Текст программы

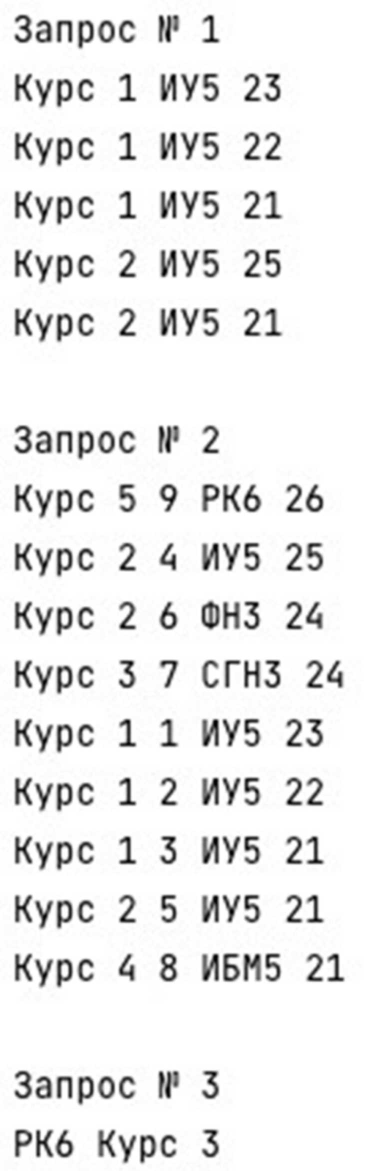
main

```
 class Group:     def __init__(self, id, name, sem,
count_students):
        self.id = id
self.name = name
self.sem = sem
        self.count_students = count_students
  class Course:     def __init__(self, id,
id_group, title):
        self.id = id
self.title = title
self.id_group = id_group
  class GroupCourse:     def __init__(self,
id_group, id_course):
        self.id_group = id_group
self.id_course = id_course
  def get_courses_by_group(groups, courses, group_name):     return [(c, g)
for c in courses for g in groups if c.id_group == g.id and
c.title.startswith("Курс") and g.name == group_name]
  def get_sorted_courses_by_group(groups,
courses):
    data = [(c, g) for c in courses for g in groups if c.id_group == g.id]
return sorted(data, key=lambda x: x[1].count_students, reverse=True)
  def get_filtered_data(groups, courses, groups_courses,
letter):
    return [(g, c, gc) for gc in groups_courses for g in groups for c in
courses if
            gc.id_group == g.id and gc.id_course == c.id and letter in g.name]
  def main():
groups = [
        Group(1, "ИУ5", 2, 23),
        Group(2, "ИУ5", 1, 22),
        Group(3, "ИУ5", 1, 21),
        Group(4, "ИУ5", 3, 25),
        Group(5, "ИУ5", 4, 21),
        Group(6, "ФН3", 3, 24),
        Group(7, "СГН3", 5, 24),
        Group(8, "ИБМ5", 7, 21),
        Group(9, "РК6", 9, 26)
    ]
    courses = [
        Course(1, 1, "Курс 1"),
        Course(1, 2, "Курс 1"),
        Course(1, 3, "Курс 1"),
        Course(2, 4, "Курс 2"),
        Course(2, 5, "Курс 2"),
        Course(2, 6, "Курс 2"),
        Course(3, 7, "Курс 3"),
        Course(4, 8, "Курс 4"),
        Course(5, 9, "Курс 5")
    ]
    groups_courses = [
GroupCourse(1, 1),
        GroupCourse(1, 2),
        GroupCourse(1, 4),
        GroupCourse(2, 1),
        GroupCourse(3, 2),
        GroupCourse(4, 4),
        GroupCourse(5, 5),
        GroupCourse(9, 3)
    ]
    print("Запрос № 1")
    data = get_courses_by_group(groups, courses, "ИУ5")
for (c, g) in data:
        print(c.title, g.name, g.count_students)
print()

    print("Запрос № 2")
    data = get_sorted_courses_by_group(groups, courses)
for (c, g) in data:
        print(c.title, g.id, g.name, g.count_students)
print()

    print("Запрос № 3")
letter = 'Р'
    filtered_data = get_filtered_data(groups, courses, groups_courses,
letter)
    filtered_data.sort(key=lambda x: x[0].name)
for (g, c, _) in filtered_data:
        print(g.name, c.title)
  if __name__ ==
"__main__":
    main()
```

