

**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:
Сикоринский А. К.
Подпись и дата:

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

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

class CD_Disk:
    def __init__(self, id, name, size, library_id):
        self.id = id
        self.name = name
        self.size = size
        self.library_id = library_id

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

class Library_to_Disk:
    def __init__(self, id, disk_id ,library_id):
        self.id = id
        self.disk_id = disk_id
        self.library_id = library_id

cd_disks = [
    CD_Disk(1, "Классика", 432, 1),
    CD_Disk(2, "Рок-концерты 2022", 420, 2),
    CD_Disk(3, "Назад в Черное", 400, 3),
    CD_Disk(4, "Аби Роуд", 425, 1),
    CD_Disk(5, "Рожденный бегать", 380, 4),
    CD_Disk(6, "Слухи", 410, 5),
    CD_Disk(7, "Led Zeppelin IV", 450, 1),
    CD_Disk(8, "Рок 2004", 470, 2),
    CD_Disk(9, "Клуб одиноких сердец сержанта Пеппера", 415,
1),
    CD_Disk(10, "Нирвана", 405, 6),
]

cd_library_list = [
    CD_Library(1, "Классическая музыка"),
    CD_Library(2, "Рок-музыка"),
    CD_Library(3, "Поп-музыка"),
    CD_Library(4, "Джаз"),
```

```

        CD_Library(5, "Электронная музыка"),
        CD_Library(6, "Фолк-музыка"),
    ]
    library_to_disk_objects = [
        Library_to_Disk(1, 1, 1),
        Library_to_Disk(2, 2, 2),
        Library_to_Disk(3, 3, 3),
        Library_to_Disk(4, 4, 4),
        Library_to_Disk(5, 5, 5),
        Library_to_Disk(6, 6, 6),
        Library_to_Disk(7, 7, 1),
        Library_to_Disk(8, 8, 2),
        Library_to_Disk(9, 9, 3),
        Library_to_Disk(10, 10, 4),
        Library_to_Disk(11, 1, 5),
        Library_to_Disk(12, 2, 6),
        Library_to_Disk(13, 3, 1),
        Library_to_Disk(14, 4, 2),
        Library_to_Disk(15, 5, 3),
        Library_to_Disk(16, 6, 4),
        Library_to_Disk(17, 7, 5),
        Library_to_Disk(18, 8, 6),
        Library_to_Disk(19, 9, 1),
        Library_to_Disk(20, 10, 2),
    ]

#B1
#«Библиотека» и «CD-Диск» связаны соотношением один-ко-многим.
#Выведите список всех CD-дисков, у которых название начинается
с "P",
#и названия их библиотеки.
def ex1():
    result = []
    for i in cd_disks:
        for g in cd_library_list:
            if i.library_id == g.id:
                if i.name[0] == "P":
                    result.append(f"Библиотека: {g.name}; CD-
диск: {i.name}")
    return result

#B2
#«Библиотека» и «CD-Диск» связаны соотношением один-ко-многим.
#Выведите список библиотек с минимальной емкостью CD-диска в
каждом библиотеке,
#отсортированный по минимальной емкости.

def ex2():
    dict = {}

```

```

for i in cd_disks:
    for g in cd_library_list:
        if i.library_id == g.id:
            if g.name not in dict:
                dict[g.name] = i.size
            else:
                dict[g.name] = min(dict[g.name], i.size)
    return [f"Библиотека: {h}; минимальная емкость диска
{dict[h]} МБ" for h in dict]

```

#B3

#«Библиотека» и «CD-Диск» связаны соотношением многие-ко-многим.

#Выведите список всех связанных CD-дисков и библиотек, отсортированный по жестким дискам,

#сортировка по Библиотекаам произвольная.

```

def ex3():
    def lm(obj):
        for i in cd_disks:
            if obj.disk_id == i.id:
                return i.name
    def lib(obj):
        for i in cd_library_list:
            if obj.library_id == i.id:
                return i.name

    sortes = list(map(lambda x: (lm(x), lib(x)),
sorted(library_to_disk_objects, key = lm)))
    return [f"Библиотека: {i[1]}; CD-диск: {i[0]}" for i in
sortes]

```

```

class MyTests(unittest.TestCase):

```

```

    def test_ex1(self):
        expected_result = ['Библиотека: Рок-музыка; CD-диск:
Рок-концерты 2022', 'Библиотека: Джаз; CD-диск: Рожденный
бегать', 'Библиотека: Рок-музыка; CD-диск: Рок 2004']
        self.assertEqual(ex1(), expected_result)
    def test_ex2(self):
        expected_result = ['Библиотека: Классическая музыка;
минимальная емкость диска 415 МБ', 'Библиотека: Рок-музыка;
минимальная емкость диска 420 МБ', 'Библиотека: Поп-музыка;
минимальная емкость диска 400 МБ', 'Библиотека: Джаз;
минимальная емкость диска 380 МБ', 'Библиотека: Электронная
музыка; минимальная емкость диска 410 МБ', 'Библиотека: Фолк-
музыка; минимальная емкость диска 405 МБ']
        self.assertEqual(ex2(), expected_result)
    def test_ex3(self):
        expected_result = ['Библиотека: Классическая музыка;
CD-диск: Led Zeppelin IV', 'Библиотека: Электронная музыка;
CD-диск: Led Zeppelin IV',
        'Библиотека: Джаз; CD-диск: Аби Роуд', 'Библиотека:

```

```

Рок-музыка; CD-диск: Аби Роуд', 'Библиотека: Классическая
музыка; CD-диск: Классика',
    'Библиотека: Электронная музыка; CD-диск: Классика',
    'Библиотека: Поп-музыка; CD-диск: Клуб одиноких сердец
сержанта Пеппера',
    'Библиотека: Классическая музыка; CD-диск: Клуб
одиноких сердец сержанта Пеппера', 'Библиотека: Поп-музыка;
CD-диск: Назад в Черное',
    'Библиотека: Классическая музыка; CD-диск: Назад в
Черное', 'Библиотека: Джаз; CD-диск: Нирвана', 'Библиотека:
Рок-музыка; CD-диск: Нирвана',
    'Библиотека: Электронная музыка; CD-диск: Рожденный
бегать', 'Библиотека: Поп-музыка; CD-диск: Рожденный бегать',
    'Библиотека: Рок-музыка; CD-диск: Рок 2004',
    'Библиотека: Фолк-музыка; CD-диск: Рок 2004',
    'Библиотека: Рок-музыка; CD-диск: Рок-концерты 2022',
    'Библиотека: Фолк-музыка; CD-диск: Рок-концерты 2022',
    'Библиотека: Фолк-музыка; CD-диск: Слухи',
    'Библиотека: Джаз; CD-диск: Слухи']
    self.assertEqual(ex3(), expected_result)

if __name__ == "__main__":
    unittest.main()

```

Анализ результатов

```
PS D:\Downloads\atom> py course3.py
```

```
...
```

```
-----
```

```
Ran 3 tests in 0.000s
```

```
OK
```