

Московский государственный технический университет им. Н.Э. Баумана

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

**Курс «Базовые компоненты интернет-технологий»
Отчет по рубежному контролю №1**

Выполнил:

студент группы ИУ5-31Б
Бондаренко Денис
Константинович

Проверил:

преподаватель каф. ИУ5
Гапанюк Юрий
Евгеньевич

Подпись: _____

Дата: _____

Подпись: _____

Дата: _____

Москва, 2021 г.

Рубежный контроль №1

Текст программы

```
from operator import itemgetter

class Pup:
    """ШКОЛЬНИК"""

    def __init__(self, id, fio, dbt, cls_id):
        self.id = id
        self.fio = fio
        self.dbt = dbt
        self.cls_id = cls_id
```

```
class Cls:
    """Класс"""

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

```
class PupCls:
    """
    'Школьники класса' для реализации
    СВЯЗИ МНОГИЕ-КО-МНОГИМ
    """

    def __init__(self, cls_id, pup_id):
        self.cls_id = cls_id
        self.pup_id = pup_id
```

```
# Классы
class = [
    Cls(1, '11А'),
    Cls(2, '7Б'),
    Cls(3, '5В'),

    Cls(11, '11Б'),
    Cls(22, '7В'),
    Cls(33, '5А'),
]

# Школьники
pups = [
    Pup(1, 'Абуховский', 5000, 1),
    Pup(2, 'Рыжкова', 0, 2),
    Pup(3, 'Зелинский', 10000, 2),
    Pup(4, 'Бондаренко', 10000, 3),
    Pup(5, 'Смыслов', 30000, 3),
]

pups_cls = [
    PupCls(1, 1),
    PupCls(2, 2),
    PupCls(2, 3),
    PupCls(3, 4),
    PupCls(3, 5),
```

```

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

def main():

    # Соединение данных один-ко-многим
    one_to_many = [(p.fio, p.dbt, c.name)
                    for c in clss
                    for p in pups
                    if p.cls_id == c.id]

    # Соединение данных многие-ко-многим
    many_to_many_temp = [(c.name, pc.cls_id, pc.pup_id)
                          for c in clss
                          for pc in pups_clss
                          if c.id == pc.cls_id]

    many_to_many = [(p.fio, p.dbt, cls_name)
                     for cls_name, cls_id, pup_id in many_to_many_temp
                     for p in pups if p.id == pup_id]

    print('Задание A1')
    res_1 = sorted(one_to_many, key=itemgetter(2))
    print(res_1)

    print('\nЗадание A2')
    res_2_unsorted = []
    # Перебираем все классы
    for c in clss:
        # Список школьников класса
        c_pups = list(filter(lambda i: i[2] == c.name, one_to_many))
        # Если класс не пустой
        if len(c_pups) > 0:
            # Долги за обучение школьников класса
            c_dbts = [dbt for _, dbt, _ in c_pups]
            # Суммарный долг школьников класса
            c_dbts_sum = sum(c_dbts)
            res_2_unsorted.append((c.name, c_dbts_sum))

    # Сортировка по суммарному долгу
    res_2 = sorted(res_2_unsorted, key=itemgetter(1), reverse=True)
    print(res_2)

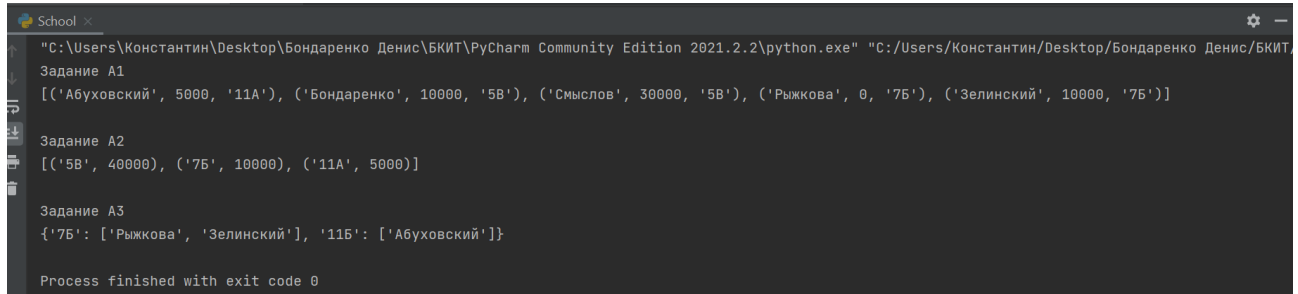
    print('\nЗадание A3')
    res_3 = {}
    # Перебираем все классы
    for c in clss:
        if 'Б' in c.name:
            # Список школьников класса
            c_pups = list(filter(lambda i: i[2] == c.name, many_to_many))
            # Только ФИО школьников
            c_pups_names = [x for x, _, _ in c_pups]
            # Добавляем результат в словарь
            # ключ - класс, значение - список фамилий
            res_3[c.name] = c_pups_names

    print(res_3)

```

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

Результаты выполнения программы



```
"C:\Users\Константин\Desktop\Бондаренко Денис\БКИТ\PyCharm Community Edition 2021.2.2\python.exe" "C:/Users/Константин/Desktop/Бондаренко Денис/БКИТ/...  
Задание A1  
[('Абуховский', 5000, '11А'), ('Бондаренко', 10000, '5В'), ('Смыслов', 30000, '5В'), ('Рыжкова', 0, '7Б'), ('Зелинский', 10000, '7Б')]  
Задание A2  
[('5В', 40000), ('7Б', 10000), ('11А', 5000)]  
Задание A3  
{'7Б': ['Рыжкова', 'Зелинский'], '11Б': ['Абуховский']}  
Process finished with exit code 0
```

В задании A1 («Класс» и «Школьник» связаны соотношением один-ко-многим) выводится список всех связанных школьников и классов, отсортированный по классам, сортировка по школьникам произвольная.

В задании A2 («Класс» и «Школьник» связаны соотношением один-ко-многим) выводится список классов с суммарным долгом школьников за обучение каждого класса, отсортированный по суммарному долгу.

В задании A3 («Класс» и «Школьник» связаны соотношением многие-ко-многим) выводится список всех классов, у которых в названии присутствует буква «Б», и список учащихся в них школьников.