

Отчёт по рубежному контролю №1 по курсу «Базовые компоненты
интернет-технологий»

Вариант 1.

Руководитель
Гапанюк Ю.Е.

Выполнил
студент группы ИУ5-31Б
Абуховский Иван
23.10.2021

МГТУ им. Н.Э. Баумана
2021 г.

Задание

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

1. Класс «Студент», содержащий поля:
 - ID записи о студенте;
 - Фамилия студента;
 - Количество долгов;
 - ID записи о группе. (для реализации связи один-ко-многим)
2. Класс «Группа», содержащий поля:
 - ID записи о группе;
 - Наименование группы.
3. (Для реализации связи многие-ко-многим) Класс «Студенты группы», содержащий поля:
 - ID записи о студенте;
 - ID записи о группе.

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

3) Необходимо разработать запросы в соответствии с вариантом. Запросы сформулированы в терминах классов «Студент» и «Группа», которые используются в примере. Нужно перенести эти требования в выданный вариант предметной области.

Предметная область: Вариант 1 - Студент-Группа

Запрос: Вариант А.

1. «Группа» и «Студент» связаны соотношением один-ко-многим. Выведите список всех связанных студентов и групп, отсортированный по группам, сортировка по студентам произвольная.
2. «Группа» и «Студент» связаны соотношением один-ко-многим. Выведите список групп с суммарным количеством долгов студентов в каждой группе, отсортированный по суммарной количеству долгов.
3. «Группа» и «Студент» связаны соотношением многие-ко-многим. Выведите список всех групп, у которых в названии присутствует индекс «Б», и список учащихся в них студентов.

Текст программы

```
from operator import itemgetter

class Student:
    """Студент"""
    def __init__(self, id, surname, dolgi,
group_id):
        self.id = id
        self.surname = surname
        self.dolgi = dolgi
        self.group_id = group_id

class Group:
    """Группа"""
    def __init__(self, id, name):
        self.id = id
        self.name = name

class StudentGroup:
    """
    'Студенты группы' 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, 'ИУ5-31Б'),
    Group(2, 'ИУ6-12Б'),
    Group(3, 'ИУ7-53Б'),
    Group(4, 'ИУ8-34Б'),
    Group(5, 'ИУ9-15Б'),

    Group(11, 'МТ5-11'),
    Group(22, 'РК5-32'),
    Group(33, 'СМ5-13'),
    Group(44, 'Э5-54'),
    Group(55, 'ОЭП5-35'),
]

# Студенты
students = [
    Student(1, 'Абуховский', 25, 1),
    Student(2, 'Черноморец', 47, 2),
    Student(3, 'Шагиахметов', 39, 3),
    Student(4, 'Стельмах', 81, 4),
    Student(5, 'Сыса', 17, 5),
    Student(6, 'Слепченкова', 43, 11),
```

```
Student(7, 'Кузьмин', 31, 22),
Student(8, 'Соколов', 29, 33),
Student(9, 'Заточен', 48, 44),
Student(10, 'Проценко', 33, 55),
Student(11, 'Калинников', 21, 1),
Student(12, 'Милевич', 21, 2),
Student(13, 'Слоква', 65, 3),
Student(14, 'Барабанщиков', 15, 4),
Student(15, 'Акулова', 83, 5),
Student(16, 'Ашуров', 12, 11),
Student(17, 'Бекетов', 14, 22),
Student(18, 'Поляков', 11, 33),
Student(19, 'Нигматуллин', 10, 44),
Student(20, 'Собакевич', 10, 55),
Student(21, 'Ахтамбаев', 32, 3),
Student(22, 'Цуприков', 15, 33),
]
```

```
students_groups = [
    StudentGroup(1,1),
    StudentGroup(1,11),
    StudentGroup(2,2),
    StudentGroup(2,12),
    StudentGroup(3,3),
    StudentGroup(3,13),
```

```
StudentGroup(4, 4),
StudentGroup(4, 13),
StudentGroup(5, 5),
StudentGroup(5, 15),
StudentGroup(11, 6),
StudentGroup(11, 16),
StudentGroup(22, 7),
StudentGroup(22, 17),
StudentGroup(33, 8),
StudentGroup(33, 18),
StudentGroup(44, 9),
StudentGroup(44, 19),
StudentGroup(55, 10),
StudentGroup(55, 20),
StudentGroup(3, 21),
StudentGroup(33, 22),
]
```

