Test Driven Development

More JUnit Tests for the DVD app

Produced by: Mairead Meagher
Dr. Siobhán Drohan

Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

Department of Computing and Mathematics
http://www.wit.ie/

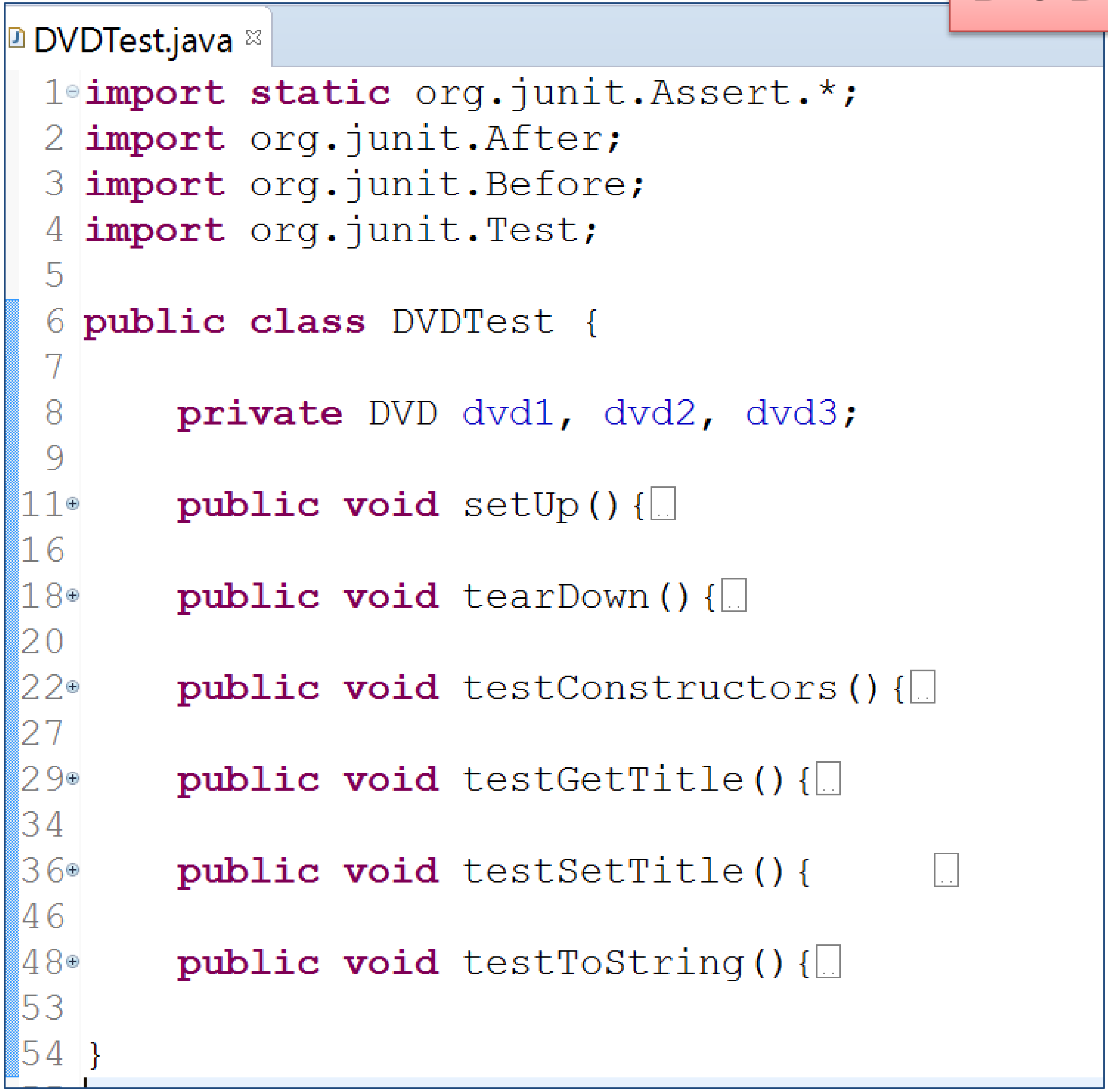
Topic List

- DVD and DVDTest.java
- JUnit Testing of Library.java (which includes testing of XML reading/writing)
- Testing Driver.java

```java
public class DVD
{
    private String title;

    public DVD(String title){
        setTitle(title);
    }

    public void setTitle(String title) {
        if (title.length() <= 20){
            this.title = title;
        }
        else{
          this.title = title.substring(0,20);
         }
    }

    public String getTitle() {
      return title;
    }

    public String toString() {
        return "DVD Title is: " + title;
    }
}
```

