

Отчет по РК № 1 по курсу
"Разработка Интернет-Приложений"

Выполнил:

Студент группы

ИУ5-55Б

Бахман А.А.

Задание: Вариант Д.

1. «Школьник» и «Класс» связаны соотношением один-ко-многим. Выведите список всех школьников, у которых фамилия заканчивается на «ов», и названия их классов.
2. «Школьник» и «Класс» связаны соотношением один-ко-многим. Выведите список классов со средней оценкой школьников в каждом классе, отсортированный, по средней оценке, (*отдельной функции вычисления среднего значения в Python нет, нужно использовать комбинацию функций вычисления суммы и количества значений*).
3. «Класс» и «Школьник» связаны соотношением многие-ко-многим. Выведите список всех классов, у которых название начинается с цифры «5», и список обучающихся в них школьников.

№ варианта	Класс 1	Класс 2
1	Студент	Группа
2	Школьник	Класс

Текст программы:

```
# Бахман Александр ИУ5-55Б
# Запрос Д
# Предметная область 2 - Школьник и Класс

# используется для сортировки
from operator import itemgetter

class Stud:
    """Школьник"""
    def __init__(self, id, fio, mark, class_id):
        self.id = id
        self.fio = fio
        self.mark = mark
        self.class_id = class_id

class Class:
    """Класс"""
    def __init__(self, id, name):
        self.id = id
        self.name = name

class StudClass:
    """
```

```

    'Школьники в классе' для реализации
    связи многие-ко-многим
    """
    def __init__(self, class_id, stud_id):
        self.class_id = class_id
        self.stud_id = stud_id

# Классы
classes = [
    Class(1, '1Г'),
    Class(2, '2А'),
    Class(3, '3Б'),
    Class(4, '4В'),
    Class(5, '5А'),
    Class(6, '5Г'),
    Class(7, '7А')
]

# Школьники
studs = [
    Stud(1, 'Зубарева', 5, 1),
    Stud(2, 'Петров', 4.3, 2),
    Stud(3, 'Финк', 4.2, 3),
    Stud(4, 'Турчин', 4.59, 3),
    Stud(5, 'Амрахов', 3.8, 4),
    Stud(6, 'Бахман', 4.5, 5),
    Stud(7, 'Яковлев', 4.4, 6),
    Stud(8, 'Озеров', 4.9, 6),
    Stud(9, 'Рябова', 4.9999, 5),
    Stud(10, 'Кожиев', 4.88, 5),
    Stud(11, 'Шкарин', 4.88, 6),
    Stud(12, 'Хижняков', 4.999999, 5),
]

studs_classes = [
    StudClass(1,1),
    StudClass(2,2),
    StudClass(3,3),
    StudClass(3,4),
    StudClass(3,5),

    StudClass(4,5),
    StudClass(4,6),
    StudClass(5,7),
    StudClass(5,8),
    StudClass(5,9),
    StudClass(6,10),
    StudClass(6,11),
    StudClass(6,12)
]

```

```

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

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

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

    many_to_many = [(e.fio, e.mark, dep_name)
                    for dep_name, class_id, stud_id in many_to_many_temp
                    for e in studs if e.id==stud_id]

    print('Задание D1')
    res1 = []
    for o in one_to_many:
        if o[0][-2:] == "ов":
            res1.append(o[0:3:2])
    print(res1)

    print('\nЗадание D2')
    res_12_unsorted = []
    # Перебираем все классы
    for d in classes:
        # Список школьников класса
        d_studs = list(filter(lambda i: i[2]==d.name, one_to_many))
        # Если класс не пустой
        if len(d_studs) > 0:
            # Средние оценки школьников класса
            d_marks = [mark for _,mark,_ in d_studs]
            # Суммарная средняя оценка класса
            d_marks_sum = sum(d_marks)
            d_marks_count = len(d_marks)
            d_marks_average = d_marks_sum / d_marks_count
            res_12_unsorted.append((d.name, round(d_marks_average,4)))

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

    print('\nЗадание D3')
    res_13 = {}
    # Перебираем все классы

```

```

for d in classes:
    if d.name[0] == '5':
        # Список школьников класса
        d_studs = list(filter(lambda i: i[2]==d.name, many_to_many))
        # Только ФИО школьников
        d_studs_names = [x for x,_,_ in d_studs]
        # Добавляем результат в словарь
        # ключ - класс, значение - список фамилий
        res_13[d.name] = d_studs_names

print(res_13)

if __name__ == '__main__':
    main()

```

Результат работы:

```

p/Study/5 sem/Python/pk1/main.py"
Задание D1
[('Петров', '2А'), ('Амрахов', '4В'), ('Хижняков', '5А'), ('Озеров', '5Г')]

Задание D2
[('1Г', 5.0), ('5А', 4.845), ('5Г', 4.7267), ('3Б', 4.395), ('2А', 4.3), ('4В', 3.8)]

Задание D3
{'5А': ['Яковлев', 'Озеров', 'Рябова'], '5Г': ['Кожиев', 'Шкарин', 'Хижняков']}
PS C:\Users\Xiaomi> 

```