

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

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

Курс «Парадигмы и конструкции языков программирования»  
Отчет по рубежному контролю №1  
**«Вариант А, 2»**

Выполнил:  
Студент группы ИУ5-31Б  
Баринов Егор

Проверил:  
Гапанюк Ю. Е.

2025 г.

# Листинг программы

```
1 from operator import itemgetter
2
3
4 5 usages
5 class Student:
6     def __init__(self, id, last_name, score, class_id):
7         self.id = id
8         self.last_name = last_name
9         self.score = score # количественный признак (баллы)
10        self.class_id = class_id # для связи один-ко-многим
11
12 6 usages
13 class SchoolClass:
14     def __init__(self, id, name):
15         self.id = id
16         self.name = name
17
18 10 usages
19 class StudentClass:
20     def __init__(self, class_id, student_id):
21         self.class_id = class_id
22         self.student_id = student_id
23
24 classes = [
25     SchoolClass(id=1, name='математический класс'),
26     SchoolClass(id=2, name='физический класс'),
27     SchoolClass(id=3, name='химический класс'),
28     SchoolClass(id=11, name='математический кружок'),
29     SchoolClass(id=22, name='физический кружок'),
30     SchoolClass(id=33, name='химический кружок'),
31 ]
32
33 students = [
34     Student(id=1, last_name='Иванов', score=85, class_id=1),
35     Student(id=2, last_name='Петров', score=92, class_id=2),
36     Student(id=3, last_name='Сидоров', score=78, class_id=3),
37     Student(id=4, last_name='Козлов', score=95, class_id=1),
38     Student(id=5, last_name='Смирнов', score=88, class_id=2),
39 ]
40
41 students_classes = [
42     StudentClass(class_id=1, student_id=1),
43     StudentClass(class_id=2, student_id=2),
44     StudentClass(class_id=3, student_id=3),
```

```

45     StudentClass( class_id: 1, student_id: 4),
46     StudentClass( class_id: 2, student_id: 5),
47     StudentClass( class_id: 11, student_id: 1),
48     StudentClass( class_id: 22, student_id: 2),
49     StudentClass( class_id: 33, student_id: 3),
50     StudentClass( class_id: 11, student_id: 4),
51     StudentClass( class_id: 22, student_id: 5),]
52
53
54 1 usage
55 def main():
56
57     one_to_many = [(s.last_name, s.score, c.name)
58                     for c in classes
59                     for s in students
60                     if s.class_id == c.id]
61
62     many_to_many_temp = [(c.name, sc.class_id, sc.student_id)
63                           for c in classes
64                           for sc in students_classes
65                           if c.id == sc.class_id]
66
67     many_to_many = [(s.last_name, s.score, class_name)
68                     for class_name, class_id, student_id in many_to_many_temp
69                     for s in students if s.id == student_id]
70
71     print('Задание A1')
72     print('Список всех связанных школьников и классов, отсортированный по классам:')
73     res_11 = sorted(one_to_many, key=itemgetter(2))
74     for item in res_11:
75         print(f'Школьник: {item[0]}, Баллы: {item[1]}, Класс: {item[2]}')
76
77     print('\n' + '=' * 50 + '\n')
78
79     print('Задание A2')
80     print('Список классов с суммарными баллами школьников, отсортированный по суммарным баллам:')
81     res_12_unsorted = []
82
83     class_groups = {}
84     for s_name, s_score, c_name in one_to_many:
85         if c_name not in class_groups:
86             class_groups[c_name] = []
87         class_groups[c_name].append(s_score)
88
89     for c_name, scores in class_groups.items():
90         total_score = sum(scores)

```

```

90     res_12_unsorted.append((c_name, total_score))
91
92     res_12 = sorted(res_12_unsorted, key=itemgetter(1), reverse=True)
93     for item in res_12:
94         print(f'Класс: {item[0]}, Суммарные баллы: {item[1]}')
95
96     print('\n' + '=' * 50 + '\n')
97
98     print('Задание A3')
99     print('Список всех классов, у которых в названии присутствует слово "класс", и список школьников в них:')
100     res_13 = {}
101
102
103     for c in classes:
104         if 'класс' in c.name.lower():
105             # Фильтруем школьников для данного класса из связи многие-ко-многим
106             c_students = [s_name for s_name, s_score, class_name in many_to_many
107                           if class_name == c.name]
108             res_13[c.name] = c_students
109
110     for class_name, students_list in res_13.items():
111         print(f'Класс: {class_name}')
112         print(f'Школьники: {"", ".join(students_list)}')
113
114
115 if __name__ == '__main__':
116     main()

```

# Результат выполнения

```
D:\analytics\.venv\пикап\Scripts\python.exe D:\пикап\rk1_varA2.py
Задание A1
Список всех связанных школьников и классов, отсортированный по классам:
Школьник: Иванов, Баллы: 85, Класс: математический класс
Школьник: Козлов, Баллы: 95, Класс: математический класс
Школьник: Петров, Баллы: 92, Класс: физический класс
Школьник: Смирнов, Баллы: 88, Класс: физический класс
Школьник: Сидоров, Баллы: 78, Класс: химический класс

=====

Задание A2
Список классов с суммарными баллами школьников, отсортированный по суммарным баллам:
Класс: математический класс, Суммарные баллы: 180
Класс: физический класс, Суммарные баллы: 180
Класс: химический класс, Суммарные баллы: 78

=====

Задание A3
Список всех классов, у которых в названии присутствует слово "класс", и список школьников в них:
Класс: математический класс
    Школьники: Иванов, Козлов
Класс: физический класс
    Школьники: Петров, Смирнов
Класс: химический класс
    Школьники: Сидоров

Process finished with exit code 0
```