



**Министерство науки и высшего образования Российской  
Федерации Федеральное государственное бюджетное  
образовательное учреждение высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)**

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

**Курс «Базовые компоненты интернет-технологий»  
Отчет по рубежному контролю №2**

**Выполнила:  
студентка группы ИУ5-33Б  
Пономарева М.О.**

**Проверил:  
Гапанюк Ю.Е.**

### Описание задачи:

- 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

```
from operator import itemgetter

class CD:
    """CD-диск"""
    def __init__(self, id, title, sal, lib_id):
        self.id = id
        self.title = title
        self.sal = sal
        self.lib_id = lib_id

class Lib:
    """CD-диск"""

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

class CDLib:
    """
    ' для реализации
    СВЯЗИ МНОГИЕ-КО-МНОГИМ
    """

    def __init__(self, lib_id, cd_id):
        self.lib_id = lib_id
        self.cd_id = cd_id

# библиотеки
libs = [
    Lib(1, 'NewCD'),
    Lib(2, 'CDAGE'),
    Lib(3, 'Discs'),
    Lib(4, 'CDshop'),
    Lib(5, 'Archive'),
]

# Диски
cds = [
    CD(1, 'All music', 250, 3),
    CD(2, 'Orchestra music', 220, 2),
    CD(3, 'Rock music', 450, 5),
    CD(4, 'Amy Winehouse Pop hits', 350, 2),
    CD(5, 'Jazz for all', 550, 3),
]

cds_libs = [
    CDLib(1, 3),
    CDLib(2, 2),
    CDLib(3, 5),
```

```

CDLib(4, 2),
CDLib(5, 3),

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

def main():
    """Основная функция"""
    # Соединение данных один-ко-многим
def B1 (one_to_many):
    res11 = []
    for title, _, libs in one_to_many:
        if title[0] == "A":
            res11.append((title, libs))
    return res11

def B2 (one_to_many):
    res12 = [[one_to_many[0][2], one_to_many[0][1]]]
    for title, sal, libs in one_to_many:
        if libs == res12[len(res12) - 1][0]:
            if sal < res12[len(res12) - 1][1]:
                res12[len(res12) - 1][1] = sal
            else:
                res12.append([libs, sal])
    res12 = sorted(res12, key=itemgetter(1))
    return res12
def B3 (many_to_many):
    res13 = []
    for title, _, libs in many_to_many:
        res13.append((title, libs))
    res13 = sorted(res13, key=itemgetter(0))
    return res13

def main():

    one_to_many = [(c.title, c.sal, l.name)
                    for l in libs
                    for c in cds
                    if c.lib_id == l.id]

    # Соединение данных многие-ко-многим
    many_to_many_temp = [(l.name, cl.lib_id, cl.cd_id)
                          for l in libs
                          for cl in cds_libs
                          if l.id == cl.lib_id]

    many_to_many = [(c.title, c.sal, lib_name)
                    for lib_name, lib_id, cd_id in many_to_many_temp
                    for c in cds if c.id == cd_id]

    print('Задание B1')
    print(B1(one_to_many))

    print('\n Задание B2')
    print(B2(one_to_many))

    print('\n Задание B3')
    print(B3(many_to_many))

if __name__ == '__main__':
    main()

```

## Файл tdd.py

```
import unittest
import sys, os

from main import *

class TestMain(unittest.TestCase):
    def test_1(self):
        one_to_many = [(c.title, c.sal, l.name)
                        for l in libs
                        for c in cds
                        if c.lib_id == l.id]
        self.assertEqual(B1(one_to_many), [('Amy Winehouse Pop hits',
'CDAGE'), ('All music', 'Discs')])

    def test_2(self):
        one_to_many = [(c.title, c.sal, l.name)
                        for l in libs
                        for c in cds
                        if c.lib_id == l.id]
        self.assertEqual(B2(one_to_many), [['CDAGE', 220], ['Discs', 250],
['Archive', 450]])

    def test_3(self):
        many_to_many_temp = [(l.name, cl.lib_id, cl.cd_id)
                              for l in libs
                              for cl in cds_libs
                              if l.id == cl.lib_id]

        many_to_many = [(c.title, c.sal, lib_name)
                          for lib_name, lib_id, cd_id in many_to_many_temp
                          for c in cds if c.id == cd_id]
        self.assertEqual(B3(many_to_many), [('All music', 'Discs'), ('All
music', 'Archive'), ('Amy Winehouse Pop hits', 'CDAGE'), ('Amy Winehouse Pop
hits', 'CDshop'), ('Jazz for all', 'Discs'), ('Orchestra music', 'NewCD'),
('Orchestra music', 'CDAGE'), ('Orchestra music', 'CDshop'), ('Rock music',
'NewCD'), ('Rock music', 'Archive')])
```

## Результаты выполнения программы:

```
C:\Users\mm198\PycharmProjects\pythonProject1\env\Scripts\python.exe "C:\Program Files\JetBrains\PyCharm Community Edition 2021.2.3\plugins\pyt
Testing started at 16:30 ...
Launching unittests with arguments python -m unittest tdd.TestMain in C:\Users\mm198\PycharmProjects\pythonProject1

Ran 3 tests in 0.011s

OK

Process finished with exit code 0
```