DVD.java

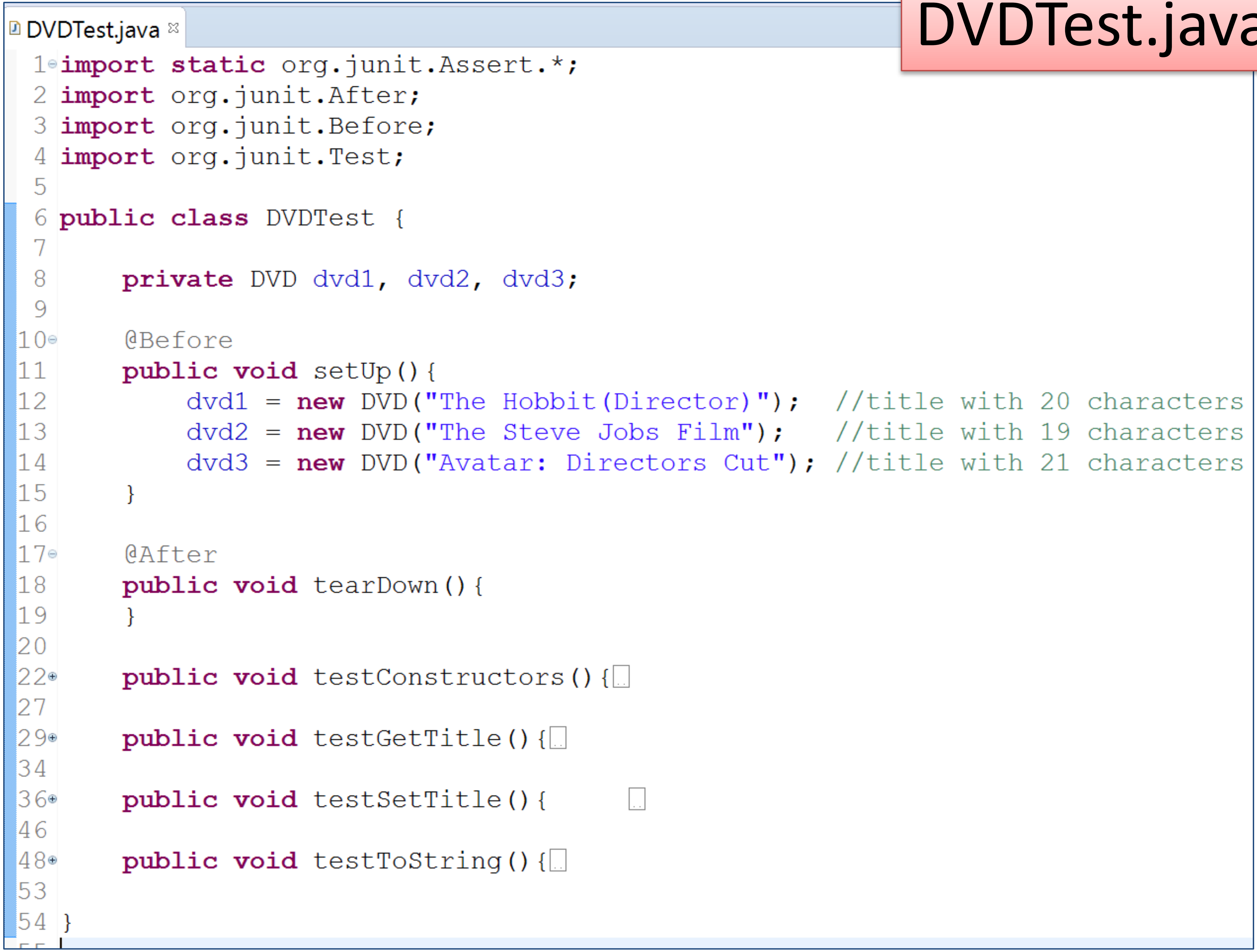
DVDTest.java



DVDTest.java

```java
import static org.junit.Assert.*;
import org.junit.After;
import org.junit.Before;
import org.junit.Test;

public class DVDTest {

    private DVD dvd1, dvd2, dvd3;

    public void setUp(){

    public void tearDown(){

    public void testConstructors(){

    public void testGetTitle(){

    public void testSetTitle(){

    public void testToString(){

}
```

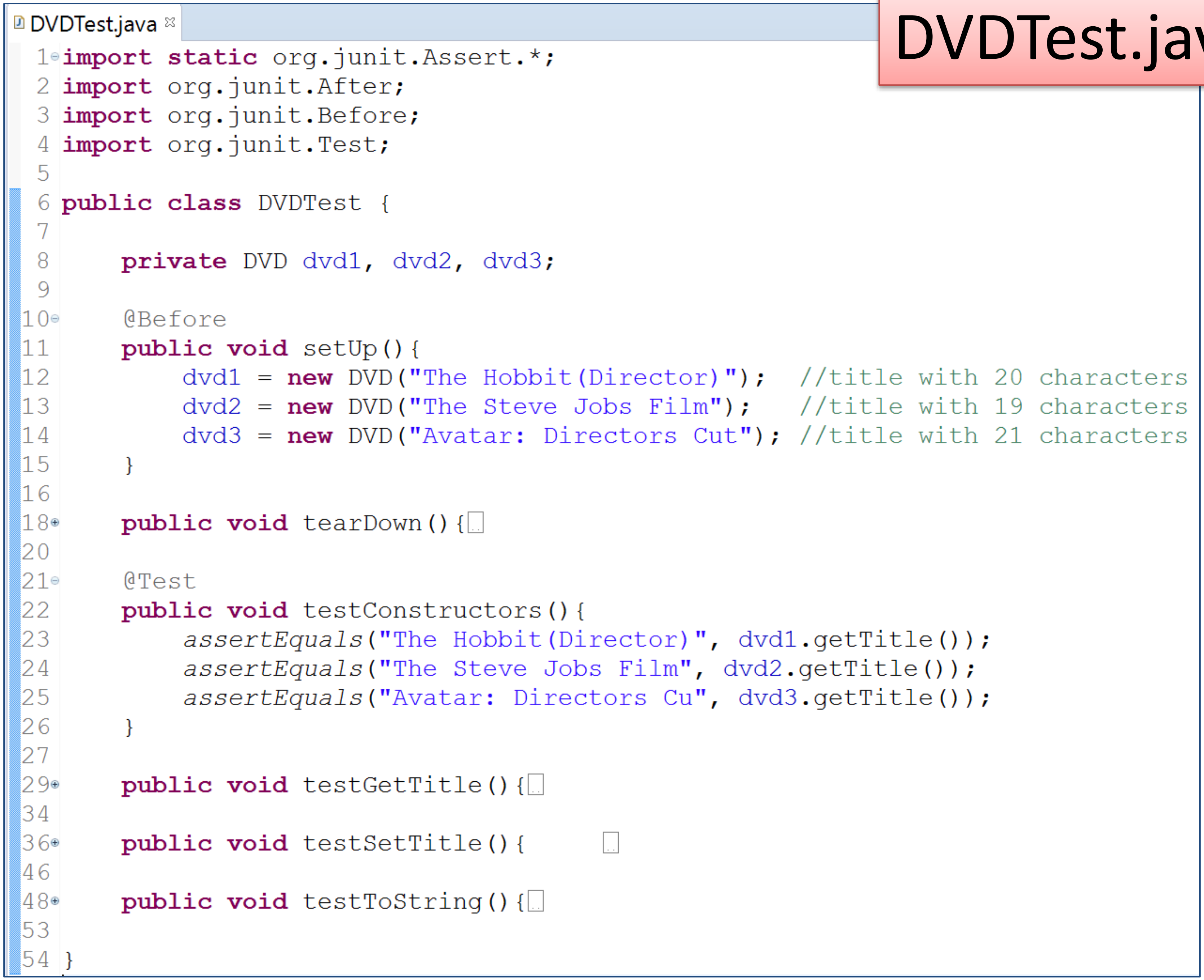
DVDTest.java



DVDTest.java

```java
import static org.junit.Assert.*;
import org.junit.After;
import org.junit.Before;
import org.junit.Test;

public class DVDTest {

    private DVD dvd1, dvd2, dvd3;

    @Before
    public void setUp(){
        dvd1 = new DVD("The Hobbit(Director)");  //title with 20 characters
        dvd2 = new DVD("The Steve Jobs Film");   //title with 19 characters
        dvd3 = new DVD("Avatar: Directors Cut"); //title with 21 characters
    }

    @After
    public void tearDown(){
    }

    public void testConstructors(){

    public void testGetTitle(){

    public void testSetTitle(){

    public void testToString(){

}
```

