

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.”

5

(количество листов)

Вариант № 1

Исполнитель

студент группы РТ5-21 _____

Аюшиев Т.О.

“27” октября 2022 г.

Проверил

Доцент кафедры ИУ5 _____

Гапанюк Ю.Е.

“27” октября 2022 г.

Вариант Е. Предметная область 1.

1. “Студент” и “Группа” связаны отношением один-ко-многим. Выведите список всех групп, у которых в названии присутствует слово “Group”, и список студентов в них.
2. “Студент” и “Группа” связаны отношением один-ко-многим. Выведите список всех групп со средней зарплатой студентов в каждой группе, отсортированный по среднему рейтингу. Средний рейтинг должен быть округлен до 2 знаков после запятой.
3. “Студент” и “Группа” связаны отношением многие-ко-многим. Выведите список всех студентов, у которых фамилия начинается с буквы “А”, и список группы, в которой они учатся.

Листинг программы.

```
from operator import itemgetter
```

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

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

```
# Многие-ко-многим
class StudentGroup:
    def __init__(self, group_id, student_id): #
Constructor
        self.group_id = group_id
        self.student_id = student_id
```

```
# Students
```

```
student = [  
    Student(1, "Almaev", 1, 54.96),  
    Student(2, "Petrov", 2, 73.34),  
    Student(3, "Ayushiev", 3, 99.12),  
    Student(4, "Vasilev", 4, 12.34),  
    Student(5, "Akhmetzyanov", 5, 3.14),  
]
```

```
#Groups  
group = [  
    Group("Group 1", 1),  
    Group("RT-31B", 2),  
    Group("Group 3", 3),  
]
```

```
# Students and Groups  
student_group = [  
    StudentGroup(1, 1),  
    StudentGroup(1, 2),  
    StudentGroup(2, 3),  
    StudentGroup(2, 4),  
    StudentGroup(3, 5),  
]
```

```
# Student и Group связаны соотношением один-ко-многим.  
# Выведите список всех групп, у которых в названии  
присутствует  
# слово "Group", и список студентов в них.
```

```
def task1():  
    result = []  
    for i in group:  
        mid_result = []  
        if "Group" in i.name:  
            mid_result.append(i.name)  
            for j in student_group:  
                if j.group_id == i.id:  
                    for k in student:
```

```
        if k.id == j.student_id:
            mid_result.append(k.name)
    result.append(mid_result)
return result
```

Student и Group связаны соотношением один-ко-многим.
Выведите список всех групп со средней зарплатой студентов в каждой группе, отсортированный по среднему рейтингу. Средний рейтинг должен быть округлен до 2 знаков после запятой

```
def task2():
    result = []
    for i in group:
        sum = 0
        count = 0
        for j in student_group:
            if j.group_id == i.id:
                for k in student:
                    if k.id == j.student_id:
                        sum += k.rating
                        count += 1
        result.append((i.name, round(sum / count, 2)))
    return sorted(result, key=itemgetter(1), reverse=True)
```

Студент и группа связаны отношением многие-ко-многим.
Выведите список всех студентов, у которых фамилия начинается с буквы "А", и список группы, в которых они учатся.

```
def task3():
    result = []
    for i in student:
        if i.name[0] == "А":
            for j in student_group:
                if i.id == j.student_id:
```

```
        for k in group:
            if j.group_id == k.id:
                result.append((i.name,
k.name))
    return result
```

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

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

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

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

Результаты работы программы.

```
Task 1:
[['Group 1', 'Almaev', 'Petrov'], ['Group 3', 'Akhmetzyanov']]

Task 2:
[('Group 1', 64.15), ('RT-31B', 55.73), ('Group 3', 3.14)]

Task 3:
[('Almaev', 'Group 1'), ('Ayushiev', 'RT-31B'), ('Akhmetzyanov', 'Group 3')]
```