Казакова Валерия, ИУ5Ц-51Б, вариант 29

Вариант предметной области:

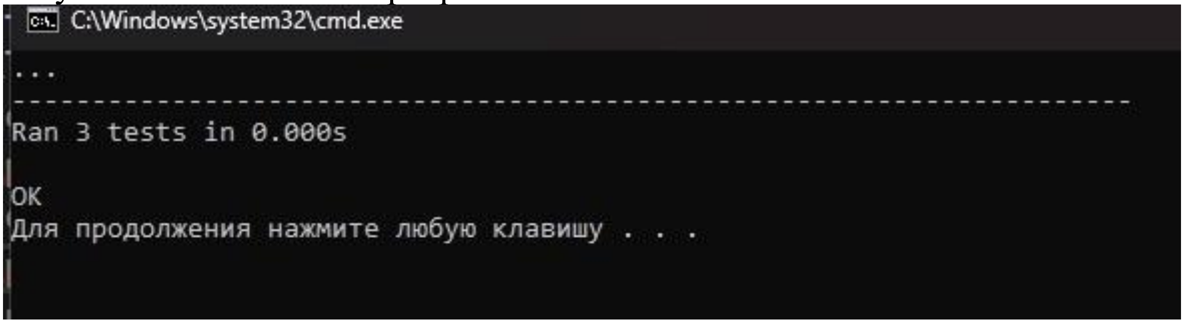
29	Кафедра	Факультет

Текст программы:

```
import unittest
from operator import itemgetter

class Department:
    def __init__(self, id, name, disciplines, faculty_id):
        self.id = id
        self.name = name
        self.disciplines = disciplines
        self.faculty_id = faculty_id

class Faculty:
    def __init__(self, faculty_id, name):
        self.id = faculty_id
        self.name = name

class FacultyDepartment:
    def __init__(self, department_id, faculty_id):
        self.department_id = department_id
        self.faculty_id = faculty_id

def get_departments_by_faculty(faculties, faculty_departments, departments):
    result = []
    for faculty in faculties:
        faculty_department_ids = [fd.department_id for fd in faculty_departments if
fd.faculty_id == faculty.id]
        related_departments = [department for department in departments if
department.id in faculty_department_ids]
        result.append((faculty.name, [department.name for department in
related_departments]))
    return result

def get_faculty_discipline_counts(faculties, faculty_departments, departments):
    result = []
    for faculty in faculties:
        faculty_department_ids = [fd.department_id for fd in faculty_departments if
fd.faculty_id == faculty.id]
        related_departments = [department for department in departments if
department.id in faculty_department_ids]
        disciplines_sum = sum([department.disciplines for department in
related_departments])
        result.append((faculty.name, disciplines_sum, [(department.name,
department.disciplines) for department in related_departments]))
    return result

def get_matching_faculties(faculties, faculty_contains_word, faculty_departments,
departments):
    result = []
    for faculty in faculties:
        if faculty_contains_word.lower() in faculty.name.lower():
            faculty_department_ids = [fd.department_id for fd in faculty_departments
if fd.faculty_id == faculty.id]
            related_departments = [department for department in departments if
department.id in faculty_department_ids]
            result.append((faculty.name, [department.name for department in
related_departments]))
    return result
```

```
class TestDepartmentFacultyQueries(unittest.TestCase):
    def setUp(self):
        self.departments = [
            Department(1, "Кафедра Информатики", 5, 1),
            Department(2, "Кафедра Систем управления летательных аппаратов", 3, 1),
            Department(3, "Кафедра Промышленного дизайна", 4, 2),
            Department(4, "Кафедра Физики", 2, 3),
            Department(5, "Кафедра Технологий обработки металлов", 6, 2)
        ]
        self.faculties = [
            Faculty(1, "Факультет Специального машиностроения"),
            Faculty(2, "Факультет Машиностроения"),
            Faculty(3, "Фундаментальные науки")
        ]
        self.faculty_departments = [
            FacultyDepartment(1, 1),
            FacultyDepartment(2, 1),
            FacultyDepartment(3, 2),
            FacultyDepartment(4, 3),
            FacultyDepartment(5, 2)
        ]

    def test_get_departments_by_faculty(self):
        result = get_departments_by_faculty(self.faculties,
self.faculty_departments, self.departments)
        expected_result = [
            ('Факультет Специального машиностроения', ['Кафедра Информатики',
'Кафедра Систем управления летательных аппаратов']),
            ('Факультет Машиностроения', ['Кафедра Промышленного дизайна', 'Кафедра
Технологий обработки металлов']),
            ('Фундаментальные науки', ['Кафедра Физики'])
        ]
        self.assertEqual(result, expected_result)

    def test_get_faculty_discipline_counts(self):
        result = get_faculty_discipline_counts(self.faculties,
self.faculty_departments, self.departments)
        expected_result = [
            ('Факультет Специального машиностроения', 8,
             [('Кафедра Информатики', 5), ('Кафедра Систем управления летательных
аппаратов', 3)]),
            ('Факультет Машиностроения', 10,
             [('Кафедра Промышленного дизайна', 4), ('Кафедра Технологий обработки
металлов', 6)]),
            ('Фундаментальные науки', 2,
             [('Кафедра Физики', 2)])
        ]
        self.assertEqual(result, expected_result)

    def test_get_matching_faculties(self):
        result = get_matching_faculties(self.faculties, "факультет",
self.faculty_departments, self.departments)
        expected_result = [
            ('Факультет Специального машиностроения', ['Кафедра Информатики',
'Кафедра Систем управления летательных аппаратов']),
            ('Факультет Машиностроения', ['Кафедра Промышленного дизайна', 'Кафедра
Технологий обработки металлов'])
        ]
        self.assertEqual(result, expected_result)

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

Результат выполнения программы:



```
...
----------------------------------------------------------------------
Ran 3 tests in 0.000s

OK
Для продолжения нажмите любую клавишу . . .
```