

Текст main.py

```
main.py > Course > __init__
1  from operator import itemgetter
2
3  class StudentGroup:
4      """Студенческая группа"""
5
6      def __init__(self, id, name):
7          self.id = id
8          self.name = name
9
10 class Course:
11     """Учебный курс"""
12
13     def __init__(self, id, title, avg_grade, group_id):
14         self.id = id
15         self.title = title
16         self.avg_grade = avg_grade
17         self.group_id = group_id
18
19 def task1(groups, courses):
20     """
21     Возвращает список всех студентов и курсов, отсортированный по курсам.
22     """
23     one_to_many = [(course.title, group.name)
24                    for group in groups
25                    for course in courses
26                    if course.group_id == group.id]
27     return sorted(one_to_many, key=itemgetter(0))
28
29 def task2(groups, courses):
30     """
31     Возвращает список курсов с суммарной средней оценкой студентов.
32     """
33     course_avg_grades = {}
34
35     for group in groups:
36         group_courses = [course for course in courses if course.group_id == group.id]
37         for course in group_courses:
38             if course.title not in course_avg_grades:
39                 course_avg_grades[course.title] = []
40             course_avg_grades[course.title].append(course.avg_grade)
41
42     return sorted([(course, sum(grades) / len(grades))
43                   for course, grades in course_avg_grades.items()], key=itemgetter(1), reverse=True)
44
45 def task3(groups, courses, keyword="Курс"):
46     """
47     Возвращает список курсов с указанным ключевым словом в названии и студентов на этих курсах.
48     """
49     result = []
50     courses_with_keyword = [course for course in courses if keyword in course.title]
51
52     for course in courses_with_keyword:
53         students_in_course = [group.name for group in groups if group.id == course.group_id]
54         result.append((course.title, students_in_course))
55
56     return result
```

```
56     return result
57
58 if __name__ == '__main__':
59     groups = [
60         StudentGroup(1, 'Группа по экономике'),
61         StudentGroup(2, 'Группа по искусственному интеллекту'),
62         StudentGroup(3, 'Группа по биотехнологиям'),
63         StudentGroup(4, 'Группа по инженерии'),
64         StudentGroup(5, 'Группа по медицине'),
65         StudentGroup(6, 'Группа по химии'),
66     ]
67
68     courses = [
69         Course(1, 'Курс Основы экономики', 85, 1),
70         Course(2, 'Машинное обучение', 90, 2),
71         Course(3, 'Биотехнологии и устойчивое развитие', 75, 3),
72         Course(4, 'Механика материалов', 80, 4),
73         Course(5, 'Курс Анатомия человека', 88, 5),
74         Course(6, 'Органическая химия', 93, 6),
75         Course(7, 'Финансовый менеджмент', 87, 1),
76         Course(8, 'Нейронные сети', 76, 2),
77         Course(9, 'Биохимия', 84, 3),
78         Course(10, 'Системы управления проектами', 92, 4),
79         Course(11, 'Фармакология', 89, 5),
80         Course(12, 'Неорганическая химия', 78, 6),
81     ]
82
83     print('Задание 1')
84     for course, group in task1(groups, courses):
85         print(f"{course} - {group}")
86
87     print('\nЗадание 2')
88     for course, avg_grade in task2(groups, courses):
89         print(f"{course} - {avg_grade:.2f}")
90
91     print('\nЗадание 3')
92     for course, students in task3(groups, courses):
93         print(f"{course}: {' '.join(students)}")
```

Результат main.py

```
/usr/local/bin/python3 /Users/min/Desktop/rk/main.py
● min@MacBook-Pro-cua-Do rk % /usr/local/bin/python3 /Users/min/Desktop/
Задание 1
Биотехнологии и устойчивое развитие – Группа по биотехнологиям
Биохимия – Группа по биотехнологиям
Курс Анатомия человека – Группа по медицине
Курс Основы экономики – Группа по экономике
Машинное обучение – Группа по искусственному интеллекту
Механика материалов – Группа по инженерии
Нейронные сети – Группа по искусственному интеллекту
Неорганическая химия – Группа по химии
Органическая химия – Группа по химии
Системы управления проектами – Группа по инженерии
Фармакология – Группа по медицине
Финансовый менеджмент – Группа по экономике

Задание 2
Органическая химия – 93.00
Системы управления проектами – 92.00
Машинное обучение – 90.00
Фармакология – 89.00
Курс Анатомия человека – 88.00
Финансовый менеджмент – 87.00
Курс Основы экономики – 85.00
Биохимия – 84.00
Механика материалов – 80.00
Неорганическая химия – 78.00
Нейронные сети – 76.00
Биотехнологии и устойчивое развитие – 75.00

Задание 3
Курс Основы экономики: Группа по экономике
Курс Анатомия человека: Группа по медицине
○ min@MacBook-Pro-cua-Do rk %
```

Текст test.py

```
test.py > testCourseTasks > test_task2
1  import unittest
2  from main import StudentGroup, Course, task1, task2, task3
3
4
5  class TestCourseTasks(unittest.TestCase):
6
7      def setUp(self):
8          self.groups = [
9              StudentGroup(1, 'Группа по экономике'),
10             StudentGroup(2, 'Группа по искусственному интеллекту'),
11         ]
12
13         self.courses = [
14             Course(1, 'Курс Основы экономики', 85, 1),
15             Course(2, 'Машинное обучение', 90, 2),
16             Course(3, 'Курс Финансовый менеджмент', 87, 1),
17         ]
18
19     def test_task1(self):
20         expected = [
21             ('Курс Основы экономики', 'Группа по экономике'),
22             ('Курс Финансовый менеджмент', 'Группа по экономике'),
23             ('Машинное обучение', 'Группа по искусственному интеллекту'),
24         ]
25         result = task1(self.groups, self.courses)
26         self.assertEqual(result, expected)
27
28     def test_task2(self):
29         expected = [
30             ('Машинное обучение', 90.0),
31             ('Курс Финансовый менеджмент', 87.0),
32             ('Курс Основы экономики', 85.0),
33         ]
34         result = task2(self.groups, self.courses)
35         self.assertEqual(result, expected)
36
37     def test_task3(self):
38         expected = [
39             ('Курс Основы экономики', ['Группа по экономике']),
40             ('Курс Финансовый менеджмент', ['Группа по экономике']),
41         ]
42         result = task3(self.groups, self.courses, keyword="Курс")
43         self.assertEqual(result, expected)
44
45
46 if __name__ == '__main__':
47     unittest.main()
```

Результат test.py

```
min@MacBook-Pro-cua-Do rk % /usr/local/bin/python3 /Users/min/Desktop/rk/test.py
...
-----
Ran 3 tests in 0.000s

OK
min@MacBook-Pro-cua-Do rk %
```