

## ИУ5-35Б Бенц Ян БКИТ РК1

```
# используется для сортировки
from operator import itemgetter

# Задания Д вариант 2

class Student:
    """ШКОЛЬНИК"""

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

class Class:
    """Класс"""

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

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

    def __init__(self, class_id, student_id):
        self.class_id = class_id
        self.student_id = student_id

# Классы
classes = [
    Class(1, '5Б'),
    Class(2, '8А'),
    Class(3, '2Г'),
    Class(4, '11В'),
    Class(5, '9А'),
    Class(6, '7В')
]

# ШКОЛЬНИКИ
students = [
    Student(1, 'Иванова', 5, 2),
    Student(2, 'Бенц', 4, 5),
    Student(3, 'Афанасьев', 2, 3),
    Student(4, 'Большаков', 1, 6),
    Student(5, 'Коновалов', 4, 1),
    Student(6, 'Лаптев', 5, 1),
```

```

        Student(7, 'Козявкин', 5, 4),
        Student(8, 'Смирнова', 3, 3),
        Student(9, 'Воронцова', 3, 6),

    ]

students_classes = [
    StudentClass(1, 2),
    StudentClass(2, 5),
    StudentClass(3, 3),
    StudentClass(4, 6),
    StudentClass(5, 1),
    StudentClass(6, 1),
    StudentClass(7, 4),
    StudentClass(8, 3),
    StudentClass(9, 6),
]

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

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

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

    many_to_many = [(e.fio, e.last_mark, class_name)
                     for class_name, class_id, student_id in
many_to_many_temp
                     for e in students if e.id == student_id]

    print('---Задание A1---')
    res_a1 = {}
    for s in students:
        if s.fio[len(s.fio) - 1] == 'в' and s.fio[len(s.fio) - 2] ==
'o':
            print(s.fio, classes[(s.class_id) - 1].name)

    print('\n---Задание A2---')
    res_12_unsorted = []
    # Перебираем все классы
    for d in classes:
        # Список школьников в классе
        d_students = list(filter(lambda i: i[2] == d.name,
one_to_many))
        # Если класс не пустой
        if len(d_students) > 0:

```

```

        # Последняя оценка школьника в классе
        d_last_grade = [last_grade for _, last_grade, _ in
d_students]

        # Суммарная последняя оценка школьника в классе
        d_last_grade_sum = sum(d_last_grade)\
        # Средняя последняя оценка школьника в классе
        d_last_grade_av = d_last_grade_sum / len(d_last_grade)
        res_12_unsorted.append((d.name, d_last_grade_av))

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

print('\n---Задание А3---')
res_13 = {}
# Перебираем все классы
for c in classes:
    if c.name[len(c.name) - 1] == 'А':
        # Список школьников в класса
        d_students = list(filter(lambda i: i[2] == c.name,
many_to_many))
        # Только ФИО школьника
        d_student_names = [x for x, _, _ in d_students]
        # Добавляем результат в словарь
        # ключ - класс, значение - список фамилий
        res_13[c.name] = d_student_names

print(res_13)

if __name__ == '__main__':
    main()

```

The screenshot shows an IDE window with a Python file named `test.py`. The code defines a `Student` class and processes a list of students. The output window shows the results of the execution, including the output of the `print` statements.

```

# используется для сортировки
from operator import itemgetter

# Задания Д вариант 2

class Student:
    """Школьник"""

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

Run: test
"C:\Users\Yan note\Desktop\6y6a\venv\Scripts\python.exe" "C:\Users\Yan note\Desktop\test.py"
---Задание А1---
Большаков 7Б
Коновалов 5Б

---Задание А2---
[('8А', 5.0), ('11В', 5.0), ('5Б', 4.5), ('9А', 4.0), ('2Г', 2.5), ('7Б', 2.0)]

---Задание А3---
{'8А': ['Коновалов'], '9А': ['Иванова']}

Process finished with exit code 0

```