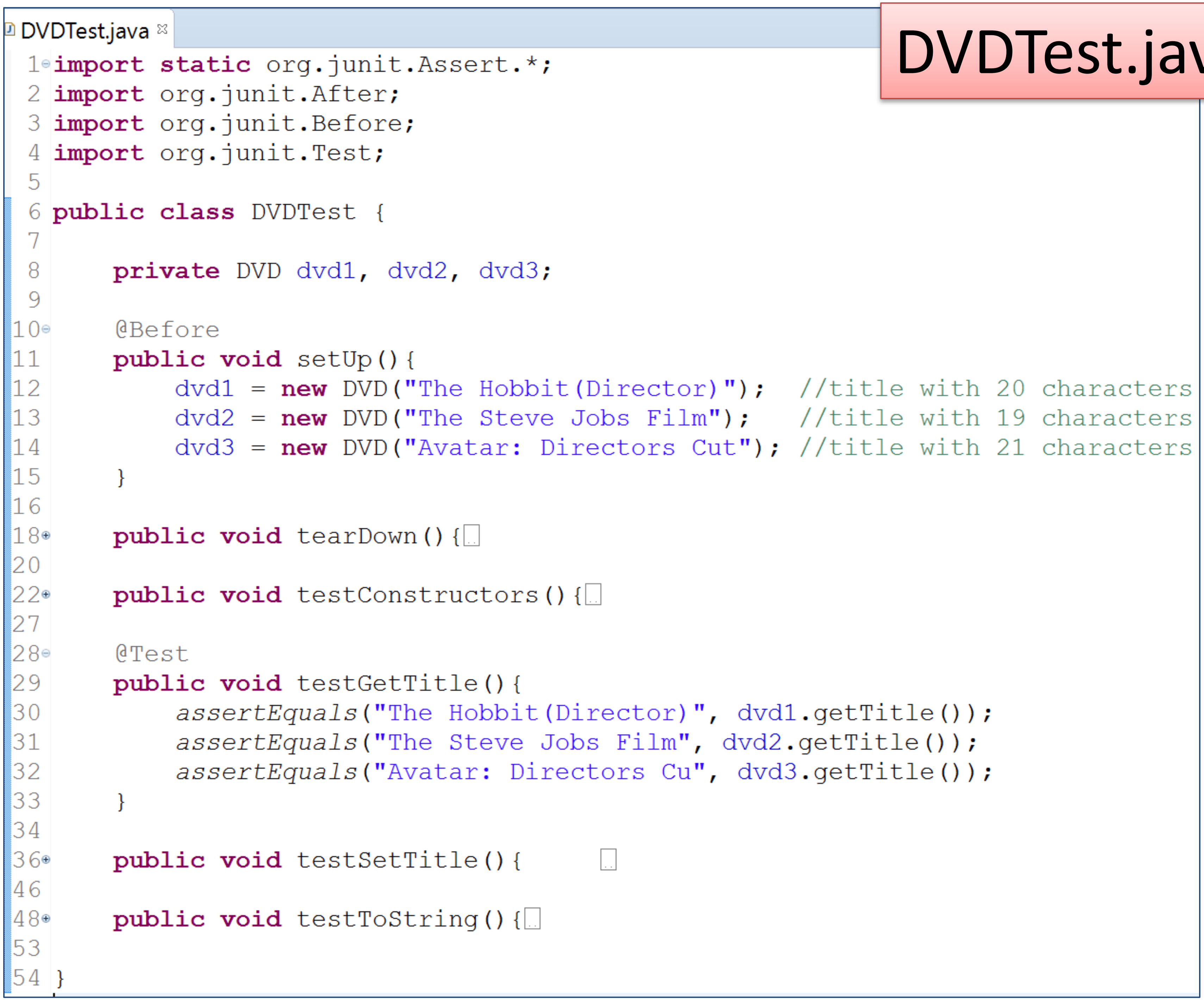
DVDTest.java



DVDTest.java

```java
import static org.junit.Assert.*;
import org.junit.After;
import org.junit.Before;
import org.junit.Test;

public class DVDTest {

    private DVD dvd1, dvd2, dvd3;

    @Before
    public void setUp(){
        dvd1 = new DVD("The Hobbit(Director)");  //title with 20 characters
        dvd2 = new DVD("The Steve Jobs Film");   //title with 19 characters
        dvd3 = new DVD("Avatar: Directors Cut"); //title with 21 characters
    }

    public void tearDown(){

    @Test
    public void testConstructors(){
        assertEquals("The Hobbit(Director)", dvd1.getTitle());
        assertEquals("The Steve Jobs Film", dvd2.getTitle());
        assertEquals("Avatar: Directors Cu", dvd3.getTitle());
    }

    public void testGetTitle(){

    public void testSetTitle(){

    public void testToString(){

}
```

