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

Курс «Парадигмы и конструкции языков программирования»

Отчет по РК №2

Вариант Д, номер 5

Выполнил: Проверил:

студент группы ИУ5-35Б преподаватель каф. ИУ5

Гонов Марат Гапанюк Ю. Е.

Москва, 2023 г.

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:   
class Musician:

    def \_\_init\_\_(self, id, last\_name, salary, orchestra\_id):

        self.id = id

        self.last\_name = last\_name

        self.salary = salary

        self.orchestra\_id = orchestra\_id

class Orchestra:

    # Класс Orchestra остается без изменений.

    def \_\_init\_\_(self, id, name):

        self.id = id

        self.name = name

class MusiciansOrchestra:

    # Класс MusiciansOrchestra остается без изменений.

    def \_\_init\_\_(self, musician\_id, orchestra\_id):

        self.musician\_id = musician\_id

        self.orchestra\_id = orchestra\_id

# Функция для задачи 1

def filter\_musicians\_by\_last\_name\_and\_orchestra(musicians, orchestras, last\_name\_suffix):

    return [(m.last\_name, o.name)

            for m in musicians

            for o in orchestras

            if m.orchestra\_id == o.id and m.last\_name.endswith(last\_name\_suffix)]

# Функция для задачи 2

def calculate\_average\_salaries(musicians, orchestras):

    avg\_salaries = {}

    for o in orchestras:

        o\_musicians = [m.salary for m in musicians if m.orchestra\_id == o.id]

        avg\_salaries[o.name] = sum(o\_musicians) / len(o\_musicians) if len(o\_musicians) > 0 else 0

    return sorted(avg\_salaries.items(), key=lambda x: x[1], reverse=True)

# Функция для задачи 3

def filter\_orchestras\_and\_list\_musicians(musicians, orchestras, name\_prefix):

    return {o.name: [m.last\_name for m in musicians if m.orchestra\_id == o.id] for o in orchestras if o.name.startswith(name\_prefix)}

# Пример использования данных

musicians = [

    # Ваши данные...

    Musician(1, 'Smith', 30000, 1),

    Musician(2, 'Johnson', 35000, 2),

    Musician(3, 'Williams', 40000, 1),

    Musician(4, 'Jones', 32000, 2),

    Musician(5, 'Brown', 28000, 1)

]

orchestras = [

    # Ваши данные...

    Orchestra(1, 'Symphony Orchestra'),

    Orchestra(2, 'Chamber Orchestra')

]

musicians\_orchestras = [

    # Ваши данные...

    MusiciansOrchestra(1, 1),

    MusiciansOrchestra(2, 2),

    MusiciansOrchestra(3, 1),

    MusiciansOrchestra(4, 2),

    MusiciansOrchestra(5, 1)

]

# Вызов функций

print("Task 1 Result:", filter\_musicians\_by\_last\_name\_and\_orchestra(musicians, orchestras, 'th'))

print("Task 2 Result:", calculate\_average\_salaries(musicians, orchestras))

print("Task 3 Result:", filter\_orchestras\_and\_list\_musicians(musicians, orchestras, 'C'))

import unittest

class TestFilterMusiciansByLastNameAndOrchestra(unittest.TestCase):

    def test\_filter\_by\_suffix(self):

        musicians = [Musician(1, 'Smith', 30000, 1), Musician(2, 'Johnson', 35000, 2)]

        orchestras = [Orchestra(1, 'Symphony Orchestra'), Orchestra(2, 'Chamber Orchestra')]

        result = filter\_musicians\_by\_last\_name\_and\_orchestra(musicians, orchestras, 'th')

        self.assertEqual(result, [('Smith', 'Symphony Orchestra')])

class TestCalculateAverageSalaries(unittest.TestCase):

    def test\_average\_salaries(self):

        musicians = [Musician(1, 'Smith', 30000, 1), Musician(2, 'Johnson', 35000, 1)]

        orchestras = [Orchestra(1, 'Symphony Orchestra')]

        result = calculate\_average\_salaries(musicians, orchestras)

        self.assertEqual(result, [('Symphony Orchestra', 32500.0)])

class TestFilterOrchestrasAndListMusicians(unittest.TestCase):

    def test\_filter\_orchestras(self):

        musicians = [Musician(1, 'Smith', 30000, 1), Musician(2, 'Johnson', 35000, 2)]

        orchestras = [Orchestra(1, 'Symphony Orchestra'), Orchestra(2, 'Chamber Orchestra')]

        result = filter\_orchestras\_and\_list\_musicians(musicians, orchestras, 'C')

        self.assertEqual(result, {'Chamber Orchestra': ['Johnson']})

if \_\_name\_\_ == '\_\_main\_\_':

    unittest.main()

Вывод:  
