

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.”**

Вариант № 30

Исполнитель

студент группы РТ5-31Б

Чинаев А.В.

“28” октября 2022 г.

Проверил

Доцент кафедры ИУ5

Гапанюк Ю.Е.

“28” октября 2022 г.

Вариант Е. Предметная область 30

1. University и Faculty связаны отношением один-ко-многим. Выведите список всех университетов, у которых в названии присутствует слово "University", и факультетов в них.
2. University и Faculty связаны отношением один-ко-многим. Выведите список всех университетов со средним количеством студентов в каждом факультете, отсортированный по среднему количеству. Среднее количество должно быть округлено до 2 знаков после запятой рейтингу. Средний рейтинг должен быть округлен до 2 знаков после запятой.
3. University и Faculty связаны отношением многие-ко-многим. Выведите список всех факультетов, у которых название начинается с буквы "G", и список вузов, в которых имеется такой факультет.

Листинг программы

```
from operator import itemgetter
```

```
class Faculty:
    def __init__(self, name, id, university_id,
students_count): # Constructor
        self.name = name
        self.id = id
        self.university_id = university_id
        self.students_count = students_count
```

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

```
# Многие-ко-многим
```

```
class FacultyUniversity:
```

```
    def __init__(self, faculty_id, university_id): #
```

```
Constructor
```

```
        self.faculty_id = faculty_id
```

```
        self.university_id = university_id
```

```
# Students
```

```
university = [
```

```
    University(1, "BMSTU"),
```

```
    University(2, "SPSU"),
```

```
    University(3, "Moscow State University"),
```

```
    University(4, "Novosibirsk State University"),
```

```
    University(5, "TSU"),
```

```
]
```

```
# Groups
```

```
faculty = [
```

```
    Faculty("FN4", 1, 1, 1000),
```

```
    Faculty("Georgafical", 2, 2, 2343),
```

```
    Faculty("mathematical", 3, 4, 3524),
```

```
    Faculty("chemical", 4, 3, 2345),
```

```
    Faculty("CS", 5, 3, 2676),
```

```
    Faculty("social", 6, 4, 2345),
```

```
    Faculty("philosophic", 7, 5, 1289),
```

```
Faculty("Germanic", 8, 3, 790),  
]
```

```
# Students and Groups  
faculty_university = [  
    FacultyUniversity(1, 1),  
    FacultyUniversity(1, 2),  
    FacultyUniversity(2, 3),  
    FacultyUniversity(3, 4),  
    FacultyUniversity(4, 3),  
    FacultyUniversity(4, 3),  
    FacultyUniversity(5, 3),  
    FacultyUniversity(5, 1),  
    FacultyUniversity(6, 4),  
    FacultyUniversity(7, 5),  
    FacultyUniversity(8, 4),  
    FacultyUniversity(8, 3),  
    FacultyUniversity(8, 5),  
]
```

```
# University и Faculty связаны отношением один-ко-многим.  
# Выведите список всех университетов, у которых в названии  
присутствует  
# слово "University", и факультетов в них.
```

```
def task1():
```

```

result = []
for i in university:
    middle_result = []
    if "University" in i.name:
        middle_result.append(i.name)
        for j in faculty:
            if j.university_id == i.id:
                middle_result.append(j.name)
        result.append(middle_result)
return result

```

University и Faculty связаны отношением один-ко-многим.

Выведите список всех университетов со средним количеством студентов в каждом факультете,

отсортированный по среднему количеству. Среднее количество должно быть округлено до 2 знаков после запятой

```

def task2():
    result = []
    for i in university:
        sum = 0
        count = 0
        for j in faculty:
            if j.university_id == i.id:
                sum += j.students_count
                count += 1

```

```

        if (count == 0):
            result.append((i.name, 0))
        else:
            result.append((i.name, round(sum / count, 2)))
    return sorted(result, key=itemgetter(1), reverse=True)

```

```

# University и Faculty связаны отношением многие-ко-многим.
# Выведите список всех факультетов, у которых название начинается
с буквы "G",
# и список вузов, в которых имеется такой факультет.

```

```

def task3():
    result = []
    for i in faculty:
        if i.name[0] == "G":
            middle_result = [i.name]
            for j in faculty_university:
                if i.id == j.faculty_id:
                    for k in university:
                        if j.university_id == k.id:
                            middle_result.append(k.name)
            result.append(middle_result)
    return result

```

```

def main():

```

```
print("\nTask 1:")  
print(task1())
```

```
print("\nTask 2:")  
print(task2())
```

```
print("\nTask 3:")  
print(task3())
```

```
if __name__ == '__main__':  
    main()
```

Результаты выполнения программы

```
Task 1:  
[['Moscow State University', 'chemical', 'CS', 'Germanic'], ['Novosibirsk State University', 'mathematical', 'social']]  
  
Task 2:  
[('Novosibirsk State University', 2934.5), ('SPSU', 2343.0), ('Moscow State University', 1937.0), ('TSU', 1289.0), ('BMSTU', 1000.0)]  
  
Task 3:  
[['Georgafical', 'Moscow State University'], ['Germanic', 'Novosibirsk State University', 'Moscow State University', 'TSU']]  
  
Process finished with exit code 0
```