**Test01 - testCreateBookMapDAO:**

Description:

The “testBookMapDAO” test checks the “BookMapDAO(IBookHelper helper)” constructor functions correctly.

Pre-conditions:

A helper exists.

An Id exists.

A map exists.

Post-conditions:

Creates an instance of helper.

Id is set to 1.

A map is created.

Data required:

BookMapDAO(IBookHelper helper)

**Test02 - testCreateBookMapDAOHelperNull:**

Description:

The “testBookMapDAO” test checks whether the IllegalArgumentException is throw in the “BookMapDAO(IHelper helper)” constructor.

Pre-conditions:

The helper is null.

Post-conditions:

An IllegalArgumentException is thrown.

Data required:

BookMapDAO(IBookHelper helper)

**Test03 - testAddBook:**

Description:

The “testAddBook” test checks the “addBook(String author, String title, String callNo)” method functions correctly.

Pre-conditions:

Helper must exist.

Post-conditions:

New book is created with a unique id.

New book is added into a list of books.

New book is returned.

Data required:

addBook(String author, String title, String callNo)

**Test04 - testAddBookAuthorIsNull:**

Description:

The “testAddBookuthorIsNull” test checks the “addBook(String author, String title, String callNo)” method functions correctly.

Pre-conditions:

The book’s author is null.

Post-conditions:

An IllegalArgumentException is thrown.

Data required:

addBook(String author, String title, String callNo)

**Test05 - testAddBookAuthorIsBlank:**

Description:

The “testAddBookAuthorIsBlank” test checks the “addBook(String author, String title, String callNo)” method functions correctly.

Pre-conditions:

The book’s author is blank.

Post-conditions:

An IllegalArgumentException is thrown.

Data required:

addBook(String author, String title, String callNo)

**Test06 - testAddBookTitleIsNull:**

Description:

The “testAddBookTitleIsNull” test checks the “addBook(String author, String title, String callNo)” method functions correctly.

Pre-conditions:

The book’s title is null.

Post-conditions:

An IllegalArgumentException is thrown.

Data required:

addBook(String author, String title, String callNo)

**Test07 - testAddBookTitleIsBlank:**

Description:

The “testAddBookTitleIsBlank” test checks the “addBook(String author, String title, String callNo)” method functions correctly.

Pre-conditions:

The book’s title is blank.

Post-conditions:

An IllegalArgumentException is thrown.

Data required:

addBook(String author, String title, String callNo)

**Test08 - testAddBookCallNumberIsNull:**

Description:

The “testAddBookCallNumberIsNull” test checks the “addBook(String author, String title, String callNo)” method functions correctly.

Pre-conditions:

The book’s call number is null.

Post-conditions:

An IllegalArgumentException is thrown.

Data required:

addBook(String author, String title, String callNo)

**Test09 – testAddBookCallNumberIsBlank:**

Description:

The “testAddBookCallNumberIsBlank” test checks the “addBook(String author, String title, String callNo)” method functions correctly.

Pre-conditions:

The book’s call number is blank.

Post-conditions:

An IllegalArgumentException is thrown.

Data required:

addBook(String author, String title, String callNo)

**Test10 - testGetBookByID:**

Description:

The “testGetBookByID” test checks the “getBookByID(int id)” method functions correctly.

Pre-conditions:

A book must exist.

The book must have a valid id.

The book must be within a list.

Post-conditions:

Returns the book.

Data required:

getBookByID(int id)

**Test11 - testListBooks:**

Description:

The “testListBooks” test checks the “listBooks()” method functions correctly.

Pre-conditions:

A list (collection) on Books must exist.

Post-conditions:

Returns a list of all Books within the collection.

Data required:

listBooks()

**Test12 - testListBooksIsEmpty:**

Description:

The “testListBooksIsEmpty” test checks the “listBooks()” method functions correctly.

Pre-conditions:

A list (collection) on Books must exist.

Post-conditions:

Empty list.

Data required:

listBooks()

**Test13 - testFindBooksByAuthor:**

Description:

The “testFindBooksByAuthor” test checks the “findBooksByAuthor(String author)” method functions correctly.

Pre-conditions:

A book must exist.

The book must have a valid author.

The book must be within a list.

Post-conditions:

Returns the book.

Data required:

findBooksByAuthor(String author)

**Test14 - testFindBooksByAuthorIsEmpty:**

Description:

The “testFindBooksByAuthorIsEmpty” test checks the “findBooksByAuthor(String author)” method functions correctly.

Pre-conditions:

The author must be empty.

Post-conditions:

Returns empty list.

Data required:

findBooksByAuthor(String author)

**Test15 - testFindBooksByAuthorIsNull:**

Description:

The “testFindBooksByAuthorIsNull” test checks the “findBooksByAuthor(String author)” method functions correctly.

Pre-conditions:

The author must be null.

Post-conditions:

Returns empty list.

Data required:

findBooksByAuthor(String author)

**Test16 - testFindBooksByTitle:**

Description:

The “testFindBooksByTitle” test checks the “findBooksByTitle(String title)” method functions correctly.

Pre-conditions:

A book must exist.

The book must have a valid title.

The book must be within a list.

Post-conditions:

Returns the book.

Data required:

findBooksByTitle(String title)

**Test17 - testFindBooksByTitleIsEmpty:**

Description:

The “testFindBooksByTitleIsEmpty” test checks the “findBooksByTitle(String title)” method functions correctly.

Pre-conditions:

The title must be empty.

Post-conditions:

Returns empty list.

Data required:

findBooksByTitle(String title)

**Test18 - testFindBooksByTitleIsNull:**

Description:

The “testFindBooksByTitleIsNull” test checks the “findBooksByTitle(String title)” method functions correctly.

Pre-conditions:

The book must be null.

Post-conditions:

Returns empty list.

Data required:

findBooksByTitle(String title)

**Test19 - testFindBooksByAuthorTitle:**

Description:

The “testFindBooksByAuthorTitle” test checks the “findBooksByAuthorTitle(String author, String title)” method functions correctly.

Pre-conditions:

A book must exist.

The book must have a valid author.

The book must have a valid title.

The book must be within a list.

Post-conditions:

Returns the book.

Data required:

findBooksByAuthorTitle(String author, String title)

**Test20 - testFindBooksByAuthorTitleIsEmpty:**

Description:

The “testFindBooksByAuthorTitleIsEmpty” test checks the “findBooksByAuthorTitle(String author, String title)” method functions correctly.

Pre-conditions:

The author must be empty.

The title must be empty.

Post-conditions:

Returns empty list.

Data required:

findBooksByAuthorTitle(String author, String title)

**Test21 - testFindBooksByAuthorTitleIsNull:**

Description:

The “testFindBooksByAuthorTitleIsNull” test checks the “findBooksByAuthorTitle(String author, String title)” method functions correctly.

Pre-conditions:

The author must be empty.

The title must be null.

Post-conditions:

Returns empty list.

Data required:

findBooksByAuthorTitle(String author, String title)