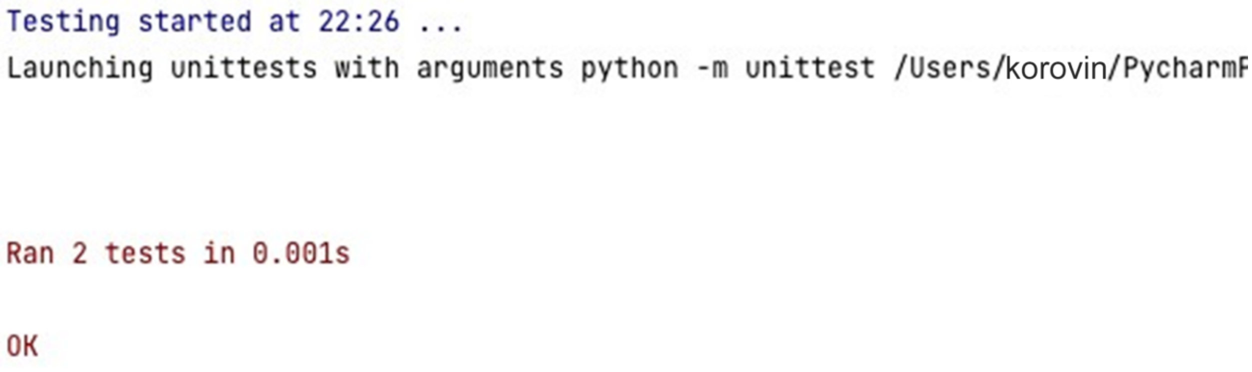
tests

```
import unittest
from unittest.mock import patch
from main import get_courses_by_group, get_sorted_courses_by_group,
get_filtered_data, Group, Course, GroupCourse
  class
TestProgram(unittest.TestCase):
def setUp(self):         self.groups
= [
            Group(1, "ИУ5", 2, 23),
            Group(2, "ИУ5", 1, 22),
            Group(3, "ИУ5", 1, 21),
            Group(4, "ИУ5", 3, 25),
            Group(5, "ИУ5", 4, 21),
            Group(6, "ФН3", 3, 24),
            Group(7, "СГН3", 5, 24),
            Group(8, "ИБМ5", 7, 21),
            Group(9, "РК6", 9, 26)
        ]
        self.courses = [
            Course(1, 1, "Курс 1"),
            Course(1, 2, "Курс 1"),
            Course(1, 3, "Курс 1"),
            Course(2, 4, "Курс 2"),
            Course(2, 5, "Курс 2"),
            Course(2, 6, "Курс 2"),
            Course(3, 7, "Курс 3"),
            Course(4, 8, "Курс 4"),
            Course(5, 9, "Курс 5")
        ]
        self.groups_courses = [
GroupCourse(1, 1),
            GroupCourse(1, 2),
            GroupCourse(1, 4),
            GroupCourse(2, 1),
            GroupCourse(3, 2),
            GroupCourse(4, 4),
            GroupCourse(5, 5),
            GroupCourse(9, 3)
        ]      def
test_get_courses_by_group(self):
        result = get_courses_by_group(self.groups, self.courses, "ИУ5")
self.assertEqual(result, [
            (self.courses[0], self.groups[0]),
            (self.courses[1], self.groups[1]),
            (self.courses[2], self.groups[2]),
            (self.courses[3], self.groups[3]),
            (self.courses[4], self.groups[4])
        ])      def
test_get_sorted_courses_by_group(self):
        result = get_sorted_courses_by_group(self.groups, self.courses)
self.assertEqual(result, [
            (self.courses[3], self.groups[4]),
            (self.courses[4], self.groups[3]),
            (self.courses[5], self.groups[0]),
            (self.courses[6], self.groups[5]),
            (self.courses[7], self.groups[6]),
            (self.courses[8], self.groups[8]),
            (self.courses[0], self.groups[1]),
            (self.courses[1], self.groups[2]),
            (self.courses[2], self.groups[7])
        ])      def
test_get_filtered_data(self):
        result = get_filtered_data(self.groups, self.courses,
self.groups_courses, 'P')         self.assertEqual(result, [
            (self.groups[8], self.courses[6], self.groups_courses[7])
        ])
  if __name__ ==
'__main__':
unittest.main()
```

Результаты выполнения



```
Запрос № 1
Курс 1 ИУ5 23
Курс 1 ИУ5 22
Курс 1 ИУ5 21
Курс 2 ИУ5 25
Курс 2 ИУ5 21

Запрос № 2
Курс 5 9 РК6 26
Курс 2 4 ИУ5 25
Курс 2 6 ФН3 24
Курс 3 7 СГН3 24
Курс 1 1 ИУ5 23
Курс 1 2 ИУ5 22
Курс 1 3 ИУ5 21
Курс 2 5 ИУ5 21
Курс 4 8 ИБМ5 21

Запрос № 3
РК6 Курс 3
```



```
Testing started at 22:26 ...
Launching unittests with arguments python -m unittest /Users/korovin/PycharmP



Ran 2 tests in 0.001s

OK
```