МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Тверской государственный технический университет»

(ТвГТУ)

Кафедра “Программного обеспечения”

**Лабораторная работа**

по дисциплине “Конструирование программных систем”

Тема: «Unit-тестирование»

Выполнил: студент группы

ПИН 17.06

Иванов Р.В

Проверил:

Мальков А.А

Тверь 2020

# Класс тестов

public class BookServiceTest {  
  
 private IBookService bookService = new BookServiceImpl();  
 private IAuthorService authorService = new AuthorServiceImpl();  
 private Book book = new Book("name", 123, 123, "000-0000000009", "publisher", authorService.getAuthor(32), false);  
 private long bookId;  
  
 public BookServiceTest() throws DataException {  
 }  
  
  
 @Test  
 @Before  
 public void addBookToDateBaseTest() throws DataException {  
  
 bookService.addBookToDateBase(book.getBookName(), book.getReleaseYear(), book.getPageCount(), book.getISBN(), book.getPublisher(), book.getAuthor().getAuthorId());  
 Book book1 = bookService.searchBookByISBN(book.getISBN());  
 bookId = book1.getBookId();  
 Assert.*assertTrue*(bookService.getAll().stream().anyMatch(e -> e.equals(book)));  
  
 }  
  
 @Test  
 public void getBookTest() throws DataException {  
 Assert.*assertNotNull*(bookService.getBook(15));  
 }  
  
 @Test  
 public void getAllTest() throws DataException {  
 List<Book> books = bookService.getAll();  
 Assert.*assertNotNull*(books);  
 }  
  
  
 @Test  
 public void searchBookByPartOfBookNameTest() throws DataException {  
 Assert.*assertTrue*(bookService.searchBookByPartOfBookName(book.getBookName()).stream().anyMatch(e -> e.equals(book)));  
 }  
  
 @Test  
 public void searchBookByPartOfAuthorNameTest() throws DataException {  
  
 Assert.*assertTrue*(bookService.searchBookByPartOfAuthorName(book.getAuthor().getName()).stream().anyMatch(e -> e.equals(book)));  
 }  
  
 @Test  
 public void searchBookByISBNTest() throws DataException {  
 Book book1 = bookService.searchBookByISBN(book.getISBN());  
 bookId = book1.getBookId();  
 Assert.*assertEquals*(book1, book);  
 }  
  
 @Test  
 public void searchBookByYearRangeTest() throws DataException {  
 Assert.*assertTrue*(bookService.searchBookByYearRange(122, 124).stream().anyMatch(e -> e.equals(book)));  
 }  
  
 @Test  
 public void searchBookByYearAndPageCountAndPartNameTest() throws DataException {  
 Assert.*assertTrue*(bookService.searchBookByYearAndPageCountAndPartName(book.getReleaseYear(), book.getPageCount(), book.getBookName()).stream().anyMatch(e -> e.equals(book)));  
 }  
  
 @Test  
 public void searchBookWhereUserBookMarkTest() throws DataException {  
 Assert.*assertNotNull*(bookService.searchBookWhereUserBookMark(2));  
 }  
  
 @Test  
 public void getAllNotTakenTest() throws DataException {  
 Assert.*assertNotNull*(bookService.getAllNotTaken());  
 }  
  
}

|  |  |  |  |
| --- | --- | --- | --- |
| Требование | Тест | Ожидание | Комментарии |
| Добавление книги в библиотеку | Добавление книги со следующими параметрами:  "name", 123, 123, "000-0000000009", "publisher", 32, false | Книга будет добавлена | Проверяем работу функции  addBookToDateBase(String bookName, int releaseYear, int pageCount, String ISBN, String publisher, long authorId) |
| Получить список всех книг библиотеки | Получение всех книг из базы данных библиотеки | Все книги будут получены | Проверяем работу функции  getAll() |
| Поиск книги по названию | Поиск книги по названию «name» | Такая книга будет найдена | Проверяем работу функции  searchBookByPartOfBookName(String partOfName) |
| Поиск книги по части имени автора | Поиск книги по части имени автора «author» | Такая книга будет найдена | Проверяем работу функции  searchBookByPartOfAuthorName(String partOfAuthorName) |
| Поиск книги по ISBN | Поиск книги по ISBN « 000-0000000009» | Такая книга будет найдена | Проверяем работу функции  searchBookByISBN(String ISBN) |
| Поиск книги по промежутку между годами | Поиск книги по промежутку между годами  «122-124» | Такие книги будут найдены | Проверяем работу функции  searchBookByYearRangeTest( int from, int to) |
| Поиск книги по году, количеству страниц и имени | Поиск книги по году, количеству страниц и имени «123,123, name» | Такие книги будут найдены | Проверяем работу функции  searchBookByYearAndPageCountAndPartName(int year, int pageCount, String partOfName) |
| Поиск книг по наличию в ней закладки пользователя | Поиск книг по наличию в ней закладки пользователя с айди «2» | Такие книги будут найдены | Проверяем работу функции  searchBookWhereUserBookMark(long userId) |
| Получить все книги, которые остались в библиотеке | Получить все книги, которые остались в библиотеке | Такие книги будут найдены | Проверяем работу функции  getAllNotTaken() |

# Таблица тестов

# Результаты тестов

