношением один ко многим. Выведете список всех курсов, у которых в названии присутствует слово «Course», и список учащихся на них групп.
- 2) «Студенческая группа» и «Учебный курс» связаны соотношением один ко многим. Выведете список курсов со средним рейтингом групп на каждом курсе, отсортированном в по среднему рейтингу. Средний рейтинг должен быть округлен до 2 знаков после запятой.
- 3) «Студенческая группа» и «Учебный курс» связаны соотношением многие ко многим. Выведете список всех групп, у которых название начинается с буквы «R», и названия курсов.

## 2. Листинг программы.

```
from operator import itemgetter
from tokenize import group

class Group:
    def __init__(self, id, name, rating):    # Constructor
        self.id = id
        self.name = name
        self.rating = rating # 0 - 100

class Course:
    def __init__(self, name, id):    # Constructor
        self.name = name
        self.id = id

# more to more
class GroupCourse:
    def __init__(self, group_id, course_id):    # Constructor
        self.group_id = group_id
        self.course_id = course_id

group = [
    Group(1, "RT5-31B", 56.67),
    Group(2, "IU5-31B", 31.42),
    Group(3, "RK6-41B", 78.89),
    Group(4, "E9-21B", 99.99),
    Group(5, "MT2-11B", 3.14),
]

course = [
```

```

    Course("Math Course", 1),
    Course("Physics Course", 2),
    Course("Data Models", 3),
    Course("Python", 4),
    Course("Internet Technologies Course", 5),
]

group_course = [
    GroupCourse(1, 1),
    GroupCourse(1, 3),
    GroupCourse(2, 2),
    GroupCourse(2, 3),
    GroupCourse(3, 1),
    GroupCourse(3, 2),
    GroupCourse(4, 1),
    GroupCourse(4, 3),
    GroupCourse(5, 2),
    GroupCourse(5, 3),
]

# «Course» и «Group» связаны соотношением один-ко-многим.
# Выведите список всех Course, у которых в названии присутствует слово «Course»,
# и список Group.

def Task1 ():
    result = []
    for i in course:
        mid_result = []
        if "Course" in i.name:
            mid_result.append(i.name)
            for j in group_course:
                if j.group_id == i.id:
                    for k in group:
                        if k.id == j.course_id:
                            mid_result.append(k.name)
            result.append(mid_result)
    return result

# «Course» и «Group» связаны соотношением один-ко-многим.
# Выведите список Course со средним рейтингом (rating) Group
# в каждом Course, отсортированный по среднему значению.

def Task2 ():
    result = []
    for i in course:

```

```

        mid_result = []
        mid_result.append(i.name)
        sum = 0
        count = 0
        for j in group_course:
            if j.group_id == i.id:
                for k in group:
                    if k.id == j.course_id:
                        sum += k.rating
                        count += 1
        mid_result.append(sum / count)
        result.append(mid_result)
    result = sorted(result, key=itemgetter(1), reverse=False)
    return result

# «Course» и «Group» связаны соотношением многие-ко-многим.
# Выведите список всех Group,
# у которых название начинается с буквы «R»,
# и список Course.

def Task3 ():
    result = []
    for i in group :
        mid_result = []
        if "R" == i.name[0]:
            mid_result.append(i.name)
            for j in group_course:
                if j.course_id == i.id:
                    for k in course:
                        if k.id == j.group_id:
                            mid_result.append(k.name)
            result.append(mid_result)
    return result

def main() :
    print("\nTask 1:", Task1())

    print("\nTask 2:", Task2())

    print("\nTask 3:", Task3())

if __name__ == "__main__":
    main()

```

### 3. Результаты работы программы.

```
Task 1: [['Math Course', 'RT5-31B', 'RK6-41B'], ['Physics Course', 'IU5-31B', 'RK6-41B'], ['Internet Technologies Course', 'IU5-31B', 'RK6-41B']]
```

```
Task 2: [['Data Models', 44.045], ['Physics Course', 55.155], ['Internet Technologies Course', 55.155], ['Math Course', 67.78], ['Python', 67.78]]
```

```
Task 3: [['RT5-31B', 'Math Course', 'Data Models', 'Python'], ['RK6-41B', 'Math Course', 'Physics Course', 'Python', 'Internet Technologies Course']]
```