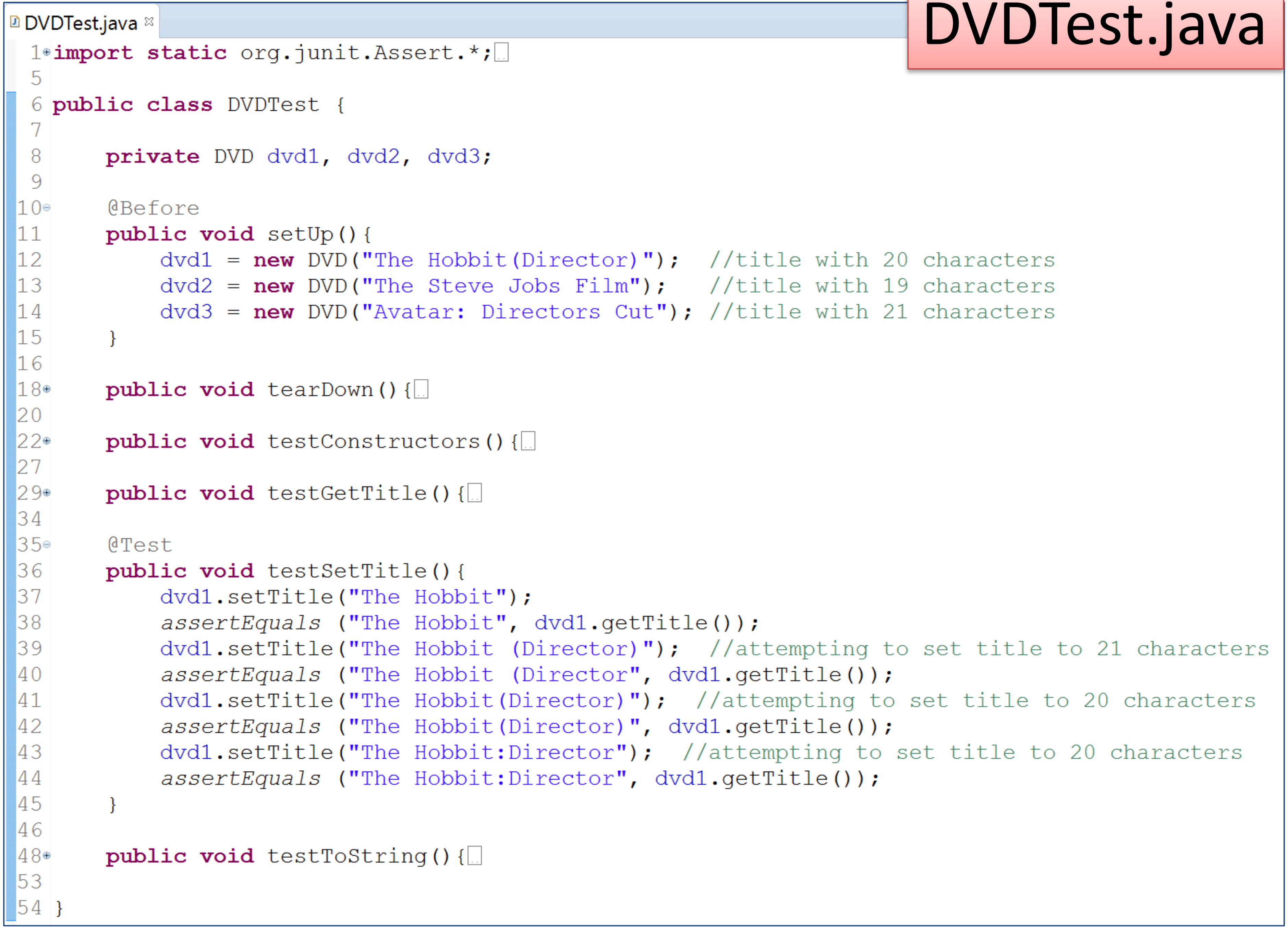
DVDTest.java



DVDTest.java

```java
import static org.junit.Assert.*;
import org.junit.After;
import org.junit.Before;
import org.junit.Test;

public class DVDTest {

    private DVD dvd1, dvd2, dvd3;

    @Before
    public void setUp(){
        dvd1 = new DVD("The Hobbit(Director)");  //title with 20 characters
        dvd2 = new DVD("The Steve Jobs Film");   //title with 19 characters
        dvd3 = new DVD("Avatar: Directors Cut"); //title with 21 characters
    }

    public void tearDown(){

    public void testConstructors(){

    @Test
    public void testGetTitle(){
        assertEquals("The Hobbit(Director)", dvd1.getTitle());
        assertEquals("The Steve Jobs Film", dvd2.getTitle());
        assertEquals("Avatar: Directors Cu", dvd3.getTitle());
    }

    public void testSetTitle(){

    public void testToString(){

}
```

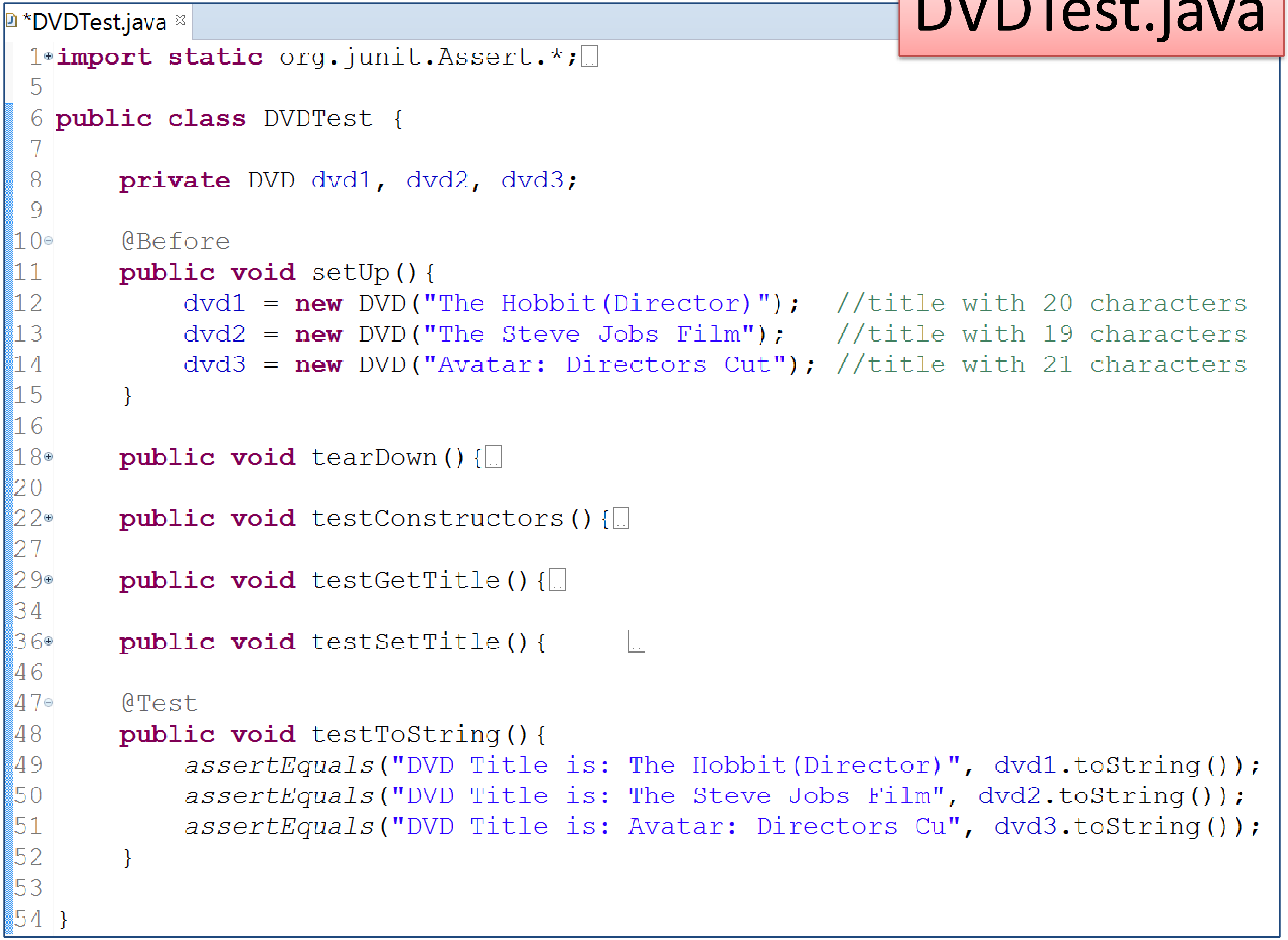
DVDTest.java



DVDTest.java

```java
import static org.junit.Assert.*;

public class DVDTest {

    private DVD dvd1, dvd2, dvd3;

    @Before
    public void setUp(){
        dvd1 = new DVD("The Hobbit(Director)");  //title with 20 characters
        dvd2 = new DVD("The Steve Jobs Film");   //title with 19 characters
        dvd3 = new DVD("Avatar: Directors Cut"); //title with 21 characters
    }

    public void tearDown(){

    public void testConstructors(){

    public void testGetTitle(){

    @Test
    public void testSetTitle(){
        dvd1.setTitle("The Hobbit");
        assertEquals ("The Hobbit", dvd1.getTitle());
        dvd1.setTitle("The Hobbit (Director)");  //attempting to set title to 21 characters
        assertEquals ("The Hobbit (Director", dvd1.getTitle());
        dvd1.setTitle("The Hobbit(Director)");  //attempting to set title to 20 characters
        assertEquals ("The Hobbit(Director)", dvd1.getTitle());
        dvd1.setTitle("The Hobbit:Director");  //attempting to set title to 20 characters
        assertEquals ("The Hobbit:Director", dvd1.getTitle());
    }

    public void testToString(){

}
```

DVDTest.java



*DVDTest.java

```
import static org.junit.Assert.*;

public class DVDTest {

    private DVD dvd1, dvd2, dvd3;

    @Before
    public void setUp(){
        dvd1 = new DVD("The Hobbit(Director)");   //title with 20 characters
        dvd2 = new DVD("The Steve Jobs Film");    //title with 19 characters
        dvd3 = new DVD("Avatar: Directors Cut"); //title with 21 characters
    }

    public void tearDown(){

    public void testConstructors(){

    public void testGetTitle(){

    public void testSetTitle(){

    @Test
    public void testToString(){
        assertEquals("DVD Title is: The Hobbit(Director)", dvd1.toString());
        assertEquals("DVD Title is: The Steve Jobs Film", dvd2.toString());
        assertEquals("DVD Title is: Avatar: Directors Cu", dvd3.toString());
    }

}
```

Topic List

- DVD and DVDTest.java

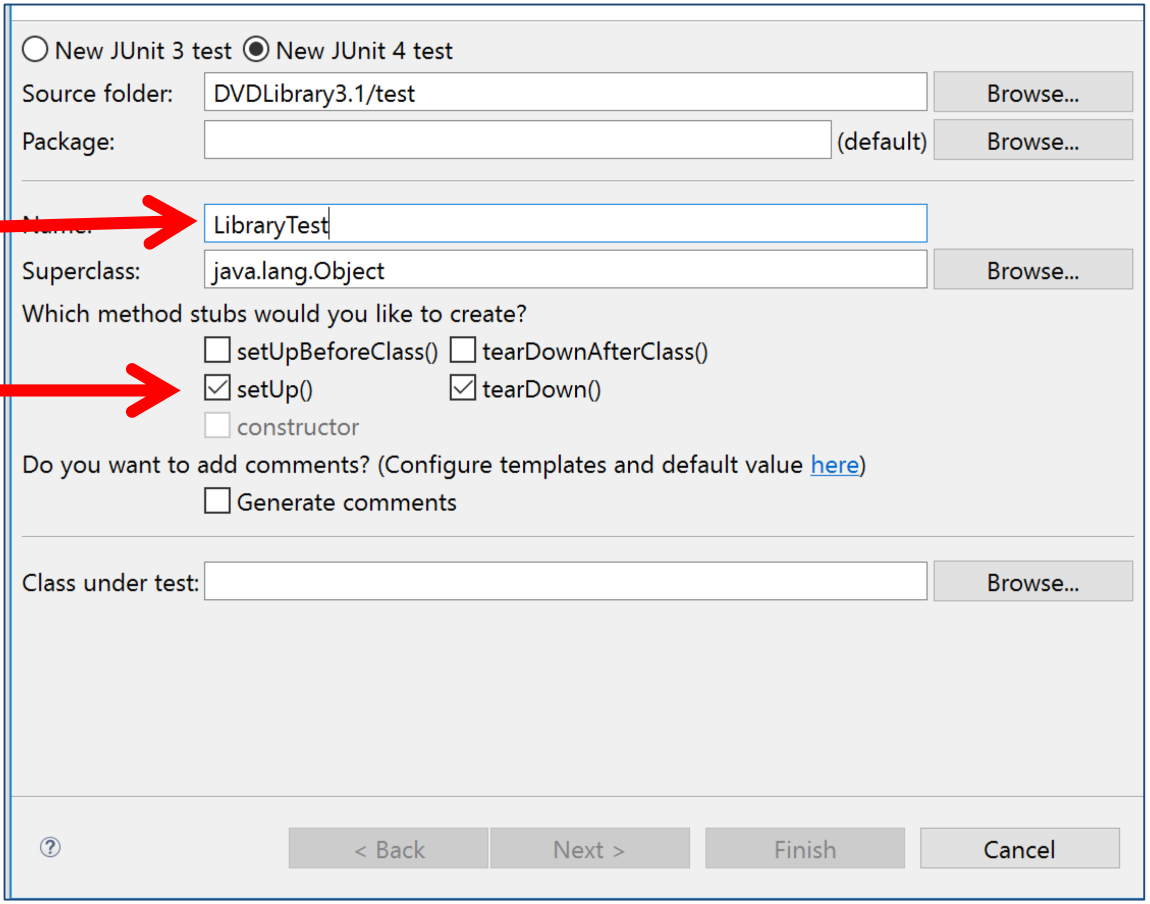
- JUnit Testing of Library.java (which includes testing of XML reading/writing)

- Testing Driver.java

Create a new JUnit Test Case within the “test” folder

Call the test class, **LibraryTest**

Generate the default setUp() and tearDown() methods.



( ) New JUnit 3 test (●) New JUnit 4 test

Source folder: DVDLibrary3.1/test

Package: (default)

Name: LibraryTest

Superclass: java.lang.Object

Which method stubs would you like to create?

☐ setUpBeforeClass() ☐ tearDownAfterClass()

☑ setUp() ☑ tearDown()

