Московский государственный технический университет им. Н.Э. Баумана

Факультет ИУ
Кафедра ИУ5

Курс «Основы информатики»
Отчет по Рубежному контролю №2

Выполнил студент группы ИУ5-33Б:
Козлов А. А.
Подпись и дата:

Проверил преподаватель каф.:
Гапанюк Ю. Е.
Подпись и дата:

Москва, 2024 г

Постановка задачи

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).

Текст программы

```
import unittest
from operator import itemgetter

class Computer():
    def __init__(self, id, name, cost, class_id):
        self.id = id
        self.name = name
        self.cost = cost
        self.class_id = class_id

class DisClass():
    def __init__(self, id, name):
        self.id = id
        self.name = name

class CompDC():
    def __init__(self, class_id, comp_id):
        self.class_id = class_id
        self.comp_id = comp_id

comps = [
    Computer(1, 'A123', 100, 1),
    Computer(2, 'B123', 200, 2),
    Computer(3, 'A233', 150, 3),
    Computer(4, 'D123', 200, 4),
    Computer(5, 'E123', 300, 5),]

dis_classes = [
    DisClass(1, '312л'),
    DisClass(2, '313л'),
    DisClass(3, '314л'),
    DisClass(4, '315л'),
    DisClass(5, '316л'),]

comp_dc = [
    CompDC(1, 1),
    CompDC(2, 2),
    CompDC(3, 3),
    CompDC(3, 4),
    CompDC(3, 5),
    CompDC(4, 1),
```

```
    CompDC(4, 2),
    CompDC(4, 3),
    CompDC(5, 4),
    CompDC(5, 5),]


one_to_many = [(c.name, c.cost, dc.name)
               for dc in dis_classes
               for c in comps
               if c.class_id == dc.id]

many_to_many_temp = [(dc.name, co.class_id, co.comp_id)
               for dc in dis_classes
               for co in comp_dc
               if dc.id == co.class_id]

many_to_many = [(c.name, c.cost, class_name)
               for class_name, class_id, comp_id in many_to_many_temp
               for c in comps if c.id == comp_id]



'''
print('Задание B1')
«дисплейный класс» и «компьютер» связаны соотношением один-ко-многим. Выведите список всех
компьютеров, у которых название начинается с буквы «А», и названия их дисплейных классов.
'''
def firstEx():
    resultB1 = sorted(x for x in one_to_many if x[0].startswith('A'))
    return[f'Название: {i[0]}, класс: {i[2]}' for i in resultB1]



'''
print('\nЗадание B2')
«дисплейный класс» и «компьютер» связаны соотношением один-ко-многим. Выведите список
классов с минимальной ценой за компьютер в каждом дисплейном классе, отсортированный по
минимальной цене.
'''
def secondEx():
    resultB2 = {}
    for class_name in set(x[2] for x in one_to_many):
        min_cost = min(x[1] for x in one_to_many if x[2] == class_name)
        resultB2[class_name] =  min_cost
    resultB2 = sorted(resultB2.items(), key=itemgetter(1))
    return[f'Класс: {i[0]}, минимальная стоимость: {i[1]}' for i in resultB2]



'''
print('\nЗадание B3')
«дисплейный класс» и «компьютер» связаны соотношением многие-ко-многим. Выведите список
всех связанных компьютеров и дисплейных классов, отсортированный по компьютерам,
сортировка по дисплейным классам произвольная.
'''
```

```python
def thirdEx():
    resultB3 = sorted(many_to_many, key=itemgetter(0))
    return[f'Название: {i[0]}, класс: {i[2]}' for i in resultB3]



class TestMyFunctions(unittest.TestCase):
    def test_firstEx(self):
        expected_result = ['Название: A123, класс: 312л', 'Название: A233, класс: 314л']
        self.assertEqual(firstEx(), expected_result)

    def test_secondEx(self):
        expected_result = ['Класс: 312л, минимальная стоимость: 100',
            'Класс: 314л, минимальная стоимость: 150',
            'Класс: 315л, минимальная стоимость: 200',
            'Класс: 313л, минимальная стоимость: 200',
            'Класс: 316л, минимальная стоимость: 300']
        self.assertEqual(secondEx(), expected_result)

    def test_thirdEx(self):
        expected_result = ['Название: A123, класс: 312л',
            'Название: A123, класс: 315л',
            'Название: A233, класс: 314л',
            'Название: A233, класс: 315л',
            'Название: B123, класс: 313л',
            'Название: B123, класс: 315л',
            'Название: D123, класс: 314л',
            'Название: D123, класс: 316л',
            'Название: E123, класс: 314л',
            'Название: E123, класс: 316л']
        self.assertEqual(thirdEx(), expected_result)

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

Анализ результатов

```
Data\Local\Programs\Python\Python313\python.exe' 'c:\U
bs\debugpy\adapter/../..\debugpy\launcher' '50422' '--
...
----------------------------------------------------------------------
Ran 3 tests in 0.000s

OK
PS C:\Users\123\Documents\Work\Programs\2ndSem-RK2>
```