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

    # Соединение данных один-ко-многим
    one_to_many = [(s.surname, s.dolgi, 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 students_groups
        if g.id==sg.group_id]

    many_to_many = [(s.surname, s.dolgi,
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 A1')

    res_11 = sorted(one_to_many,
key=itemgetter(2))
    print(res_11)

    print('\nЗадание A2')
    res_12_unsorted = []
    # Перебираем все группы
    for g in groups:
        # Список студентов группы
        g_students = list(filter(lambda i:
i[2]==g.name, one_to_many))

```

```

        # Если группа не пустая
        if len(g_students) > 0:
            # Долги студентов группы
            g_dolgi = [dolgi for _, dolgi, _ in
g_students]

            # Суммарное количество долгов
студентов группы

            g_dolgi_sum = sum(g_dolgi)
            res_12_unsorted.append((g.name,
g_dolgi_sum))

        # Сортировка по суммарной зарплате
        res_12 = sorted(res_12_unsorted,
key=itemgetter(1), reverse=True)
        print(res_12)

print('\nЗадание А3')
res_13 = {}
# Перебираем все группы
for g in groups:
    if 'Б' in g.name:
        # Список студентов группы
        g_students = list(filter(lambda i:
i[2]==g.name, many_to_many))

        # Только ФИО сотрудников
        g_students_surnames = [x for x, _, _
in g_students]

```



```

# Добавляем результат в словарь
# ключ - отдел, значение - список
# фамилий

res_13[g.name] =
g_students_surnames

print(res_13)

if __name__ == '__main__':
    main()

```

Результат выполнения программы

```

C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python37_64\python.exe
Задание A1
[('Абуховский', 25, 'ИУ5-31Б'), ('Калинников', 21, 'ИУ5-31Б'), ('Черноморец', 47, 'ИУ6-12Б'), ('Милевич', 21, 'ИУ6-12Б'),
 ('Шагиахметов', 39, 'ИУ7-53Б'), ('Слоква', 65, 'ИУ7-53Б'), ('Ахтамбаев', 32, 'ИУ7-53Б'), ('Стельмах', 81, 'ИУ8-34Б'),
 ('Барабанщиков', 15, 'ИУ8-34Б'), ('Сыса', 17, 'ИУ9-15Б'), ('Акулова', 83, 'ИУ9-15Б'), ('Слепченкова', 43, 'МТ5-11'), ('А
шуров', 12, 'МТ5-11'), ('Проценко', 33, 'ОЭП5-35'), ('Собакевич', 10, 'ОЭП5-35'), ('Кузьмин', 31, 'РК5-32'), ('Бекетов',
 14, 'РК5-32'), ('Соколов', 29, 'СМ5-13'), ('Поляков', 11, 'СМ5-13'), ('Цуприков', 15, 'СМ5-13'), ('Заточен', 48, 'Э5-54
'), ('Нигматуллин', 10, 'Э5-54')]

Задание A2
[('ИУ7-53Б', 136), ('ИУ9-15Б', 100), ('ИУ8-34Б', 96), ('ИУ6-12Б', 68), ('Э5-54', 58), ('МТ5-11', 55), ('СМ5-13', 55), ('
ИУ5-31Б', 46), ('РК5-32', 45), ('ОЭП5-35', 43)]

Задание A3
{'ИУ5-31Б': ['Абуховский', 'Калинников'], 'ИУ6-12Б': ['Черноморец', 'Милевич'], 'ИУ7-53Б': ['Шагиахметов', 'Слоква', 'Ах
тамбаев'], 'ИУ8-34Б': ['Стельмах', 'Слоква'], 'ИУ9-15Б': ['Сыса', 'Акулова']}
Press any key to continue . . .

```