



**Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)**

**Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»**

**Рубежный контроль №2
по дисциплине «Базовые компоненты интернет-технологий»**

Выполнил:
студент группы
ИУ5Ц-52Б Дзауров И.А.
_____, “ ____ ” _____ 2022 г.

Проверил:
преподаватель кафедры
ИУ5 - Гапанюк Ю.Е.
_____, “ ____ ” _____ 2022 г.

Москва, 2022 г.

Описание задания

Вариант предметной области – 27 [«Преподаватель» - «Учебный курс»].

Вариант запросов – Б.

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).

Листинг программы

#teacher.py

```
import dataclasses
```

```
@dataclasses.dataclass
```

```
class Teacher:
```

```
    id: int
```

```
    name: str
```

#teacher_courses.py

```
import dataclasses
```

```
@dataclasses.dataclass
```

```
class TeacherCourse:
```

```
    teacher_id: int
```

```
    course_id: int
```

#course.py

```
import dataclasses
```

```
@dataclasses.dataclass
```

```
class Course:
```

```
    id: int
```

```
    name: str
```

```
    hours_num: float
```

```
    teacher_id: int
```

#queries.py

```
# 1. «Преподаватель» и «Курс» связаны соотношением один-ко-многим.
```

```
# Запрос выводит список все связанных курсов и преподавателей, отсортированный по  
числу часов в курсе
```

```
def query_1(teachers, courses):
```

```
    return [(course, next(teacher for teacher in teachers if teacher.id ==  
course.teacher_id)) for course
```

```
        in sorted([course for course in courses], key=lambda course:
```

```
course.hours_num)
```

```
        if any(course.teacher_id == teacher.id for teacher in teachers)]]
```

```
# 2. «Преподаватель» и «Курс» связаны соотношением один-ко-многим.
```

```
# Запрос выводит список преподавателей с числом курсов отсортированный по  
количеству
```

```
# курсов которые ведет преподаватель
```

```
def query_2(teachers, courses):
```

```
    return sorted(
```

```
        [(teacher, sum(course.teacher_id == teacher.id for course in courses)) for  
teacher in teachers],
```

```
        key=lambda t_s: sum(course.teacher_id == t_s[0].id for course in courses))
```

```
# 3. «Преподаватель» и «Курс» связаны соотношением многие-ко-многим.
```

```
# Запрос выводит список всех курсов, название которых оканчивается на  
"программирование",
```

```
# и преподавателей этих курсов
```

```
def query_3(teachers, courses, teacher_courses):
```

```
    return [(course,
```

```
        teachers[next(teacher_course for teacher_course in teacher_courses
```

```
        if teacher_course.course_id == course.id).teacher_id])
    for course in courses if course.name.endswith("программирование")]
```

#main.py

```
from teacher import Teacher
from course import Course
from teacher_courses import TeacherCourse
from queries import query_1, query_2, query_3

def main():
    teachers = [Teacher(0, "Иванов Иван Иванович"),
                Teacher(1, "Петрова Анна Ивановна"),
                Teacher(2, "Ильин Илья Ильич"),
                Teacher(3, "Петров Петр Петрович"),
                ]

    courses = [
        Course(0, "Математический анализ", 100.5, 1),
        Course(1, "Физика", 90, 2),
        Course(2, "Объектно-ориентированное программирование", 80, 0),
        Course(3, "Функциональное программирование", 94.5, 3),
        Course(4, "Веб-программирование", 120, 0),
    ]

    teacher_courses = [
        TeacherCourse(0, 2),
        TeacherCourse(0, 4),
        TeacherCourse(1, 0),
        TeacherCourse(2, 1),
        TeacherCourse(3, 3),
    ]

    print("Список преподавателей:")
    for i in teachers:
        print(i)

    print("Список курсов:")
    for i in courses:
        print(i)

    print("Список курсов связанных с преподавателем:")
    for i in teacher_courses:
        print(i)

    print("Задание Б1")
    for i in query_1(teachers, courses):
        print(i)
    print("Задание Б2")
    for i in query_2(teachers, courses):
        print(i)
    print("Задание Б3")
    for i in query_3(teachers, courses, teacher_courses):
        print(i)
    # print(query_1(teachers, courses))
    # print(query_2(teachers, courses))
    # print(query_3(teachers, courses, teacher_courses))

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

| Файл с тестами - queries test.py:

```
from teacher_course.course import Course
from teacher_course.teacher import Teacher
from teacher_course.queries import query_1, query_2, query_3
from teacher_course.teacher_courses import TeacherCourse

def test_query_1():
    teachers = [Teacher(0, "Иванов Иван Иванович"),
                Teacher(1, "Петрова Анна Ивановна"),
                Teacher(2, "Ильин Илья Ильич"),
                Teacher(3, "Петров Петр Петрович"),
                ]

    courses = [
        Course(0, "Математический анализ", 100.5, 1),
        Course(1, "Физика", 90, 2),
        Course(2, "Объектно-ориентированное программирование", 80, 0),
        Course(3, "Функциональное программирование", 94.5, 3),
        Course(4, "Веб-программирование", 120, 0),
    ]

    expected_result = [
        (Course(id=2, name='Объектно-ориентированное программирование',
hours_num=80, teacher_id=0),
         Teacher(id=0, name='Иванов Иван Иванович')),
        (Course(id=1, name='Физика', hours_num=90, teacher_id=2), Teacher(id=2,
name='Ильин Илья Ильич')),
        (Course(id=3, name='Функциональное программирование', hours_num=94.5,
teacher_id=3),
         Teacher(id=3, name='Петров Петр Петрович')),
        (Course(id=0, name='Математический анализ', hours_num=100.5, teacher_id=1),
         Teacher(id=1, name='Петрова Анна Ивановна')),
        (Course(id=4, name='Веб-программирование', hours_num=120, teacher_id=0),
         Teacher(id=0, name='Иванов Иван Иванович')),
    ]
    assert expected_result == query_1(teachers, courses)

