

**Московский государственный технический
университет им. Н.Э. Баумана.**

Факультет «Информатика и управление»
Кафедра «Системы обработки информации и управления»

Курс «Разработка интернет-приложений»
Отчет по рубежному контролю №1

Выполнил:
студент группы ИУ5-51Б
Алехин Сергей
Подпись и дата:

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю.Е.
Подпись и дата:

г. Москва, 2020 г.

Алехин Сергей ИУ5-51Б Вариант А2

1. «Класс» и «Школьник» связаны соотношением один-ко-многим. Выведите список всех связанных школьников и классов, отсортированный по классам, сортировка по школьникам произвольная.
2. «Класс» и «Школьник» связаны соотношением один-ко-многим. Выведите список классов с суммарной стоимостью обучения школьников в каждом классе, отсортированный по суммарной стоимости обучения.
3. «Класс» и «Школьник» связаны соотношением многие-ко-многим. Выведите список всех классов, у которых в названии присутствует слово «класс», и список учащихся в них школьников.

Текст программы

main.py

```
from operator import itemgetter
from rk_python_oop.Class import Class
from rk_python_oop.Pupil import Pupil
from rk_python_oop.ClassPupil import ClassPupil
```

```
# Классы
```

```
classes = [
    Class(1, '1 класс'),
    Class(2, '2 класс'),
    Class(3, 'баскетбол'),

    Class(11, '3 класс'),
    Class(22, 'футбол'),
    Class(33, 'футбол'),
]
```

```
# Школьники
```

```
pupils = [
    Pupil(1, 'Кирсанчик', 40000, 1),
    Pupil(2, 'Валерчик', 35000, 2),
    Pupil(3, 'Денчик', 45000, 3),
    Pupil(4, 'Егорчик', 35000, 3),
    Pupil(5, 'Иванчик', 40000, 3),
]
```

```
# Ученики класса
```

```
classes_pupils = [
    ClassPupil(1, 1),
    ClassPupil(2, 2),
    ClassPupil(3, 3),
    ClassPupil(3, 4),
    ClassPupil(3, 5),

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

```
def main():
```

```

# Соединение данных один-ко-многим
one_to_many = [(e.fio, e.cost, d.name)
                for d in classes
                for e in pupils
                if e.class_id == d.id]

# Соединение данных многие-ко-многим
many_to_many_temp = [(d.name, ed.class_id, ed.pupil_id)
                      for d in classes
                      for ed in classes_pupils
                      if d.id == ed.class_id]

many_to_many = [(e.fio, e.cost, dep_name)
                 for dep_name, class_id, pupil_id in many_to_many_temp
                 for e in pupils if e.id == pupil_id]

# Задание 1
print('Задание A1')
res_1 = sorted(one_to_many, key=itemgetter(2))
print(res_1)

# Задание 2
print('Задание A2')
res_2_unsorted = []
for d in classes:
    d_pupils = list(filter(lambda i: i[2] == d.name, one_to_many))
    if len(d_pupils) > 0:
        d_cost = [cost for _, cost, _ in d_pupils]
        d_cost_sum = sum(d_cost)
        res_2_unsorted.append((d.name, d_cost_sum))

res_2 = sorted(res_2_unsorted, key=itemgetter(1), reverse=True)
print(res_2)

# Задание 3
print('Задание A3')
res_3 = {}
for d in classes:
    if 'класс' in d.name:
        d_pupils = list(filter(lambda i: i[2] == d.name, many_to_many))
        d_pupils_names = [x for x, _, _ in d_pupils]
        res_3[d.name] = d_pupils_names

print(res_3)

if __name__ == '__main__':
    main()

```

Pupil.py

```

class Pupil:
    """ШКОЛЬНИК"""

    def __init__(self, id, fio, cost, class_id):
        self.id = id
        self.fio = fio
        self.cost = cost
        self.class_id = class_id

```

Class.py

```

class Class:
    """Класс"""

```

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

ClassPipul.py

```
class ClassPupil:  
    """Ученики класса"""  
  
    def __init__(self, class_id, pupil_id):  
        self.class_id = class_id  
        self.pupil_id = pupil_id
```

Примеры работы программы

Задание А1

```
[('Кирсанчик', 40000, '1 класс'), ('Валерчик', 35000, '2 класс'), ('Денчик', 45000, 'баскетбол'),  
(('Егорчик', 35000, 'баскетбол'), ('Иванчик', 40000, 'баскетбол'))]
```

Задание А2

```
[('баскетбол', 120000), ('1 класс', 40000), ('2 класс', 35000)]
```

Задание А3

```
{'1 класс': ['Кирсанчик'], '2 класс': ['Валерчик'], '3 класс': ['Кирсанчик']}
```