

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования «Петрозаводский государственный университет»

Институт математики и информационных технологий
Кафедра информатики и математического обеспечения

Сарбаев Артур Мухаматович

Отчет по дисциплине
«Методы тестирования ПО»

Направление 09.03.04 — Программная инженерия

Преподаватель:
к.ф.-м.н., доцент К. А. Кулаков

Петрозаводск

2020

Содержание

1.Объект тестирования	2
1.1. Описание программы	2
1.2. Архитектура программы	3
1.3. Функции программы	3
2. Стратегия тестирования	3
2.1. Структура объекта тестирования	3
2.2. Стратегия блочного тестирования	4
2.3. Стратегия интеграционного тестирования	4
2.4. Стратегия аттестационного тестирования	5
2.5. Критерии прохождения тестов	5
2.6. Критерии приостановки тестирования	5
2.7. Критерии возобновления тестирования	5
3. Детальный план тестов	5
3.1. Модульное тестирование	5
3.2. Интеграционное тестирование	10
3.3 Аттестационное тестирование	13
4. Журнал тестирования	17
5. Журнал найденных ошибок	18
6. Результаты	19

1.Объект тестирования

1.1. Описание программы

Программа создана для чтение электронных книг в формате epub, fb2, pdf, djvu, mobi2, azw, azw3, azw4, комиксов в форматах cbr, cbz и файлов md.

Используемые в системе технологии:

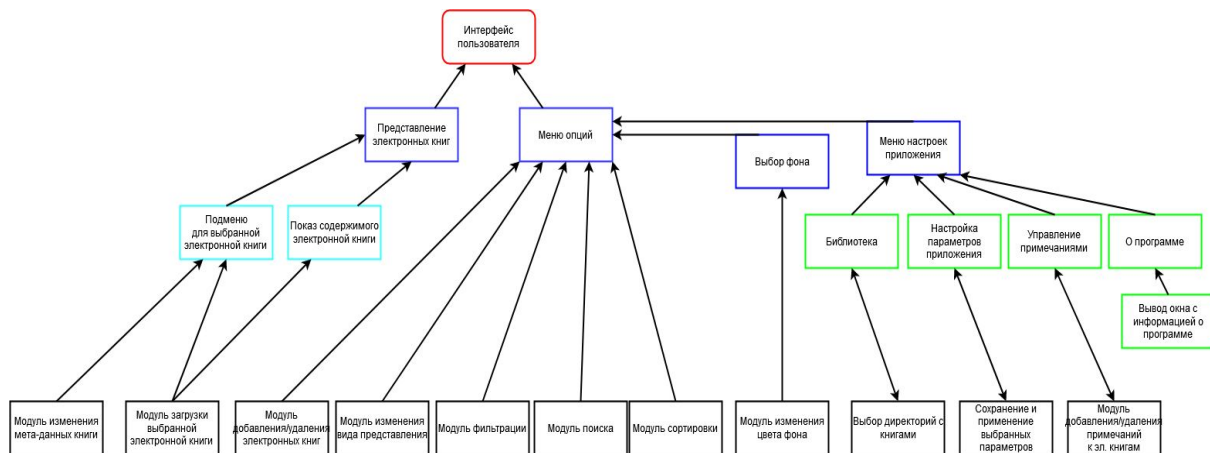
- Python3
- SQLite

Используемые библиотеки языка Python:

- PyQt5
- python-lxml
- python-beautifulsoup4
- python-xmltodict
- python-pymupdf
- python-djvulibre

- python-markdown
- textile

1.2. Архитектура программы



Описание модулей:

1. Главный модуль - графический интерфейс пользователя, Он делится на два взаимодействующих модуля:

1.1. Представление библиотеки

- library - представление всех книг библиотеки
- context_menu - управление конкретной книгой
 - delete_book - удаление книги из библиотеки и информации о ней из БД
 - change_metadata - изменение данных о книге в БД
- get_selection - показ выбранной книги в отдельном окне
- get_progress - модуль отслеживания прогресса чтения книги
- close_book_event - сохранение в БД информации о состоянии книги

1.2. Меню опций (ToolBar), с помощью которого можно управлять этой графической библиотекой.