def test_query_2():
    teachers = [Teacher(0, "Иванов Иван Иванович"),
                Teacher(1, "Петрова Анна Ивановна"),
                Teacher(2, "Ильин Илья Ильич"),
                Teacher(3, "Петров Петр Петрович"),
                ]

    courses = [
        Course(0, "Математический анализ", 100.5, 1),
        Course(1, "Физика", 90, 2),
        Course(2, "Объектно-ориентированное программирование", 80, 0),
        Course(3, "Функциональное программирование", 94.5, 3),
        Course(4, "Веб-программирование", 120, 0),
    ]

    expected_result = [
        (Teacher(id=1, name='Петрова Анна Ивановна'), 1),
        (Teacher(id=2, name='Ильин Илья Ильич'), 1),
        (Teacher(id=3, name='Петров Петр Петрович'), 1),
        (Teacher(id=0, name='Иванов Иван Иванович'), 2),
    ]
```

```

assert expected_result == query_2(teachers, courses)

def test_query_3():
    teachers = [Teacher(0, "Иванов Иван Иванович"),
                Teacher(1, "Петрова Анна Ивановна"),
                Teacher(2, "Ильин Илья Ильич"),
                Teacher(3, "Петров Петр Петрович"),
                ]

    courses = [
        Course(0, "Математический анализ", 100.5, 1),
        Course(1, "Физика", 90, 2),
        Course(2, "Объектно-ориентированное программирование", 80, 0),
        Course(3, "Функциональное программирование", 94.5, 3),
        Course(4, "Веб-программирование", 120, 0),
    ]

    teacher_courses = [
        TeacherCourse(0, 2),
        TeacherCourse(0, 4),
        TeacherCourse(1, 0),
        TeacherCourse(2, 1),
        TeacherCourse(3, 3),
    ]

    expected_result = [
        (Course(id=2, name='Объектно-ориентированное программирование',
hours_num=80, teacher_id=0),
        Teacher(id=0, name='Иванов Иван Иванович')),
        (Course(id=3, name='Функциональное программирование', hours_num=94.5,
teacher_id=3),
        Teacher(id=3, name='Петров Петр Петрович')),
        (Course(id=4, name='Веб-программирование', hours_num=120, teacher_id=0),
        Teacher(id=0, name='Иванов Иван Иванович')),
    ]

    assert expected_result == query_3(teachers, courses, teacher_courses)

```

Экранные формы с примерами выполнения программы

Результаты выполнения:

Список преподавателей:

```
Teacher(id=0, name='Иванов Иван Иванович')
Teacher(id=1, name='Петрова Анна Ивановна')
Teacher(id=2, name='Ильин Илья Ильич')
Teacher(id=3, name='Петров Петр Петрович')
```

Список курсов:

```
Course(id=0, name='Математический анализ', hours_num=100.5,
teacher_id=1)
Course(id=1, name='Физика', hours_num=90, teacher_id=2)
Course(id=2, name='Объектно-ориентированное программирование',
hours_num=80, teacher_id=0)
Course(id=3, name='Функциональное программирование', hours_num=94.5,
teacher_id=3)
Course(id=4, name='Веб-программирование', hours_num=120,
teacher_id=0)
```

Список курсов связанных с преподавателем:

```
TeacherCourse(teacher_id=0, course_id=2)
TeacherCourse(teacher_id=0, course_id=4)
TeacherCourse(teacher_id=1, course_id=0)
TeacherCourse(teacher_id=2, course_id=1)
TeacherCourse(teacher_id=3, course_id=3)
```

Задание B1

```
(Course(id=2, name='Объектно-ориентированное программирование',
hours_num=80, teacher_id=0), Teacher(id=0, name='Иванов Иван
Иванович'))
(Course(id=1, name='Физика', hours_num=90, teacher_id=2),
Teacher(id=2, name='Ильин Илья Ильич'))
(Course(id=3, name='Функциональное программирование',
hours_num=94.5, teacher_id=3), Teacher(id=3, name='Петров Петр
Петрович'))
(Course(id=0, name='Математический анализ', hours_num=100.5,
teacher_id=1), Teacher(id=1, name='Петрова Анна Ивановна'))
(Course(id=4, name='Веб-программирование', hours_num=120,
teacher_id=0), Teacher(id=0, name='Иванов Иван Иванович'))
```

Задание B2

```
(Teacher(id=1, name='Петрова Анна Ивановна'), 1)
```

```
(Teacher(id=2, name='Ильин Илья Ильич'), 1)
(Teacher(id=3, name='Петров Петр Петрович'), 1)
(Teacher(id=0, name='Иванов Иван Иванович'), 2)
```

Задание В3

```
(Course(id=2, name='Объектно-ориентированное программирование',
hours_num=80, teacher_id=0), Teacher(id=0, name='Иванов Иван
Иванович'))
```

```
(Course(id=3, name='Функциональное программирование',
hours_num=94.5, teacher_id=3), Teacher(id=3, name='Петров Петр
Петрович'))
```

```
(Course(id=4, name='Веб-программирование', hours_num=120,
teacher_id=0), Teacher(id=0, name='Иванов Иван Иванович'))
```

```
===== test session starts =====
collecting ... collected 3 items

queries_test.py::test_query_1 PASSED [ 33%]
queries_test.py::test_query_2 PASSED [ 66%]
queries_test.py::test_query_3 PASSED [100%]

===== 3 passed in 0.01s =====

Process finished with exit code 0
```