

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

Факультет ИУ  
Кафедра ИУ5

Курс «Основы информатики»

Отчет по Рубежному контролю №2  
Вариант В 13

Выполнил:  
студент группы ИУ5-33Б:

Селиханович Д.С.  
Подпись и дата:

Проверил:  
преподаватель каф.

\_\_\_\_\_  
Подпись и дата:

Москва, 2024 г.

## 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

`\RK2\models.py`

```
class Book:
    """Книга"""
    def __init__(self, id, title, author, year, lib_id):
        self.id = id
        self.title = title
        self.author = author
        self.year = year
        self.lib_id = lib_id

class Library:
    """Библиотека"""
    def __init__(self, id, name):
        self.id = id
        self.name = name

class BookLib:
    """
    'Книги в библиотеке' для реализации
    связи многие-ко-многим
    """
    def __init__(self, lib_id, book_id, count):
        self.lib_id = lib_id
        self.book_id = book_id
        self.count = count
```

`\RK2\data.py`

```
from models import Library, Book, BookLib

# Библиотеки
libs = [
    Library(1, 'Главная библиотека'),
    Library(2, 'Университетская библиотека'),
    Library(3, 'Городская библиотека'),
    Library(4, 'Детская библиотека'),
    Library(5, 'Научная библиотека'),
    Library(6, 'Историческая библиотека'),
]
```

```

# Книги
books = [
    Book(1, 'Война и мир', 'Лев Толстой', 1865, 1),
    Book(2, 'Анна Каренина', 'Лев Толстой', 1877, 1),
    Book(3, 'Преступление и наказание', 'Федор Достоевский', 1866, 2),
    Book(4, 'Братья Карамазовы', 'Федор Достоевский', 1880, 2),
    Book(5, '1984', 'Джордж Оруэлл', 1949, 3),
    Book(6, 'Скотный двор', 'Джордж Оруэлл', 1945, 3),
    Book(7, 'Мастер и Маргарита', 'Михаил Булгаков', 1939, 4),
    Book(8, 'Собачье сердце', 'Михаил Булгаков', 1925, 5),
    Book(9, 'Доктор Живаго', 'Борис Пастернак', 1957, 6),
    Book(10, 'Лолита', 'Владимир Набоков', 1955, 6),
]

```

```

# Связь книг и библиотек
books_libs = [
    BookLib(1, 1, 5),
    BookLib(1, 2, 3),
    BookLib(2, 3, 2),
    BookLib(2, 4, 1),
    BookLib(3, 5, 1),
    BookLib(3, 6, 2),
    BookLib(4, 7, 1),
    BookLib(5, 8, 3),
    BookLib(6, 9, 2),
    BookLib(6, 10, 1),
    BookLib(2, 1, 9),
    BookLib(3, 2, 8),
    BookLib(4, 3, 7),
    BookLib(5, 4, 6),
    BookLib(6, 5, 5),
    BookLib(6, 6, 4),
    BookLib(1, 7, 9),
    BookLib(3, 8, 3),
    BookLib(2, 9, 3),
    BookLib(1, 10, 10),
]

```

**\RK2\RK2.py**

```

import unittest
from models import Book
from models import Library
from models import BookLib

class TestBook(unittest.TestCase):
    def test_book_init(self):
        id = 1
        title = 'Война и мир'

```

```

        author = 'Лев Толстой'
        year = 1865
        lib_id = 1

        book = Book(id, title, author, year, lib_id)

        self.assertEqual(book.id, id)
        self.assertEqual(book.title, title)
        self.assertEqual(book.author, author)
        self.assertEqual(book.year, year)
        self.assertEqual(book.lib_id, lib_id)

class TestLibrary(unittest.TestCase):
    def test_library_init(self):

        id = 1
        name = 'Главная библиотека'

        library = Library(id, name)

        self.assertEqual(library.id, id)
        self.assertEqual(library.name, name)

class TestBookLib(unittest.TestCase):
    def test_book_lib_init(self):
        lib_id = 1
        book_id = 1
        count = 10

        book_lib = BookLib(lib_id, book_id, count)

        self.assertEqual(book_lib.lib_id, lib_id)
        self.assertEqual(book_lib.book_id, book_id)
        self.assertEqual(book_lib.count, count)

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

```

## Анализ результатов

...

-----  
Ran 3 tests in 0.002s

OK