- open_book - добавление книги в библиотеку(обработка содержимого)
- get_book - просмотр содержимого выбранной книги
- filter - Модуль фильтрации книг по параметрам

- search - поиск книг в библиотеке по параметрам
- sort - сортировка книг в библиотеке
- tableView - выбор типа представления библиотеки
- settingsMenu - модуль управления приложением
 - add_books - добавление целой папки(библиотеки) в графическую библиотеку с сохранением их в БД
 - paramsMenu - установки для отображения содержимого книги (цвет фона, размер шрифта и тд.)
 - about - вывод информации о программе
 - annotations - управления примечаниями, оставленными в книге.
 - logs - вывод лога программы в отдельное окно

1.3. Функции программы

Из-за большого объема модулей в программе было решено сократить часть функций.

Поэтому в приложении будут рассмотрены следующие функции:

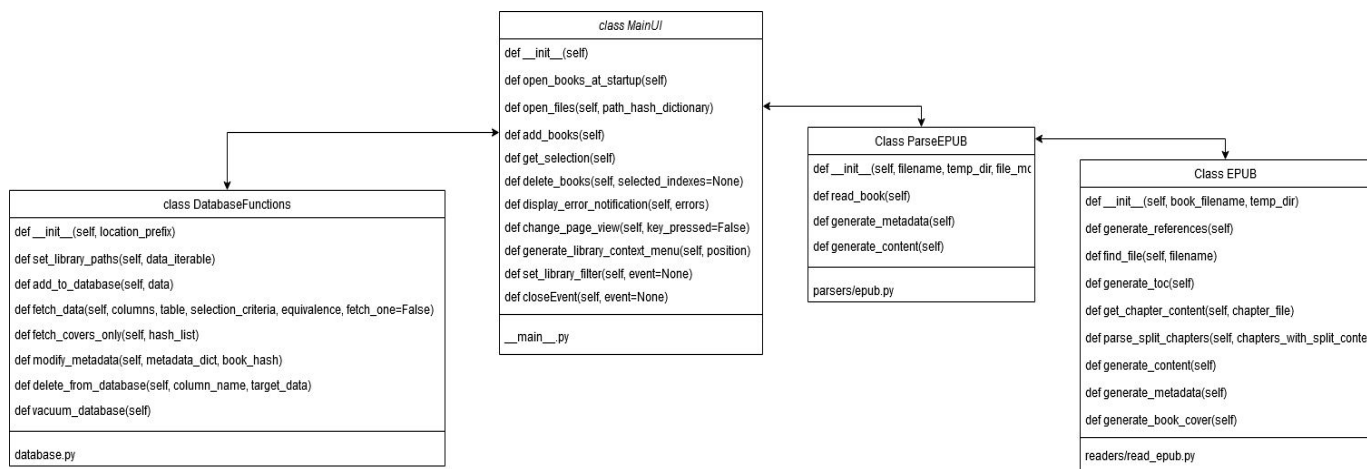
- 1.Открытие электронной книги (обработка файлов книги)
2. Просмотр содержимого книги
- 3.Добавление библиотеки
- 4.Удаление эл. книги из библиотеки

В аттестационном тестировании принимают участие все вышеописанные функции.

2. Стратегия тестирования

2.1. Структура объекта тестирования

Сервисы реализуются за счет программных модулей. Программная архитектура тестируемых модулей системы представлена на Рис. 1



Система объемная, она не может быть в полной мере протестирована в рамках прохождения данного курса. Таким образом, в рамках данной работы будут протестированы следующие модули:

1. Модуль чтения электронных книг в формате epub (ParseEPUB) - обрабатывает все файлы и ищет нужные данные в заданном .epub архиве.

Модуль принимает три аргумента:

1. filename - путь к файлу-электронной книге;
2. temp_dir - путь, где хранятся уже обработанные книги;
3. file_md5 - хэш от названия книги для дальнейшей обработки книги

Возвращает экземпляр класса со всем содержанием книги (обложка книги, информация о главах книги, содержание этих глав, метаданные книги)

2. Модуль добавления библиотеки - выбранную пользователем папку сначала обрабатывает считыватель эл. книг, а после него весь список данных добавляется в созданную модулем БД и выводит все книги в библиотеку (графический элемент, где отображаются загруженные книги).

Модуль принимает два аргумента:

1. file_list - массив данных всех книг;
2. database_path - путь к БД;

Модуль возвращает все книги в выбранной папке в графическую библиотеку.

3. Модуль удаления книги из библиотеки - удаление из библиотеки и из БД выбранной пользователем книги.

Модуль принимает один аргумент:

1. selected_index - индекс выбранной книги в массиве загруженных в библиотеку книг;

В результате модуль убирает из графической библиотеки выбранную книгу.

4. Модуль перевода программы на язык системы - обнаруживает язык системы и, если есть для этого языка перевод, переводит все надписи в программе.

Модуль принимает один аргумент:

locale - строку с языком системы

Модуль возвращает в лог системы что перевод либо найден и устанавливает его, либо не найден и язык остается по умолчанию английским.

2.2. Стратегия блочного тестирования

Первый вид тестирования, которому будет подвержена система, будет блочное тестирование.

Блочные тесты проверяют работоспособность отдельных функций программы. Их поведение не зависит от результатов работы других тестируемых функций. Для их проверки в качестве аргументов при вызове функции передаются различные значения, после чего возвращаемый результат функции проверяется на наличие ошибок.

В реализации тестов используются библиотека unittest языка python, которая облегчает создание и автоматизацию тестов.

2.3. Стратегия интеграционного тестирования

Второй этап - интеграционное тестирование.

При интеграционном тестировании проверяется взаимодействие существующих модулей. Описанные выше, в блочном тестировании, функции, принимают ввод и отдают вывод другим. Таким образом проверяется отсутствие ошибок во взаимодействии их между собой.

Интеграционное тестирование будет проведено для следующих взаимодействий между модулями:

- ParseEPUB и EPUB

Начало теста -> Вызов метода generate_toc() -> Вызов метода generate_content() -> Конец теста

- MainUI и Translator

Начало теста -> Вызов метода load() -> вызов метода installTranslator() -> Конец теста

2.4. Стратегия аттестационного тестирования

Аттестационное тестирование будет проводиться методом «живого человека». В роли такого человека выступает сам автор тестирования.

Тестирующий человек, по заранее заданным инструкциям (TestCases), производит требуемые действия и сверяется с заранее заданными результатами. Тест считается пройденным, если ожидаемый результат совпадает с фактическим результатом. В противном случае тест считается не пройденным.

2.5 Стратегия специального тестирования

Четвертый этап – специальное тестирование. В качестве специальных тестов будет проводиться нагрузочное тестирование. Нагрузочное тестирование – это процесс умышленной нагрузки системы, с целью определения показателей производительности, времени отклика, проверки соответствия требованиям, которые были предъявлены к данной системе или отдельному устройству. Целью данного тестирования является оценка производительности и работоспособности тестируемого модуля.

Специальные тесты:

1. Открытие электронной книги большого объема
2. Одновременное открытие большого количества книг(30)

2.6. Критерии прохождения тестов

Тест считается успешно пройденным, если ожидаемый и фактический результаты совпадают. Если тест завершается неудачей, то перед принятием решения целесообразно проверить правильность самого теста. Если тест завершился неудачей и тест реализован правильно, то производится заключение о найденной ошибке. Тестирование считается пройденным, если во время его прохождения не выявлено критических ошибок, а процент не пройденных тестов меньше 1.

2.7. Критерии приостановки тестирования

Тестирование должно быть приостановлено, если количество не пройденных тестов превысит 40% от общего количества.

2.8. Критерии возобновления тестирования

Тестирование возобновляется после исправления ошибок, выявленных при предыдущем тестировании.

3. Детальный план тестов

3.1 Подготовка к тестированию

Для проведения полноценного тестирования требуются заранее подготовленные локальные файлы.

В папке tests/data/EPUB содержатся следующие файлы:

1. test_book.epub - валидная книга, небольшого размера
2. test_book_with_cover.epub - книга с измененным названием обложки(testtest.jpeg)
3. test_book_without_cover.epub - книга с удаленной обложкой
4. test_find_file.epub - книга, в которой есть файл с длинным (255 знаков) названием и файл со спецсимволами в имени
5. not_exist.epub - пустой файл с расширением epub
6. Kyerroll_L_Alisa_V_Strane_ChudesII.epub - книга с большим количеством файлов и, соответственно, большим объемом (63МБ, том войны и мира, к примеру весит меньше 1 МБ)
7. Vakulin_A_Gravitaciya_I_Yefir.epub - валидная книга с нестандартным файлом оглавления
8. Vakulin_A_Gravitaciya_I_Yefir.epub - cover - изображение - обложка книги
Vakulin_A_Gravitaciya_I_Yefir.epub
9. content0.html - содержание одной из глав книги test_book.epub
10. test_book_generate_metadata.epub - книга с уже удаленными метаданными

11. test_book_not_valid_toc.epub - книга с удаленным файлом оглавления
12. Zamyatin_E_Spisokshkolnoy_Myi.epub - валидная книга
13. pg63992-images.epub - валидная книга с нестандартным расположением и именами файлов

При дальнейшем описании тестов во входных параметрах номер используемого файла будет написан в круглых скобах.

3.2. Модульное тестирование

Тест	M1
Цель теста	Проверка правильного нахождения обложки книги
Тип теста	Общий
Объект тестирования	функция generate_book_cover() класса EPUB
Входные параметры	self - валидный экземпляр класса EPUB (1)
Косвенные данные	
Ожидаемый результат	Найдено верное изображение

Тест	M2
Цель теста	Проверка правильного нахождения обложки книги
Тип теста	Общий
Объект тестирования	функция generate_book_cover() класса EPUB
Входные параметры	self - экземпляр класса EPUB, в котором ссылка на файл с отсутствующей обложкой (3)

Косвенные данные	
Ожидаемый результат	Возврат значения None

Тест	M3
Цель теста	Проверка правильного нахождения обложки книги
Тип теста	Общий
Объект тестирования	функция generate_book_cover() класса EPUB
Входные параметры	Экземпляр класса EPUB, в котором ссылка на файл с обложкой с другим именем (2)
Косвенные данные	
Ожидаемый результат	Найдено верное изображение

Тест	M4
Цель теста	Проверка правильного нахождения метаданных в книжке
Тип теста	Общий
Объект тестирования	функция generate_metadata() класса EPUB
Входные параметры	self - валидный экземпляр класса, в котором ссылка на файл с известными метаданными (12)
Косвенные данные	
Ожидаемый результат	запись в self.metadata заданных метаданных

Тест	M5
-------------	-----------

Цель теста	Проверка правильных действий при не нахождении метаданных в файле электронной книги
Тип теста	Общий
Объект тестирования	функция generate_metadata() класса EPUB
Входные параметры	self - валидный экземпляр класса, в котором ссылка на файл с удаленными метаданными (10)
Косвенные данные	
Ожидаемый результат	запись в self.metadata значений None

Тест	М6
Цель теста	Проверка правильного нахождения содержания книги
Тип теста	Общий
Объект тестирования	функция generate_content() класса EPUB
Входные параметры	self - экземпляр класса, в котором ссылка на валидный файл (7)
Косвенные данные	
Ожидаемый результат	Все прочитанные данные записываются в переменные экземпляра класса

Тест	М7
Цель теста	Проверка правильного нахождения содержания книги
Тип теста	Общий
Объект тестирования	функция generate_content() класса EPUB

Входные параметры	self - экземпляр класса, в котором ссылка на валидный файл с нестандартным psx файлом (12)
Косвенные данные	
Ожидаемый результат	Все прочитанные данные записываются в переменные экземпляра класса

Тест	M8
Цель теста	Проверка правильного нахождения содержания определенной главы книги
Тип теста	Общий
Объект тестирования	функция <code>get_chapter_content()</code> класса EPUB
Входные параметры	self - валидный экземпляр класса, в котором ссылка на валидный файл, (7) file - название файла с содерж. (9)
Косвенные данные	
Ожидаемый результат	строки с содержанием главы

Тест	M9
Цель теста	Проверка правильного нахождения информации о содержании книги
Тип теста	Общий
Объект тестирования	функция <code>generate_toc()</code> класса EPUB
Входные параметры	self - экземпляр класса EPUB, в котором ссылка на валидную книгу(12)
Косвенные данные	
Ожидаемый результат	Список с названиями глав

Тест	M10
Цель теста	Проверка правильного нахождения информации о содержании книги
Тип теста	Общий
Объект тестирования	функция generate_toc() класса EPUB
Входные параметры	self - экземпляр класса EPUB, в котором ссылка на файл с нестандартным .npx файлом(7)
Косвенные данные	
Ожидаемый результат	Список с названиями глав

Тест	M11
Цель теста	Проверка правильного нахождения файлов со спецсимволами в названии
Тип теста	Общий
Объект тестирования	функция find_file() класса EPUB
Входные параметры	self - экземпляр класса EPUB (4), file - название нужного файла с нестандартным расширением
Косвенные данные	
Ожидаемый результат	Возврат значения file

Тест	M12
Цель теста	Проверка правильного нахождения файлов со сверхдлинным названием(255 символов)
Тип теста	Пограничный
Объект тестирования	функция find_file() класса EPUB

Входные параметры	self - экземпляр класса EPUB (4), file - название нужного файла с нестандартным расширением
Косвенные данные	
Ожидаемый результат	Возврат значения file

3.3. Интеграционное тестирование

Тест	И1
Цель теста	Тест проверяет корректность передачи списка глав книги из модуля generate_toc() в модуль поиска содержания глав generate_content()
Тип теста	общий
Объект тестирования	generate_toc(), generate_content()
Входные параметры	self - экземпляр класса ParseEPUB, указывающий на файл большого объема (6)
Косвенные данные	
Ожидаемый результат	Значение получено из первоначального модуля generate_toc() и успешно обработано модулем generate_content() без появления непредвиденных ошибок и аномалий

Тест	И2
Цель теста	Тест проверяет корректность передачи списка глав книги из модуля generate_toc() в модуль поиска содержания глав generate_content()
Тип теста	отрицательный

Объект тестирования	<code>generate_toc()</code> , <code>generate_content()</code>
Входные параметры	<code>self</code> - экземпляр класса <code>ParseEPUB</code> , указывающий на случайный не валидный файл с расширением <code>.epub</code> (5)
Косвенные данные	
Ожидаемый результат	Возврат <code>None</code> при попытке чтения книги

Тест	ИЗ
Цель теста	Тест проверяет корректность передачи списка глав книги из модуля <code>generate_toc()</code> в модуль поиска содержания глав <code>generate_content()</code>
Тип теста	общий
Объект тестирования	<code>generate_toc()</code> , <code>generate_content()</code>
Входные параметры	<code>self</code> - экземпляр класса <code>ParseEPUB</code> , указывающий на файл с отсутствующим файлом оглавления (7)
Косвенные данные	
Ожидаемый результат	В модуль <code>generate_content()</code> из модуля <code>generate_toc()</code> передается список с порядковыми номерами глав, без названий

Тест	И4
Цель теста	Тест проверяет корректность передачи языка системы из модуля <code>translator.load()</code> в модуль установки

	переведенного текста надписей программы installTranslator
Тип теста	общий
Объект тестирования	translator.load(), installTranslator()
Входные параметры	нет
Косвенные данные	система с установленным русским языком
Ожидаемый результат	Все надписи в программе на русском языке

Тест	И5
Цель теста	Тест проверяет корректность передачи языка системы из модуля translator.load() в модуль установки переведенного текста надписей программы installTranslator
Тип теста	отрицательный
Объект тестирования	translator.load(), installTranslator()
Входные параметры	нет
Косвенные данные	система с установленным казахским языком, для которого перевод отсутствует
Ожидаемый результат	Все надписи в программе на языке по умолчанию - английском

3.4. Аттестационное тестирование

Тест	А1
Цель теста	Проверка открытия книги в случае, когда программа выбирается в

	качестве “открыть файл с помощью.. lector” если файл большого объема
Тип теста	специальный
Объект тестирования	функциональность модуля чтения книги
Входные параметры	программа открывается из контекстного меню ‘открыть файл с помощью’ lector для файла Kyerroll_L_Alisa_V_Strane_Chudesll.e pub
Косвенные данные	нет
Ожидаемый результат	книга читается, все ее данные корректны

Тест	A2
Цель теста	Проверка открытия книги в случае, когда программа выбирается в качестве “открыть файл с помощью.. lector”
Тип теста	общий
Объект тестирования	функциональность модуля чтения книги
Входные параметры	программа открывается из контекстного меню ‘открыть файл с помощью’ lector для книги rg63992-images.epub на английском языке
Косвенные данные	нет
Ожидаемый результат	книга читается, все ее данные корректны

Тест	A3
Цель теста	Проверка открытия книги в случае, когда программа запущена и книга выбирается в окне открытия книги
Тип теста	общий

