Факультет «Радиотехнический»
Кафедра «Системы обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования»

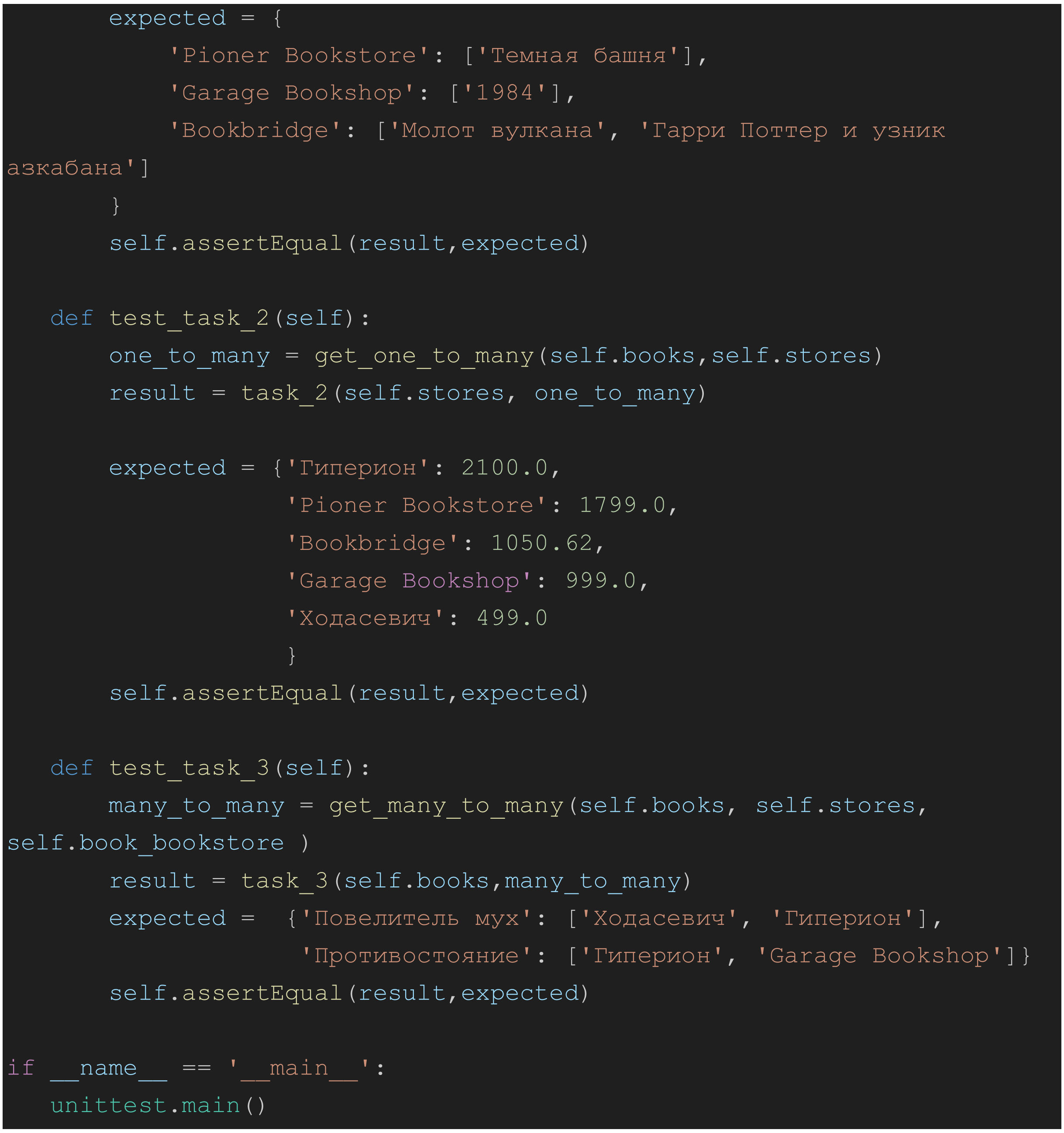
Отчет по Рубежному контролю № 2

Выполнил:	Проверил:
студент группы РТ5-31Б:	преподаватель кафедры ИУ5
Паншин М.В.	Гапанюк Ю.Е.

Москва, 2024 г.

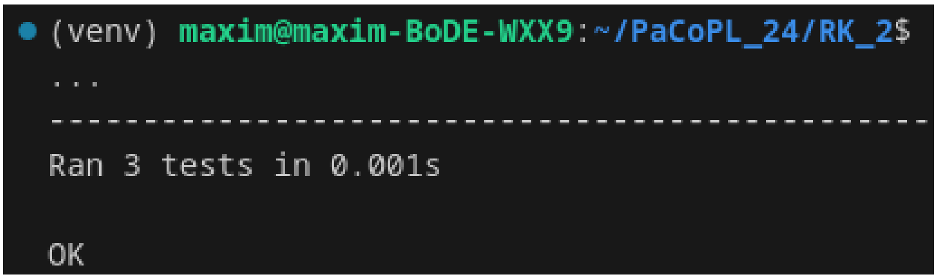
Текст программы

```
import sys
from pathlib import Path
sys.path.append(str(Path(__file__).resolve().parent.parent))
from RK_1.Panshin_16Var import
Book,BookStore,Book_BookStore,task_1,task_2,task_3,get_many_to_many,get
_one_to_many
import unittest

class TestBookStore (unittest.TestCase):
    def setUp(self):
        self.books = [
        Book(1,"Темная башня",1799,1),
        Book(2,"Молот вулкана",400,3),
        Book(3,"Повелитель мух",499,4),
        Book(4,"1984",999,2),
        Book(5,"Противостояние",2100,5),
        Book(6,"Гарри Поттер и узник азкабана", 1701.23,3),

    ]
        self.stores = [
            BookStore(1,"Pioner Bookstore"),
            BookStore(2,"Garage Bookshop"),
            BookStore(3,"Bookbridge"),
            BookStore(4,"Ходасевич"),
            BookStore(5,"Гиперион")
        ]

        self.book_bookstore = [
            Book_BookStore(1,1),
            Book_BookStore(2,3),
            Book_BookStore(3,4),
            Book_BookStore(4,2),
            Book_BookStore(5,5),

            Book_BookStore(2,2),
            Book_BookStore(5,2),
            Book_BookStore(3,5)
        ]

    def test_task_1(self):
        one_to_many = get_one_to_many(self.books,self.stores)
        result = task_1(self.stores,one_to_many)
```



```
        expected = {
            'Pioner Bookstore': ['Темная башня'],
            'Garage Bookshop': ['1984'],
            'Bookbridge': ['Молот вулкана', 'Гарри Поттер и узник
азкабана']
        }
        self.assertEqual(result,expected)

    def test_task_2(self):
        one_to_many = get_one_to_many(self.books,self.stores)
        result = task_2(self.stores, one_to_many)

        expected = {'Гиперион': 2100.0,
                    'Pioner Bookstore': 1799.0,
                    'Bookbridge': 1050.62,
                    'Garage Bookshop': 999.0,
                    'Ходасевич': 499.0
                    }
        self.assertEqual(result,expected)

    def test_task_3(self):
        many_to_many = get_many_to_many(self.books, self.stores,
self.book_bookstore )
        result = task_3(self.books,many_to_many)
        expected =  {'Повелитель мух': ['Ходасевич', 'Гиперион'],
                     'Противостояние': ['Гиперион', 'Garage Bookshop']}
        self.assertEqual(result,expected)

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

Результат программы



```
(venv) maxim@maxim-BoDE-WXX9:~/PaCoPL_24/RK_2$
...
-----------------------------------------------
Ran 3 tests in 0.001s

OK
```