Рубежный контроль № 2

Максименко София ИБМ3-34Б

14 вариант

Класс 1: «CD-диск»

Класс 2: «Библиотека CD-дисков»

Рефакторинг кода

1. Исправление ошибок в методах и добавление конструкторов.
2. Разделение логики на отдельные функции, чтобы можно было тестировать каждую из них.

Рефакторинг кода:

class CdDisk:

    def \_\_init\_\_(self, id, name, number, cd\_disk\_id):

        self.id = id

        self.name = name

        self.number = number

        self.cd\_disk\_id = cd\_disk\_id

class LibraryCd:

    def \_\_init\_\_(self, id, name):

        self.id = id

        self.name = name

class CDDickLibrary:

    def \_\_init\_\_(self, library\_id, cd\_disk\_id):

        self.cd\_disk\_id = cd\_disk\_id

        self.library\_id = library\_id

# Библиотеки

libraries = [

    LibraryCd(1, 'Первая'),

    LibraryCd(2, 'Вторая'),

    LibraryCd(3, 'Третья'),

]

# cd-диски

cd\_disks = [

    CdDisk(1, 'ROM', 25000, 1),

    CdDisk(2, 'R - READ', 35000, 2),

    CdDisk(3, 'RW - Rewritable', 45000, 3),

]

# связи дисков и библиотек

cd\_disk\_libraries = [

    CDDickLibrary(1, 1),

    CDDickLibrary(2, 2),

    CDDickLibrary(1, 3),

]

def get\_one\_to\_many():

    return sorted([(i.name, i.number, k.name) for k in libraries for i in cd\_disks if i.cd\_disk\_id == k.id])

def get\_total\_disks\_per\_library():

    total\_count = {}

    for disk in cd\_disks:

        total\_count[disk.cd\_disk\_id] = total\_count.get(disk.cd\_disk\_id, 0) + disk.number

    return [(k.name, total\_count.get(k.id, 0)) for k in libraries]

def get\_cd\_disks\_with\_libraries():

    many\_to\_many = {}

    for library in libraries:

        many\_to\_many[library.name] = []

        for relation in cd\_disk\_libraries:

            if library.id == relation.library\_id:

                cd\_disk\_name = next(disk.name for disk in cd\_disks if disk.id == relation.cd\_disk\_id)

                many\_to\_many[library.name].append(cd\_disk\_name)

    return many\_to\_many

def main():

    print("Задание A1")

    print(get\_one\_to\_many())

    print("Задание A2")

    print(get\_total\_disks\_per\_library())

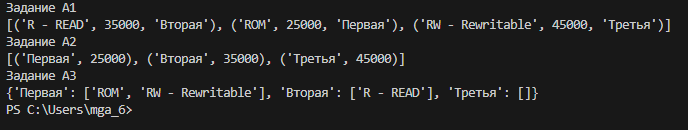
    print("Задание A3")

    print(get\_cd\_disks\_with\_libraries())

if \_\_name\_\_ == "\_\_main\_\_":

    main()

Вывод:



1. Создание модульных тестов для этих функций. В процессе будет использован unittest, стандартный фреймворк для тестирования в Python.

import unittest

class TestLibraryFunctions(unittest.TestCase):

    def test\_get\_one\_to\_many(self):

        expected\_result = [

            ('ROM', 25000, 'Первая'),

            ('RW - Rewritable', 45000, 'Первая'),

            ('R - READ', 35000, 'Вторая')

        ]

        self.assertEqual(get\_one\_to\_many(), expected\_result)

    def test\_get\_total\_disks\_per\_library(self):

        expected\_result = [

            ('Первая', 70000),

            ('Вторая', 35000),

            ('Третья', 0)

        ]

        self.assertEqual(get\_total\_disks\_per\_library(), expected\_result)

    def test\_get\_cd\_disks\_with\_libraries(self):

        expected\_result = {

            'Первая': ['ROM', 'RW - Rewritable'],

            'Вторая': ['R - READ'],

            'Третья': []

        }

        self.assertEqual(get\_cd\_disks\_with\_libraries(), expected\_result)

if \_\_name\_\_ == "\_\_main\_\_":

    unittest.main()

Вывод:

