

1. Описание задания

Вариант предметной области:

27	Преподаватель	Учебный курс
----	---------------	--------------

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).

2. Текст Программы

```
import unittest
```

```
class Teacher:
```

```
    def __init__(self, id, fio, salary, course_id):  
        self.id, self.fio, self.salary, self.course_id = id, fio, salary, course_id
```

```
class Course:
```

```
    def __init__(self, id, name):  
        self.id, self.name = id, name
```

```
class TeacherCourse:
```

```
    def __init__(self, course_id, teacher_id):  
        self.course_id, self.teacher_id = course_id, teacher_id
```

```
class UniversityManager:
```

```
    def __init__(self, courses, teachers, relations):  
        self.courses, self.teachers, self.relations = courses, teachers, relations
```

```
    def get_ov_teachers(self):
```

```
        """ИСПРАВЛЕНО: ищет на 'ов' И 'ев""""""
```

```
        d = {c.id: c.name for c in self.courses}
```

```
        return [(t.fio, d[t.course_id])
```

```
            for t in self.teachers
```

```
            if t.fio.endswith('ов') or t.fio.endswith('ев'))]
```

```
    def get_avg_salaries(self):
```

```
        from collections import defaultdict
```

```
        s = defaultdict(list)
```

```
        for t in self.teachers:
```

```
            for c in self.courses:
```

```

        if t.course_id == c.id:
            s[c.name].append(t.salary)
    r = [(c, sum(v)/len(v)) for c, v in s.items()]
    return sorted(r, key=lambda x: x[1])

def get_a_courses(self):
    r = {}
    for c in self.courses:
        if c.name.startswith('A'):
            ts = [t.fio for tc in self.relations
                  for t in self.teachers
                  if tc.course_id == c.id and tc.teacher_id == t.id]
            r[c.name] = ts
    return r

def create_data():
    cs = [Course(1, 'Математический анализ'), Course(2, 'Алгебра и
геометрия'),
          Course(3, 'Программирование на Python'), Course(4, 'Архитектура
компьютера'),
          Course(5, 'Английский язык')]
    ts = [Teacher(1, 'Иванов', 50000, 1), Teacher(2, 'Петров', 60000, 2),
          Teacher(3, 'Сидоров', 55000, 3), Teacher(4, 'Кузнецов', 65000, 3),
          Teacher(5, 'Александров', 45000, 4), Teacher(6, 'Николаев', 70000,
5)]
    rs = [TeacherCourse(1,1), TeacherCourse(2,2), TeacherCourse(3,3),
          TeacherCourse(3,4), TeacherCourse(4,5), TeacherCourse(5,6),
          TeacherCourse(1,3), TeacherCourse(2,4)]
    return cs, ts, rs
# Тест TDD
class TestUniversityManager(unittest.TestCase):
    def test1(self):
        m = UniversityManager(*create_data())
        r = m.get_ov_teachers()
        print(f"\nНайдено преподавателей: {len(r)}")
        print(r)
        self.assertEqual(len(r), 6)

    def test2(self):
        m = UniversityManager(*create_data())
        r = m.get_avg_salaries()
        self.assertEqual(len(r), 5)

```

```

def test3(self):
    m = UniversityManager(*create_data())
    r = m.get_a_courses()
    self.assertIn('Алгебра и геометрия', r)

if __name__ == '__main__':
    print("РЕЗУЛЬТАТЫ:")

    m = UniversityManager(*create_data())
    print("\n1. Преподаватели на 'ов' и 'ев':")
    for fio, course in m.get_ov_teachers():
        print(f" {fio} → {course}")

    print("\n2. Средние зарплаты:")
    for course, salary in m.get_avg_salaries():
        print(f" {course}: {salary:.0f} руб.")

    print("\n3. Курсы на 'А':")
    for course, teachers in m.get_a_courses().items():
        print(f" {course}: {', '.join(teachers)}")

print("ТЕСТЫ:")
unittest.main()

```

3. Результат

(.venv) PS C:\Users\Honor\Desktop\БАУМАНКА 3 КУРС\Пикяп\kai_cs_bot> &
 "C:/Users/Honor/Desktop/БАУМАНКА 3
 КУРС/Пикяп/kai_cs_bot/.venv/Scripts/python.exe"
 "c:/Users/Honor/Desktop/БАУМАНКА 3 КУРС/Пикяп/RK2/import unittest.py"
 РЕЗУЛЬТАТЫ:

1. Преподаватели на 'ов' и 'ев':
 Иванов → Математический анализ
 Петров → Алгебра и геометрия
 Сидоров → Программирование на Python
 Кузнецов → Программирование на Python
 Александров → Архитектура компьютера
 Николаев → Английский язык

2. Средние зарплаты:
 Архитектура компьютера: 45000 руб.
 Математический анализ: 50000 руб.
 Алгебра и геометрия: 60000 руб.

Программирование на Python: 60000 руб.
Английский язык: 70000 руб.

3. Курсы на 'A':

Алгебра и геометрия: Петров, Кузнецов

Архитектура компьютера: Александров

Английский язык: Николаев

ТЕСТЫ:

Найдено преподавателей: 6

[('Иванов', 'Математический анализ'), ('Петров', 'Алгебра и геометрия'), ('Сидоров', 'Программирование на Python'), ('Кузнецов', 'Программирование на Python'), ('Александров', 'Архитектура компьютера'), ('Николаев', 'Английский язык')]

...

Ran 3 tests in 0.001s

OK

(.venv) PS C:\Users\Honor\Desktop\БАУМАНКА 3 КУРС\Пикяп\kai_cs_bot>