

МГТУ им. Н.Э. Баумана

Отчёт по рубежному контролю №1
по курсу «Базовые компоненты и интернет-технологии»
Вариант 1.

Руководитель
Гапанюк Ю.Е.
28.10.2022

Студент группы ИУ5-34Б
Баширов Г.К.
13.12.2022

2022 г.

Полученное задание:

Необходимо создать два класса данных в соответствии с Вашим вариантом предметной области, которые связаны отношениями один-ко-многим и многие-ко-многим.

Необходимо создать списки объектов классов, содержащих тестовые данные (3-5 записей), таким образом, чтобы первичные и вторичные ключи соответствующих записей были связаны по идентификаторам.

Необходимо разработать запросы в соответствии с Вашим вариантом.

Предметная область: класс_1 – Студент, класс_2 – Группа,
вариант запросов: Г.

Запросы:

1. «Группа» и «Студент» связаны соотношением один-ко-многим.
Выведите список всех групп, у которых название начинается с «ИУ», и список состоящих в них студентов.
2. «Группа» и «Студент» связаны соотношением один-ко-многим.
Выведите список групп с максимальным количеством студентов в каждой группе, отсортированный по максимальному количеству студентов.
3. «Группа» и «Студент» связаны соотношением многие-ко-многим.
Выведите список всех связанных студентов и групп, отсортированный по группам, сортировка по студентам произвольная.

Текст программы:

```
from operator import itemgetter

class Student:
    """Студент"""

    def __init__(self, id, name, average_mark, group_id):
        self.id = id
        self.name = name
        self.average_mark = average_mark
        self.group_id = group_id

class Group:
    """Группа"""

    def __init__(self, id, name):
        self.id = id
        self.name = name

class StudGroup:
    """
    'Студенты групп' для реализации
    связи многие-ко-многим
    """
```

```

def __init__(self, group_id, student_id):
    self.group_id = group_id
    self.student_id = student_id

# Группы
groups = [
    Group(1, "ФН7-31Б"),
    Group(2, "Э2-32Б"),
    Group(3, "ИУ5-34Б"),

    Group(11, "ФН7-31М"),
    Group(22, "Э2-32М"),
    Group(33, "ИУ5-34М"),
]

# Студенты
students = [
    Student(1, "Андреев", 4.7, 1),
    Student(2, "Семенов", 4.5, 2),
    Student(3, "Иванов", 4.2, 3),
    Student(4, "Сергеев", 4.6, 3),
    Student(5, "Федоров", 4.3, 3),
]

stud_group = [
    StudGroup(1, 1),
    StudGroup(2, 2),
    StudGroup(3, 3),
    StudGroup(3, 4),
    StudGroup(3, 5),

    StudGroup(11, 1),
    StudGroup(22, 2),
    StudGroup(33, 3),
    StudGroup(33, 4),
    StudGroup(33, 5),
]

def main():
    """Основная функция"""

    # Соединение данных один-ко-многим
    one_to_many = [(s.name, s.average_mark, g.name)
                    for g in groups
                    for s in students
                    if s.group_id == g.id]

    # Соединение данных многие-ко-многим
    many_to_many_temp = [(g.name, sg.group_id, sg.student_id)
                           for g in groups
                           for sg in stud_group
                           if g.id == sg.group_id]

    many_to_many = [(s.name, s.average_mark, group_name)
                     for group_name, group_id, student_id in many_to_many_temp
                     for s in students if s.id == student_id]

    print('Задание Г1')
    res_1 = {}
    for g in groups:
        if g.name[:2] == "ИУ":
            g_students = list(filter(lambda i: i[2] == g.name, one_to_many))
            g_students_names = [x for x, _, _ in g_students]
            if g_students_names:
                res_1[g.name] = g_students_names

    print(res_1)

    print('\nЗадание Г2')

```

```

res_2_unsorted = []
for g in groups:
    g_students = list(filter(lambda i: i[2] == g.name, one_to_many))
    if len(g_students) > 0:
        g_av = [av for _, av, _ in g_students]
        # Суммарная зарплата сотрудников отдела
        g_av_max = max(g_av)
        res_2_unsorted.append((g.name, g_av_max))
res_2 = sorted(res_2_unsorted, key=itemgetter(1), reverse=True)
print(res_2)

print('\nЗадание Г3')
res_3 = sorted(many_to_many, key=itemgetter(2))
for i in res_3:
    print(i)

if __name__ == '__main__':

```

Результаты выполнения:

Задание Г1

```
{'ИУ5-34Б': ['Иванов', 'Сергеев', 'Федоров']}
```

Задание Г2

```
[('ФН7-31Б', 4.7), ('ИУ5-34Б', 4.6), ('Э2-32Б', 4.5)]
```

Задание Г3

```

('Иванов', 4.2, 'ИУ5-34Б')
('Сергеев', 4.6, 'ИУ5-34Б')
('Федоров', 4.3, 'ИУ5-34Б')
('Иванов', 4.2, 'ИУ5-34М')
('Сергеев', 4.6, 'ИУ5-34М')
('Федоров', 4.3, 'ИУ5-34М')
('Андреев', 4.7, 'ФН7-31Б')
('Андреев', 4.7, 'ФН7-31М')
('Семенов', 4.5, 'Э2-32Б')
('Семенов', 4.5, 'Э2-32М')

```