

**Московский государственный технический университет
им. Н.Э. Баумана**

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

Курс «Парадигмы и конструкции языков программирования»

**Отчет по РК№2
Вариант запросов: Г
Вариант предметной области: 14**

Выполнил:
студент группы ИУ5-34Б
Сергеев Максим
Подпись и дата:

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю.Е.
Подпись и дата:

Москва, 2023 г.

Задача:

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:

```
# Файл main.py
class Disk:
    def __init__(self, id, name, publisher, release_date, lib_id):
        self.id = id,
        self.name = name,
        self.publisher = publisher,
        self.release_date = release_date
        self.lib_id = lib_id

    def __str__(self):
        return f"{self.name[0]} by {self.publisher[0]}({self.release_date})"

class Library:
    def __init__(self, id, owner):
        self.id = id,
        self.owner = owner

    def __str__(self):
        return f"{self.owner}"

class DiskLibrary:
    def __init__(self, disk_id, lib_id):
        self.disk_id = disk_id
        self.lib_id = lib_id

    def __str__(self):
        return f"{self.disk_id[0]} - {self.lib_id[0]}"

disks = [Disk(1, 'Alphaville - Jerusalem', 'WEA', 1986, 1),
         Disk(2, 'Orden Ogan - Final Days', 'AFM', 2021, 1),
         Disk(3, 'George Orwell - 1984', 'Verlag', 2007, 3),
         Disk(4, 'Read dead redemption 2', 'Rockstar', 2018, 2),
         Disk(5, 'God of War: Ragnarok', 'Sony', 2022, 2)
        ]

libs = [Library(1, 'Max Sergeev'),
        Library(2, 'Name Nickname'),
        Library(3, 'MGTU')
        ]

disk_lib = [DiskLibrary(1, 1),
            DiskLibrary(2, 1),
            DiskLibrary(1, 2),
            DiskLibrary(4, 2),
            DiskLibrary(3, 2),
            DiskLibrary(3, 3),
            DiskLibrary(2, 2),
            DiskLibrary(5, 2)
            ]

def main():
    otm = [(i, j) for i in disks for j in libs if i.lib_id == j.id[0]]
    otm.sort(key=lambda x: x[1].id)
    mtm_tmp = [(i.owner, j.disk_id) for i in libs for j in disk_lib if
i.id[0] == j.lib_id]
    mtm = [(i[0], str(j)) for i in mtm_tmp for j in disks if i[1] == j.id[0]]
    print('-----Task 1:-----')
    print('.....Their CDs:.....')
```

```

ans = []
for i in otm:
    if i[1].owner[0] == 'M':
        ans.append(i[1].owner)
        print(i[1], 'owns', i[0])
print('.....')
ans1 = sorted(list(set(ans)))
print('Owners name start with m:', ans1)

print('-----Task 2:-----')
dic = {}
for i in otm:
    if i[1].owner in dic:
        dic[i[1].owner] = min(dic[i[1].owner], i[0].release_date)
    else:
        dic[i[1].owner] = i[0].release_date
sorted_dic = sorted(dic.items(), key=lambda x: x[1])
print(sorted_dic)

print('-----Task 3:-----')
ans3 = sorted(mtm, key=lambda x: x[0])
print(ans3)

if __name__ == '__main__':
    main()

```

```

# Файл tests.py

import unittest
from main import *

class Test(unittest.TestCase):
    disks = [Disk(1, 'Alphaville - Jerusalem', 'WEA', 1986, 1),
             Disk(2, 'Orden Ogan - Final Days', 'AFM', 2021, 1),
             Disk(3, 'George Orwell - 1984', 'Verlag', 2007, 3),
             Disk(4, 'Read dead redemption 2', 'Rockstar', 2018, 2),
             Disk(5, 'God of War: Ragnarok', 'Sony', 2022, 2)
            ]

    libs = [Library(1, 'Max Sergeev'),
            Library(2, 'Name Nickname'),
            Library(3, 'MGТУ')
            ]

    disk_lib = [DiskLibrary(1, 1),
                DiskLibrary(2, 1),
                DiskLibrary(1, 2),
                DiskLibrary(4, 2),
                DiskLibrary(3, 2),
                DiskLibrary(3, 3),
                DiskLibrary(2, 2),
                DiskLibrary(5, 2)
                ]

    def test_1(self):
        otm = [(i, j) for i in disks for j in libs if i.lib_id == j.id[0]]
        otm.sort(key=lambda x: x[1].id)
        ans = []
        for i in otm:
            if i[1].owner[0] == 'M':
                ans.append(i[1].owner)

```

```

ans1 = sorted(list(set(ans)))
self.assertEqual(ans1, ['MGTU', 'Max Sergeev'])

def test_2(self):
    otm = [(i, j) for i in disks for j in libs if i.lib_id == j.id[0]]
    otm.sort(key=lambda x: x[1].id)
    dic = {}
    for i in otm:
        if i[1].owner in dic:
            dic[i[1].owner] = min(dic[i[1].owner], i[0].release_date)
        else:
            dic[i[1].owner] = i[0].release_date
    sorted_dic = sorted(dic.items(), key=lambda x: x[1])
    self.assertEqual(sorted_dic, [('Max Sergeev', 1986), ('MGTU', 2007),
('Name Nickname', 2018)])

def test_3(self):
    mtm_tmp = [(i.owner, j.disk_id) for i in libs for j in disk_lib if
i.id[0] == j.lib_id]
    mtm = [(i[0], str(j)) for i in mtm_tmp for j in disks if i[1] ==
j.id[0]]
    ans3 = sorted(mtm, key=lambda x: x[0])
    self.assertEqual(ans3, [('MGTU', 'George Orwell - 1984 by
Verlag(2007)'), ('Max Sergeev', 'Alphaville - Jerusalem by WEA(1986)'), ('Max
Sergeev', 'Orden Ogan - Final Days by AFM(2021)'), ('Name Nickname',
'Alphaville - Jerusalem by WEA(1986)'), ('Name Nickname', 'Read dead
redemption 2 by Rockstar(2018)'), ('Name Nickname', 'George Orwell - 1984 by
Verlag(2007)'), ('Name Nickname', 'Orden Ogan - Final Days by AFM(2021)'),
('Name Nickname', 'God of War: Ragnarok by Sony(2022)')])

```

Результат работы программы:

```

D:\GitRepos\Proga3sem\RK2\venv\Scripts\python.exe "D:\PyCharm Community Edition 2021.2.1\plugins\pytho
Testing started at 15:12 ...

Ran 3 tests in 0.003s

Launching unittests with arguments python -m unittest D:/GitRepos/Proga3sem/RK2/tests.py in D:\GitRepo
OK

```