Московский государственный технический университет имени Н.Э.Баумана

Факультет Радиотехнический

Кафедра ИУ5 Системы обработки информации и управления

Отчет по рубежному контролю № 2 по курсу Базовые компоненты

Исполнитель
Студент группы РТ5-31Б _____ Кекин И.А.
«__»_____ 2022 г.

Проверил
Доцент кафедры ИУ5 _____ Гапанюк Ю.Е.
«__»_____ 2022 г.

г. Москва – 2022

Задание РК2

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).

Задание РК1

Предметная область E, вариант 15. Классы: Каталог, Файл.

Задания:

1. «Каталог» и «Файл» связаны соотношением один-ко-многим. Выведите список всех каталогов, у которых в названии присутствует буква «Г», и список его файлов.
2. «Каталог» и «Файл» связаны соотношением один-ко-многим. Выведите список каталогов со средним кол-вом файлов в них, отсортированный по среднему кол-ву файлов. Среднее кол-во файлов в каталоге должно быть округлено до 2 знаков после запятой.
3. «Каталог» и «Файл» связаны соотношением многие-ко-многим. Выведите список всех файлов, у которых название начинается с буквы «g», и названия их каталогов.

Листинг программы, в которой выполняются задания и для которой был проведён рефакторинг (RK1.py)

```
'''Вариант - E, вариант предметной области - 15
("Файл - Каталог")'''
from operator import itemgetter


class file:
    """файл"""
    def __init__(self, id, name, size, catId):
        self.id = id
        self.name = name
        self.size = size
        self.catId = catId


class cat:
```

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

class fileCat:
    """
    'файлы каталога' для реализации
    связи многие-ко-многим
    """
    def __init__(self, catId, fileId):
        self.catId = catId
        self.fileId = fileId

cats = [cat(1, "D:\программы на python"),
        cat(2, "D:\отчёты по лабам"),
        cat(3, "C:\игры")]

files = [file(1,"Hello world.py",1,1),
         file(2,"лаб 2 отчёт",100,2),
         file(3,"game.py",3000,1)]

filesCats = [fileCat(1,1),
             fileCat(2,2),
             fileCat(1,3),
             fileCat(3,3)]

catsId = [c.id for c in cats]
oneToMany = [(f.name, f.size, cats[catsId.index(f.catId)].name) for f in files]

filesId = [f.id for f in files]
manyToMany = [(files[filesId.index(fc.fileId)].name,
```

```
            files[filesId.index(fc.fileId)].size,
            cats[catsId.index(fc.catId)].name)
            for fc in filesCats]


def task1(oneToMany):
    word1 = "г"
    catsE1 = [c.name for c in cats if word1 in c.name]
    filesE1 = [otm[0] for otm in oneToMany for c in catsE1 if otm[2] == c]
    return catsE1, filesE1


def task2(oneToMany):
    return sorted([[c.name, round(sum([otm[1] for otm in oneToMany if otm[2] ==
c.name])/(lambda x: 1 if x==0 else x)(len([otm[1] for otm in oneToMany if otm[2] ==
c.name])))] for c in cats], key=itemgetter(1),reverse=True)


def task3(manyToMany):
    char3 = "g"
    return [[f.name,[mtm[2] for mtm in manyToMany if mtm[0]==f.name]] for f in files if
f.name[0] == char3]


if __name__ == '__main__':
    print(task1(oneToMany))
    print(task2(oneToMany))
    print(task3(manyToMany))
```

Листинг программы, в которой проводятся тесты (RK2 Kekin RT5-31B.py)

```
import unittest
import RK1


class testRK1(unittest.TestCase):
    def setUp(self):
        self.test1 = (['D:\\программы на python', 'C:\\игры'], ['Hello world.py', 'game.py'])
        self.test2 = [['D:\\программы на python', 1500], ['D:\\отчёты по лабам', 100], ['C:\\игры',
0]]
```

```
        self.test3 = [['game.py', ['D:\\программы на python', 'C:\\игры']]]

    def test1_rk(self):
        self.assertEqual(RK1.task1(RK1.oneToMany), self.test1)

    def test2_rk(self):
        self.assertEqual(RK1.task2(RK1.oneToMany), self.test2)

    def test3_rk(self):
        self.assertEqual(RK1.task3(RK1.manyToMany), self.test3)

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

Результаты работы программы

```
...
----------------------------------------------------------------------
Ran 3 tests in 0.001s
```