Вариант 26Б

Текст программы:

rk2.py:

```python
# Класс сущности "Студенческая группа"
class Group:

    def __init__(self, id, name, number_of_students):
        self.id = id  # id группы
        self.name = name  # Название группы
        self.number_of_students = number_of_students  # Кол-во студентов


# Класс сущности "Учебный курс"
class Course:

    def __init__(self, id, name):
        self.id = id  # id курса
        self.name = name  # Название курса


# Класс сущности "Студенческая группа - Учебный курс" (Для связи многие ко
многим)
class GroupCourse:

    def __init__(self, course_id, group_id):
        self.course_id = course_id  # id курса
        self.group_id = group_id  # id группы


# Функции для выполнения запросов
def query_b1(groups, courses): #Запрос Б1
    one_to_many = [(g.name, g.number_of_students, c.name)
                   for c in courses
                   for g in groups
                   if g.id == c.id]
    one_to_many.append((groups[4].name, groups[4].number_of_students,
courses[3].name))
    return sorted(one_to_many, key=lambda x: x[0])

def query_b2(groups, courses): #Запрос Б2
    b2 = []
    for course in courses:
        course_groups = list(filter(lambda i: i[2] == course.name,
query_b1(groups, courses)))
        if len(course_groups) > 0:
            course_students = [students for _, students, _ in course_groups]
            course_students_sum = sum(course_students)
            b2.append((course.name, course_students_sum))
    return sorted(b2, key=lambda x: x[1], reverse=True)

def query_b3(groups, courses, groups_courses): #Запрос Б3
    many_to_many_group = [(c.name, gc.course_id, gc.group_id)
                          for c in courses
                          for gc in groups_courses
                          if c.id == gc.course_id]

    many_to_many = [(g.name, g.number_of_students, name)
                    for name, _, group_id in many_to_many_group
                    for g in groups if g.id == group_id]
```

```python
    b3 = {}
    """
        Пояснение к запросу Б3:
        Вместо окончания "ов" в фамилии сотрудников,
        я взял окончание "B" в имени группы
    """
    for g in groups:
        if 'B' in g.name:
            course_groups = list(filter(lambda i: i[0] == g.name,
many_to_many))
            course_groups_names = [x for _, _, x in course_groups]
            b3[g.name] = course_groups_names
    return b3
```

TDD_test.py

```python
import unittest
from rk2 import Group, Course, GroupCourse, query_b1, query_b2, query_b3

class TestQueries(unittest.TestCase):
    def setUp(self):
        # Студенческие группы
        self.groups = [
            Group(1, 'IU1-12A', 26),
            Group(2, 'IU1-11B', 28),
            Group(3, 'IU3-21C', 20),
            Group(4, 'IU4-42D', 24),
            Group(5, 'IU5-44B', 19),
        ]
        # Учебные курсы
        self.courses = [
            Course(1, 'First'),
            Course(2, 'Second'),
            Course(3, 'Third'),
            Course(4, 'Fourth'),
        ]
        # Студенческая группа - Учебный курс
        self.groups_courses = [
            GroupCourse(1, 1),
            GroupCourse(1, 2),
            GroupCourse(2, 3),
            GroupCourse(4, 4),
            GroupCourse(4, 5),
            GroupCourse(3, 5),
        ]

    def test_query_b1(self):
        result = query_b1(self.groups, self.courses)
        self.assertEqual(len(result), 5)

    def test_query_b2(self):
        result = query_b2(self.groups, self.courses)
        self.assertEqual(len(result), 4)

    def test_query_b3(self):
        result = query_b3(self.groups, self.courses, self.groups_courses)
        self.assertEqual(len(result), 2)

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

Результат выполнения:

============================ test session starts ============================

collecting ... collected 3 items

TDD_test.py::TestQueries::test_query_b1 PASSED   [33%]
TDD_test.py::TestQueries::test_query_b2 PASSED   [66%]
TDD_test.py::TestQueries::test_query_b3 PASSED   [100%]

============================= 3 passed in 0.03s =============================