

предметная область: "школьник и класс"

текст программы

```
from operator import itemgetter

class Student:
    """Школьник"""
    def __init__(self, id, name, grade, class_id):
        self.id = id
        self.name = name
        self.grade = grade
        self.class_id = class_id

class SchoolClass:
    """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 = [
    SchoolClass(1, 'Класс А'),
    SchoolClass(2, 'Класс Б'),
    SchoolClass(3, 'Класс В'),
    SchoolClass(4, 'Математический класс'),
    SchoolClass(5, 'Физический класс'),
]

# Школьники
students = [
    Student(1, 'Алексей', 90, 1),
    Student(2, 'Мария', 85, 2),
    Student(3, 'Сергей', 92, 3),
    Student(4, 'Николай', 88, 3),
    Student(5, 'Анна', 95, 1),
]
```

```

students_classes = [
    StudentClass(1, 1),
    StudentClass(2, 2),
    StudentClass(3, 3),
    StudentClass(3, 4),
    StudentClass(1, 5),
    StudentClass(4, 1),
    StudentClass(5, 2),
    StudentClass(4, 3),
    StudentClass(5, 4),
]

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

    # Соединение данных один-ко-многим
    one_to_many = [(s.name, s.grade, c.name)
                    for c in classes
                    for s in students
                    if s.class_id == c.id]

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

    many_to_many = [(s.name, s.grade, class_name)
                     for class_name, class_id, student_id in many_to_many_temp
                     for s in students if s.id == student_id]

    # Задание A1: Список всех школьников и их классов, отсортированный по классам
    print("Задание A1")
    res_a1 = sorted(one_to_many, key=itemgetter(2))
    print(res_a1)

    # Задание A2: Список классов с суммарными оценками школьников в каждом классе,
    # отсортированный по суммарной оценке
    print("\nЗадание A2")
    res_a2_unsorted = []
    for c in classes:
        class_students = list(filter(lambda i: i[2] == c.name, one_to_many))
        if class_students:
            total_grades = sum([grade for _, grade, _ in class_students])
            res_a2_unsorted.append((c.name, total_grades))
    res_a2 = sorted(res_a2_unsorted, key=itemgetter(1), reverse=True)
    print(res_a2)

```

```
# Задание А3: Список классов, название которых содержит слово "класс", и школьники,
относящиеся к ним
```

```
print("\nЗадание А3")
res_a3 = {}
for c in classes:
    if 'класс' in c.name.lower():
        class_students = list(filter(lambda i: i[2] == c.name, many_to_many))
        student_names = [name for name, _, _ in class_students]
        res_a3[c.name] = student_names
print(res_a3)

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

Результат выполнения программы

Задание А1

[('Алексей', 90, 'Класс А'), ('Анна', 95, 'Класс А'), ('Мария', 85, 'Класс Б'), ('Сергей', 92, 'Класс В'), ('Николай', 88, 'Класс В')]

Задание А2

[('Класс А', 185), ('Класс В', 180), ('Класс Б', 85)]

Задание А3

{'Класс А': ['Алексей', 'Анна'], 'Класс Б': ['Мария'], 'Класс В': ['Сергей', 'Николай'], 'Математический класс': ['Алексей', 'Сергей'], 'Физический класс': ['Мария', 'Николай']}

Дата разработки: 02.11.2024

Подпись: