Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»

Рубежный контроль №1
по дисциплине «Базовые компоненты интернет-технологий»

Выполнил:
студент группы
ИУ5Ц-52Б Дзауров И.А.
_____, “___” _____ 2022 г.

Проверил:
преподаватель кафедры
ИУ5 - Гапанюк Ю.Е.
_____, “___” _____ 2022 г.

Москва, 2022 г.

Описание задания

Вариант предметной области – 26 [«Студенческая группа» - «Учебный курс»].
Вариант запросов – Б.

1. «Учебный курс» и «Студенческая группа» связаны соотношением один-ко-многим. Выведите список всех связанных студенческих групп и учебных курсов, отсортированный по студенческим группам, сортировка по учебным курсам произвольная.
2. «Учебный курс» и «Студенческая группа» связаны соотношением один-ко-многим. Выведите список учебных курсов с количеством студенческих групп в каждом учебном курсе, отсортированный по количеству студенческих групп.
3. «Учебный курс» и «Студенческая группа» связаны соотношением многие-ко-многим. Выведите список всех студенческих групп, у которых название заканчивается на «31Б», и названия их учебных курсов.

Классы данных для предметной области [«Студенческая группа» - «Учебный курс»]:

1. Класс «Студенческая группа», содержащий поля:
   - ID записи о студенческой группе;
   - Название студенческой группы;
   - Количество студентов в студенческой группе (количественный признак);
   - ID записи об учебном курсе. (для реализации связи один-ко-многим)
2. Класс «Учебный курс», содержащий поля:
   - ID записи об учебном курсе;
   - Название учебного курса.
3. (Для реализации связи многие-ко-многим) Класс «Учебный курс в студенческой группе», содержащий поля:
   - ID записи о студенческой группе;
   - ID записи об учебном курсе.

Листинг программы

```
# This is a sample Python script.

# используется для сортировки
from operator import itemgetter


class Group:
    """Студенческая группа"""

    def __init__(self, id, name, studentsCount, course_id):
        self.id = id
        self.name = name
        self.studentsCount = studentsCount  # количество студентов
        self.course_id = course_id


class Course:
    """Учебный курс"""

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


class GroupCourse:
    """
    'Учебный курс в студенческой группе' для реализации
    связи многие-ко-многим
    """

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


# Учебные курсы
courses = [
    Course(1, 'БКИТ'),
    Course(2, 'Английский язык'),
    Course(3, 'Теория вероятности'),
    Course(4, 'Экология')
]

# Студенческие группы
groups = [
    Group(1, 'ИУ5-32Б', 18, 1),
    Group(2, 'ИУ5-31Б', 20, 1),
    Group(3, 'ИУ5-33Б', 22, 1),
    Group(4, 'ИУ5-34Б', 21, 1),
    Group(5, 'Э9-31Б', 16, 4),
    Group(6, 'Э9-22Б', 18, 4),
    Group(7, 'Э9-33Б', 20, 4),
]
```

```python
groups_courses = [
    GroupCourse(1, 1),
    GroupCourse(1, 2),
    GroupCourse(1, 3),

    GroupCourse(1, 4),
    GroupCourse(2, 1),
    GroupCourse(2, 2),

    GroupCourse(2, 3),
    GroupCourse(2, 4),
    GroupCourse(3, 1),

    GroupCourse(3, 2),
    GroupCourse(3, 3),
    GroupCourse(3, 4),

    GroupCourse(4, 1),
    GroupCourse(4, 2),
    GroupCourse(4, 3),

    GroupCourse(4, 4),
    GroupCourse(5, 2),
    GroupCourse(5, 3),

    GroupCourse(5, 4),
    GroupCourse(6, 2),
    GroupCourse(6, 3),

    GroupCourse(6, 4),
    GroupCourse(7, 2),
    GroupCourse(7, 3),
    GroupCourse(7, 4),
]


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

    # Соединение данных один-ко-многим
    one_to_many = [(g.name, g.studentsCount, c.name)
                   for c in courses
                   for g in groups
                   if g.course_id == c.id]

    # Соединение данных многие-ко-многим
    many_to_many_temp = [(c.name, gc.course_id, gc.group_id)
                         for c in courses
                         for gc in groups_courses
                         if c.id == gc.course_id]

    many_to_many = [(g.name, g.studentsCount, course_name)
        for course_name, course_id, group_id in many_to_many_temp
        for g in groups if g.id==group_id]


    print('Задание Б1')
```

```
    res_11 = sorted(one_to_many, key=itemgetter(0))
    print(res_11)

    print('\nЗадание Б2')
    res_12_unsorted = []
    # Перебираем все учебные курсы
    for c in courses:
        # Список учебных курсов
        c_groups = list(filter(lambda i: i[2] == c.name, one_to_many))
        # Если учебный курс не пустой
        if len(c_groups) > 0:
            # Количество студенческих групп
            c_lens = [len for _, len, _ in c_groups]
            res_12_unsorted.append((c.name, len(c_lens)))

    # Сортировка по количеству студенческих групп
    res_12 = sorted(res_12_unsorted, key=itemgetter(1), reverse=False)
    print(res_12)

    print('\nЗадание Б3')
    res_13 = {}
    # Перебираем все учебные курсы
    for g in groups:
        if g.name.endswith('31Б'):
            # Список учебных курсов у студенческих групп
            g_courses = list(filter(lambda i: i[0]== g.name, many_to_many))
            # Только название студенческой группы
            g_courses_name = [x[2] for x in g_courses]
            # ключ - студенческая группа, значение - список учебных курсов
            res_13[g.name] = g_courses_name

    print(res_13)


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

Экранные формы с примерами выполнения программы

Результаты выполнения:

```
Задание Б1
[('ИУ5-31Б', 20, 'БКИТ'), ('ИУ5-32Б', 18, 'БКИТ'), ('ИУ5-33Б', 22,
'БКИТ'), ('ИУ5-34Б', 21, 'БКИТ'), ('Э9-22Б', 18, 'Экология'), ('Э9-
31Б', 16, 'Экология'), ('Э9-33Б', 20, 'Экология')]

Задание Б2
[('Экология', 3), ('БКИТ', 4)]

Задание Б3
{'ИУ5-31Б': ['БКИТ', 'Английский язык', 'Теория вероятности',
'Экология'], 'Э9-31Б': ['Английский язык', 'Теория вероятности',
'Экология']}
```