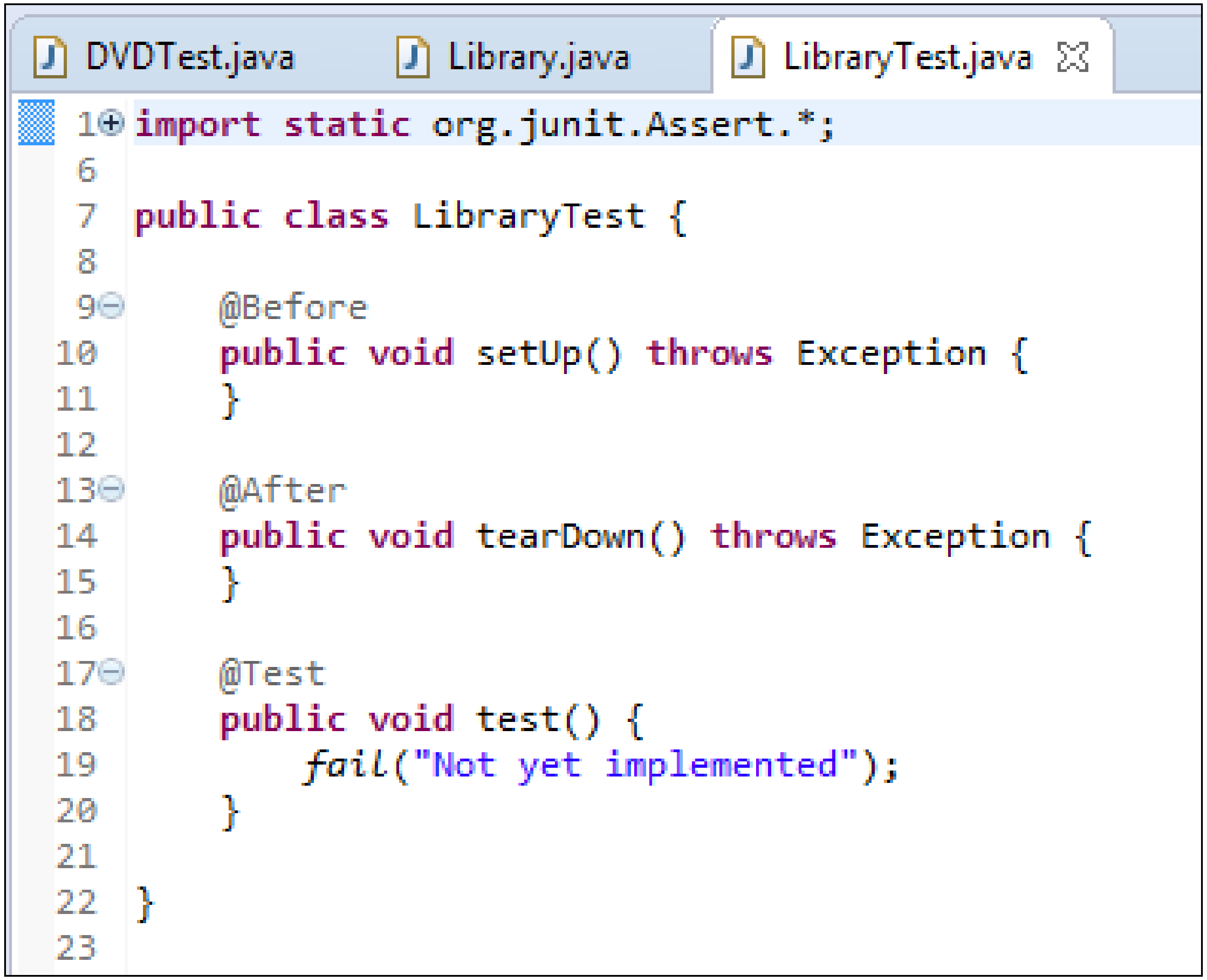
☐ constructor

Do you want to add comments? (Configure templates and default value here)

☐ Generate comments

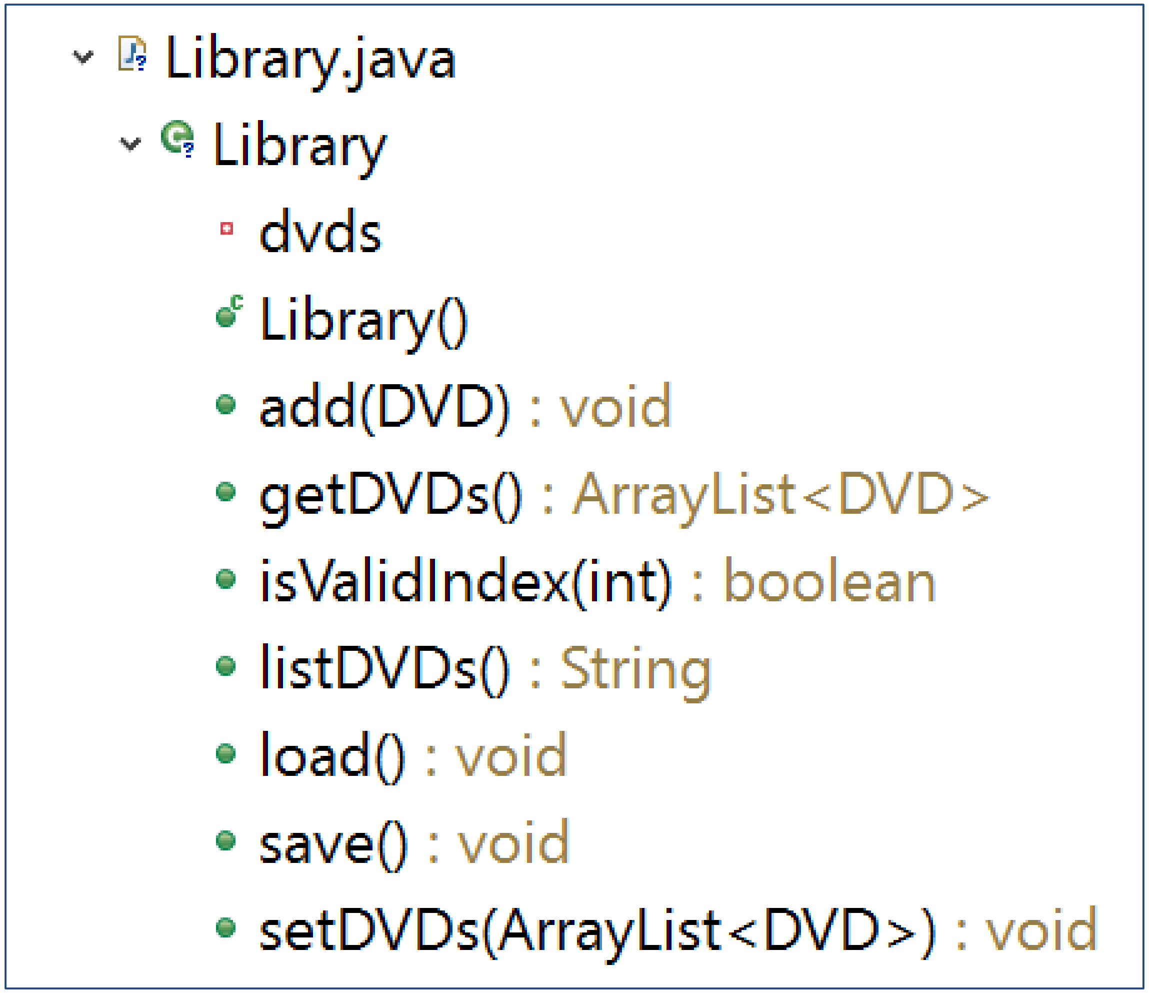
Class under test:

Generated LibraryTest.java



```java
import static org.junit.Assert.*;

public class LibraryTest {

    @Before
    public void setUp() throws Exception {
    }

    @After
    public void tearDown() throws Exception {
    }

    @Test
    public void test() {
        fail("Not yet implemented");
    }

}

```

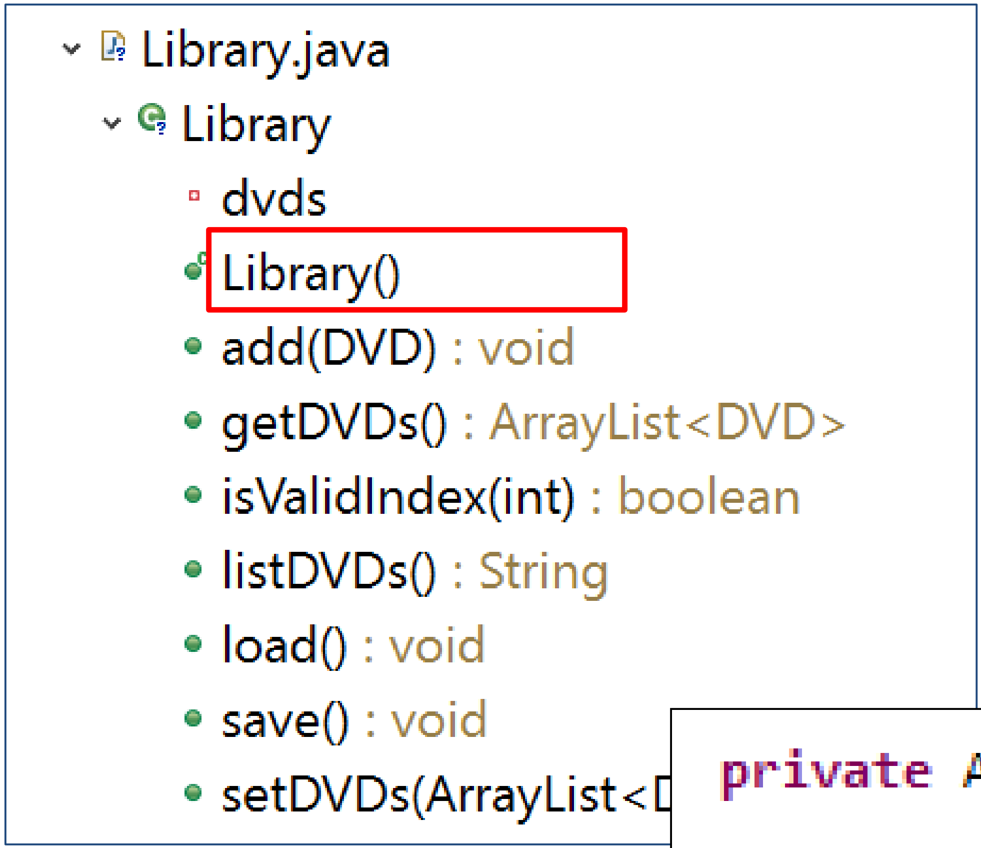
Library.java



```
Library.java
  Library
    dvds
    Library()
    add(DVD) : void
    getDVDs() : ArrayList<DVD>
    isValidIndex(int) : boolean
    listDVDs() : String
    load() : void
    save() : void
    setDVDs(ArrayList<DVD>) : void
```

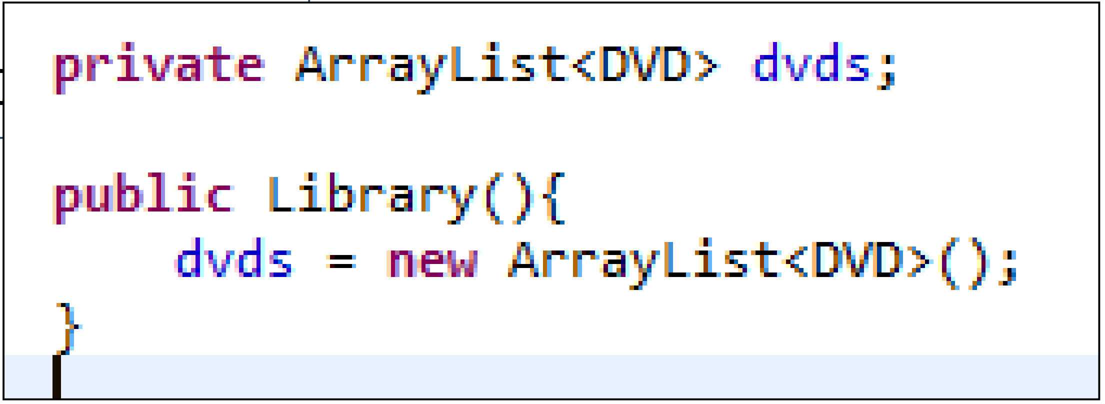
We need to write at least one test for each of these methods.

Library.java – testing the constructor



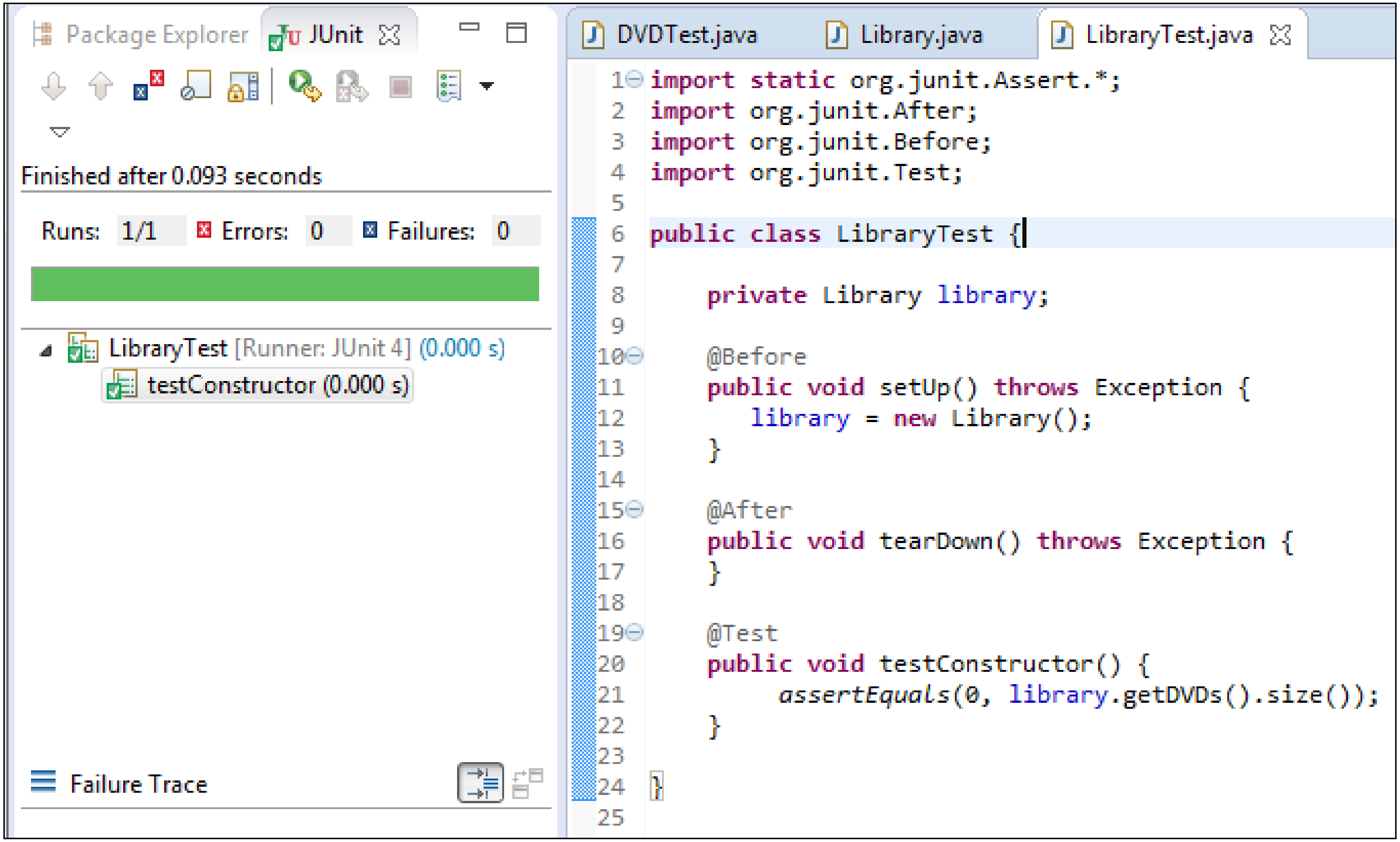
```
Library.java
  Library
    dvds
    Library()
    add(DVD) : void
    getDVDs() : ArrayList<DVD>
    isValidIndex(int) : boolean
    listDVDs() : String
    load() : void
    save() : void
    setDVDs(ArrayList<D
```

