

МГТУ им. Н.Э. Баумана

Отчёт по рубежному контролю №2
по курсу «Базовые компоненты и интернет-технологии»
Вариант 17.

Руководитель
Гапанюк Ю.Е.
16.12.2022

Студент группы ИУ5-34Б
Стукалов И.Д.
16.12.2022

2022 г.

Полученное задание:

Необходимо создать два класса данных в соответствии с вариантом
Рубежный контроль представляет собой разработку тестов на языке Python.

1) Провести рефакторинг текста программы рубежного контроля №1 таким образом, чтобы он был пригоден для модульного тестирования.

2) Для текста программы рубежного контроля №1 создать модульные тесты с применением TDD - фреймворка (3 теста).

Текст программы:

// файл DirOrch.py

```
from operator import itemgetter

class Director:
    """Дирижер"""

    def __init__(self, id, name, salary, orch_id):
        self.id = id
        self.name = name
        self.salary = salary
        self.orch_id = orch_id

class Orchestra:
    """Оркестр"""

    def __init__(self, id, title):
        self.id = id
        self.title = title

class DirectorOrchestra:
    """
    'Дирижеры оркестров' для реализации
    СВЯЗИ МНОГИЕ-КО-МНОГИМ
    """

    def __init__(self, dir_id, orch_id):
        self.dir_id = dir_id
        self.orch_id = orch_id

# Дирижеры
directors = [
    Director(1, 'Бан', 150000, 1),
    Director(2, 'Вагнер', 2800, 2),
    Director(3, 'Деймур', 100000, 2),
    Director(4, 'Барииниис', 86400, 3),
    Director(5, 'Поляков', 44444, 4)
]

# Оркестры
orchestras = [
    Orchestra(1, 'Банановая республика'),
    Orchestra(2, 'AGSPD'),
```

```

    Orchestra(3, 'BTS'),
    Orchestra(4, 'Дрилл Авиация')
]

directors_orchestras = [
    DirectorOrchestra(1, 1),
    DirectorOrchestra(2, 2),
    DirectorOrchestra(3, 3),
    DirectorOrchestra(4, 4),
    DirectorOrchestra(5, 1),

    DirectorOrchestra(5, 1),
    DirectorOrchestra(2, 3),
    DirectorOrchestra(3, 2),
    DirectorOrchestra(4, 1),
    DirectorOrchestra(5, 4)
]

# Соединение данных один-ко-многим
one_to_many = [(dir.name, dir.salary, orch.title)
                for orch in orchestras
                for dir in directors
                if dir.orch_id == orch.id]

# Соединение данных многие-ко-многим
many_to_many_temp = [(orch.title, dir.orch_id, dir.dir_id)
                      for orch in orchestras
                      for dir in directors_orchestras
                      if orch.id == dir.orch_id]

many_to_many = [(dir.name, dir.salary, orchestra_name)
                 for orchestra_name, orch_id, dir_id in many_to_many_temp
                 for dir in directors if dir.id == dir_id]

```

//файл test.py

```

from operator import itemgetter
import unittest
import DirOrch

class TestingRK(unittest.TestCase):

    def test_task_1(self):
        task_1 = {}
        for orch in DirOrch.orchestras:
            if orch.title[0] == 'A':
                d_emps = list(filter(lambda i: i[2] == orch.title,
DirOrch.one_to_many))
                d_emps_names = [x for x, _, _ in d_emps]
                task_1[orch.title] = d_emps_names
        self.assertEqual(task_1, {'AGSPD': ['Ватнер', 'Деймур']})

    def test_task_2(self):
        task_2_unsorted = []
        for orch in DirOrch.orchestras:
            orch_Directors = list(filter(lambda i: i[2] == orch.title,
DirOrch.one_to_many))
            if len(orch_Directors) > 0:
                orch_sals = [sal for _, sal, _ in orch_Directors]
                orch_sals_sum = max(orch_sals)
                task_2_unsorted.append((orch.title, orch_sals_sum))
        task_2 = sorted(task_2_unsorted, key=itemgetter(1), reverse=True)

```

```

self.assertEqual(task_2, [('Банановая республика', 150000),
                           ('AGSPD', 100000),
                           ('BTS', 86400),
                           ('Дрилл Авиация', 44444)])

def test_task_3(self):
    task_3 = sorted(DirOrch.many_to_many, key=itemgetter(2))
    self.assertEqual(task_3, [('Ватнер', 2800, 'AGSPD'),
                              ('Деймур', 100000, 'AGSPD'),
                              ('Деймур', 100000, 'BTS'),
                              ('Ватнер', 2800, 'BTS'),
                              ('Бан', 150000, 'Банановая республика'),
                              ('Поляков', 44444, 'Банановая республика'),
                              ('Поляков', 44444, 'Банановая республика'),
                              ('Барииниис', 86400, 'Банановая
республика'),
                              ('Барииниис', 86400, 'Дрилл Авиация'),
                              ('Поляков', 44444, 'Дрилл Авиация')])

```

Результаты выполнения:

```

"C:\work\Бомонка\3 сем\ВКИД\ПК\venv\Scripts\python.exe" "C:/Program Files/JetBrains/PyCharm Community Edition 2022.2.1/plugins/python-ce/help
Testing started at 14:15 ...
Launching unittests with arguments python -m unittest test.TestingRK.test_task_2 in C:\work\Бомонка\3 сем\ВКИД\ПК

Ran 1 test in 0.003s

OK

Process finished with exit code 0

```