Объект тестирования	функциональность модуля чтения книги
Входные параметры	Валидная книга на русском языке Kyerroll_L_Alisa_V_Strane_Chudesll.e pub
Косвенные данные	нет созданной базы данных, в пути к книге есть русские символы
Ожидаемый результат	книга читается, все ее данные корректны

Тест	A4
Цель теста	Проверка открытия книги в случае, когда программа запущена и книга выбирается в окне открытия книги
Тип теста	общий
Объект тестирования	функциональность модуля чтения книги
Входные параметры	Валидная книга на русском языке Kyerroll_L_Alisa_V_Strane_Chudesll.e pub
Косвенные данные	есть база данных, но с отличным пути библиотеки от пути книги
Ожидаемый результат	книга читается, все ее данные корректны

Тест	A5
Цель теста	Проверка открытия книги в случае, когда программа запущена и книга выбирается в окне открытия книги
Тип теста	общий
Объект тестирования	функциональность модуля чтения книги
Входные параметры	Валидная книга на русском языке Kyerroll_L_Alisa_V_Strane_Chudesll.e pub
Косвенные данные	база данных создана, книга находится в папке библиотеке

Ожидаемый результат	книга читается, все ее данные корректны
---------------------	---

Тест	A6
Цель теста	Проверка загрузки всех книг в папке, при выборе папки в окне “выбор библиотеки”
Тип теста	общий
Объект тестирования	функциональность модуля открытия папки-библиотеки
Входные параметры	путь к папке с валидными книгами
Косвенные данные	нет созданной базы данных
Ожидаемый результат	Все книги из папки загружены с корректными данными

Тест	A7
Цель теста	Проверка загрузки всех книг в папке, при выборе папки в окне “выбор библиотеки”
Тип теста	общий
Объект тестирования	функциональность модуля открытия папки-библиотеки
Входные параметры	путь к папке с валидными книгами, но некоторые книги одинаковые
Косвенные данные	нет созданной базы данных
Ожидаемый результат	В библиотеку добавляются книги только в одном экземпляре

Тест	A8
-------------	-----------

Цель теста	Проверка локализации программы
Тип теста	общий
Объект тестирования	функциональность модуля локализации программы
Входные параметры	нет
Косвенные данные	система с русской локалью
Ожидаемый результат	Почти все надписи в программе на русском языке

Тест	A9
Цель теста	Проверка локализации программы
Тип теста	общий
Объект тестирования	функциональность модуля локализации программы
Входные параметры	нет
Косвенные данные	система с вьетнамской локалью
Ожидаемый результат	Все надписи в программе на английском языке

Тест	A10
Цель теста	Проверка удаления выбранной пользователем книги
Тип теста	общий
Объект тестирования	функциональность модуля удаления книги из библиотеки
Входные параметры	selected_index - индекс выбранной книги
Косвенные данные	
Ожидаемый результат	Книга из графической и запись о книге в БД удаляются

3.5. Специальное тестирование

Тест	С1
Цель теста	Тест проверяет корректность работы модуля загрузки книги при обработке электронной книги большого объема за приемлемое время
Тип теста	общий
Объект тестирования	Нагрузочная функциональность модуля обработка книги
Входные параметры	Книга большого объема
Косвенные данные	нет
Ожидаемый результат	Книга корректно обработана, все необходимые данные сохранены, данные о книге записаны в БД.

Тест	С2
Цель теста	Тест проверяет корректность работы модуля добавления библиотеки при обработке большого количества (30) электронных книг.
Тип теста	общий
Объект тестирования	Нагрузочная функциональность модуля добавления библиотеки
Входные параметры	Папка с большим количеством книг
Косвенные данные	нет
Ожидаемый результат	Все книги загружены, все их данные корректно обработаны, вся эта информация записана в БД.

