

**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 РК №2
Вариант запросов: В
Вариант предметной области: 1

Выполнил:
студент группы ИУ5-33Б
Абитов М. Р.

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю. Е.

Москва, 2023 г.

Вариант запросов В.

1. «Студент» и «Группа» связаны соотношением один-ко-многим. Выведите список всех деталей, которые начинаются с «Тормоз» и названия их производителей.
2. «Студент» и «Группа» связаны соотношением один-ко-многим. Выведите список производителей с минимальной стоимостью деталей, отсортированный их по минимальной стоимости.
3. «Студент» и «Группа» связаны соотношением многие-ко-многим. Выведите список всех связанных деталей и производителей, отсортированный по производителям, сортировка деталей по цене.

Условия рубежного контроля №2 по курсу ПиКЯП:

Рубежный контроль представляет собой разработку тестов на языке Python.

- 1) Проведите рефакторинг текста программы рубежного контроля №1 таким образом, чтобы он был пригоден для модульного тестирования.
- 2) Для текста программы рубежного контроля №1 создайте модульные тесты с применением TDD - фреймворка (3 теста).

Код программы main.py

```
class Student:
    def __init__(self, student_id, name, group_id, academic_rating):
        self.student_id = student_id
        self.name = name
        self.group_id = group_id
        self.academic_rating = academic_rating

class Group:
    def __init__(self, group_id, name):
        self.group_id = group_id
        self.name = name
```

```

def find_students_starting_with_A(student_group):
    return [(student.name, group.name) for student, group in student_group if
            student.name.startswith('A')]

def find_group_min_ratings(student_group):
    group_min_ratings = {}
    for student, group in student_group:
        if group.name in group_min_ratings:
            if student.academic_rating < group_min_ratings[group.name]:
                group_min_ratings[group.name] = student.academic_rating
        else:
            group_min_ratings[group.name] = student.academic_rating
    return sorted(group_min_ratings.items(), key=lambda x: x[1])

def list_students_groups(student_group):
    return [(student.name, group.name) for student, group in student_group]

def main():
    # Создаем объекты класса Group
    group1 = Group(1, "Группа А")
    group2 = Group(2, "Группа В")
    group3 = Group(3, "Группа С")

    # Создаем объекты класса Student
    student1 = Student(1, "Андреев", 1, 90)
    student2 = Student(2, "Борисов", 1, 85)
    student3 = Student(3, "Абрамов", 2, 92)
    student4 = Student(4, "Григорьев", 2, 88)
    student5 = Student(5, "Иванов", 3, 94)

    # Создаем список "Студенты и их группы" для связи один-ко-многим
    student_group = [
        (student1, group1),
        (student2, group1),
        (student3, group2),
        (student4, group2),
        (student5, group3)
    ]

    # Задание В1
    print("Задание В1")
    for name, group_name in find_students_starting_with_A(student_group):
        print(f"{name} - {group_name}")

    # Задание В2
    print("\nЗадание В2")
    for group, min_rating in find_group_min_ratings(student_group):
        print(f"{group} - Минимальный рейтинг: {min_rating}")

```

```

# Задание В3
print("\nЗадание В3")
for name, group_name in list_students_groups(student_group):
    print(f"{name} - {group_name}")
if __name__ == '__main__':
    main()

```

Код программы test_program.py

```

from main import Student, Group, find_students_starting_with_A,
find_group_min_ratings, list_students_groups
import unittest
import sys
import os

# Добавляем путь к папке RK2 в PYTHONPATH
sys.path.insert(0, os.path.abspath(
    os.path.join(os.path.dirname(__file__), '..')))

class TestProgram(unittest.TestCase):
    def setUp(self):
        self.group1 = Group(1, "Группа А")
        self.group2 = Group(2, "Группа В")
        self.student_group = [
            (Student(1, "Андреев", 1, 90), self.group1),
            (Student(2, "Борисов", 1, 85), self.group1),
            (Student(3, "Абрамов", 2, 92), self.group2)
        ]

    def test_find_students_starting_with_A(self):
        result = find_students_starting_with_A(self.student_group)
        self.assertEqual(len(result), 2)
        self.assertIn(("Андреев", "Группа А"), result)

    def test_find_group_min_ratings(self):
        result = find_group_min_ratings(self.student_group)
        self.assertEqual(result, [("Группа А", 85), ("Группа В", 92)])

    def test_list_students_groups(self):
        result = list_students_groups(self.student_group)
        self.assertEqual(len(result), 3)
        self.assertIn(("Борисов", "Группа А"), result)

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

```

Результат

```
PS C:\Users\GALIM\Desktop\ПикЯП\RK2> python -m unittest discover -s tests
...
-----
Ran 3 tests in 0.000s

OK
```