4. Примеры реализации тестов

Тест проверки правильного нахождения метаданных книги(M4)

```
def test_valid_generate_metadata(self):
    filename = "../data/EPUB/Zamyatin_E_Spiskshkolnoy_Myi.epub"
    temp_dir = "../data/EPUB/"
    book = EPUB(filename, temp_dir)
    Metadata = collections.namedtuple(
        'Metadata', ['title', 'author', 'year', 'isbn', 'tags', 'cover'])
    testMetadata = Metadata("Мы", "Евгений Иванович Замятин", 1920, None,
        ["Социальная фантастика", "Литература 20 века", "Учебная
литература"], book.generate_book_cover())
    book.generate_metadata()
    result = book.metadata
    self.assertEqual(result, testMetadata)
```

Тест проверки правильного нахождения обложки книги, в случае, когда название файла обложки (по умолчанию cover) изменено на testtest

```
def test_cover_find_with_other_name(self):
    filename = "../data/EPUB/test_book_with_cover.epub"
    temp_dir = "../data/EPUB/"
    book = EPUB(filename, temp_dir)
    result = book.generate_book_cover()
    self.zip_file = zipfile.ZipFile(
        filename, mode='r', allowZip64=True)
    book_cover = self.zip_file.read(book.find_file("testtest.jpeg"))
    self.assertEqual(result, book_cover)
```

5. Журнал тестирования

Номер	Дата	Результат	Отчет об ошибке
M1	25.11.2020	пройден	
M2	29.11.2020	пройден	
M3	29.11.2020	пройден	
M4	29.11.2020	пройден	
M5	29.11.2020	пройден	
M6	29.11.2020	пройден	
M7	29.11.2020	провален	Отчет об ошибке 1

M8	2.12.2020	пройден	
M9	2.12.2020	пройден	
M10	2.12.2020	провален	Отчет об ошибке 2
M11	2.12.2020	пройден	
M12	3.12.2020	пройден	
И1	5.12.2020	пройден	
И2	5.12.2020	провален	Отчет об ошибке 3
И3	5.12.2020	пройден	
И4	9.12.2020	пройден	
И5	9.12.2020	пройден	
A1	9.12.2020	пройден	
A2	9.12.2020	пройден	
A3	9.12.2020	провален	Отчет об ошибке 4
A4	9.12.2020	провален	Отчет об ошибке 5
A5	9.12.2020	пройден	
A6	9.12.2020	пройден	
A7	9.12.2020	провален	Отчет об ошибке 6
A8	9.12.2020	пройден	
A9	12.12.2020	пройден	
A10	12.12.2020	пройден	
C1	15.12.2020	пройден	
C2	15.12.2020	пройден	

6. Журнал найденных ошибок

Отчет об ошибке №1:

Тест: M7

Дата составления отчета: 12.12.2020

Ожидаемый результат: успешное чтение содержимого книг

Фактический результат: функция падает с ошибкой сегментирования (SIGSEGV)

Отчет об ошибке №2:

Тест: M10

Дата составления отчета: 12.12.2020

Ожидаемый результат: список с названием глав

Фактический результат: список с порядковыми номерами глав

Отчет об ошибке №3:

Тест: I2

Дата составления отчета: 12.12.2020

Ожидаемый результат: успешное завершение работы функции с возвратом значения None и предупреждения

Фактический результат: функция экстренно завершается с ошибкой о том, что это не zip файл

Отчет об ошибке №4:

Тест: A3

Дата составления отчета: 12.12.2020

Ожидаемый результат: вывод на экран содержимого открытой книги

Фактический результат: книга не открывается, в лог ничего не записывается

Отчет об ошибке №5:

Тест: A4

Дата составления отчета: 12.12.2020

Ожидаемый результат: вывод на экран содержимого открытой книги

Фактический результат: книга не открывается, в лог ничего не записывается

Отчет об ошибке №6:

Тест: А7

Дата составления отчета: 12.12.2020

Ожидаемый результат: в библиотеке появляются только уникальные книги

Фактический результат: в библиотеке появляется несколько одинаковых книг, отличающиеся только названием файла книги

7. Покрывтие кода тестами

Расчет тестового покрытия относительно исполняемого кода программного обеспечения проводится по формуле:

$T_{cov} = L_{tc}/L_{code} * 100\%$, где

T_{cov} - тестовое покрытие,

L_{tc} - количество строк, покрытые тестами,

L_{code} - общее количество строк тестируемых модулей

Тогда $T_{cov} = (969 / 1870) * 100 = 52 \%$

8. Результаты

Во время тестирования приложения было выявлено несколько ошибок, представленных в отчетах об ошибках №1–6. Среди найденных ошибок все являются критическими и должны быть исправлены, кроме ошибки №2.

По результатам тестирования необходимо исправить найденные ошибки и повторно протестировать приложение.