To test the constructor, we will create a Library object. Then we will call the getDVDs() method and ensure that the returned ArrayList has a size of 0.



```java
private ArrayList<DVD> dvds;

public Library(){
    dvds = new ArrayList<DVD>();
}
```

Library.java – testing the constructor



Package Explorer | JUnit

Finished after 0.093 seconds

Runs: 1/1 | Errors: 0 | Failures: 0

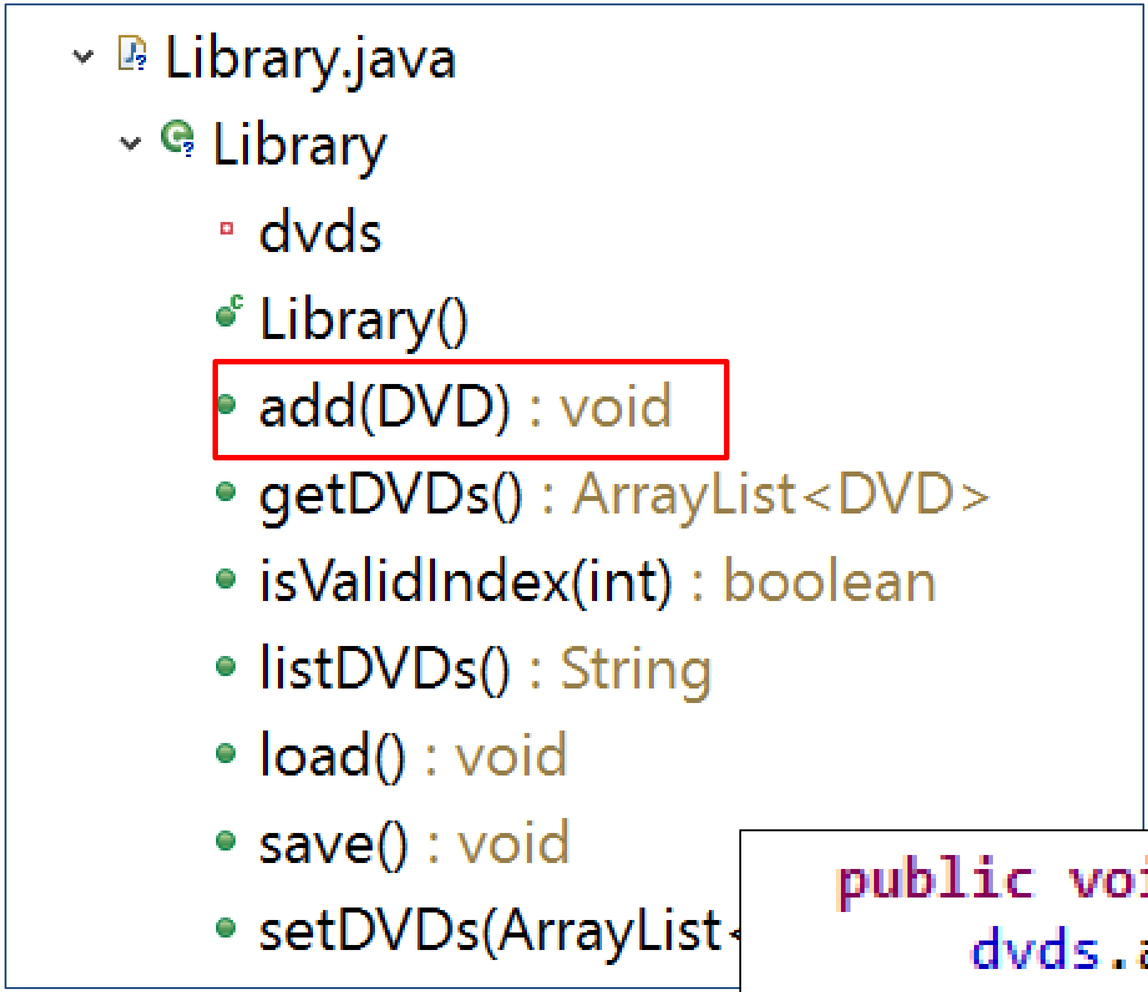
LibraryTest [Runner: JUnit 4] (0.000 s)
testConstructor (0.000 s)

Failure Trace

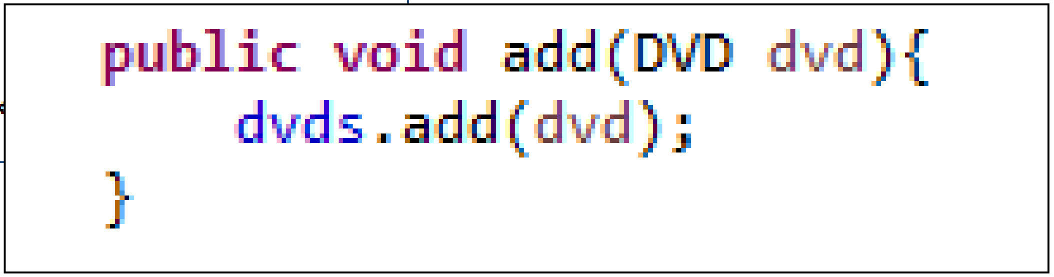
DVDTest.java | Library.java | LibraryTest.java

```java
import static org.junit.Assert.*;
import org.junit.After;
import org.junit.Before;
import org.junit.Test;

public class LibraryTest {

    private Library library;

    @Before
    public void setUp() throws Exception {
        library = new Library();
    }

    @After
    public void tearDown() throws Exception {
    }

    @Test
    public void testConstructor() {
        assertEquals(0, library.getDVDs().size());
    }

}
```

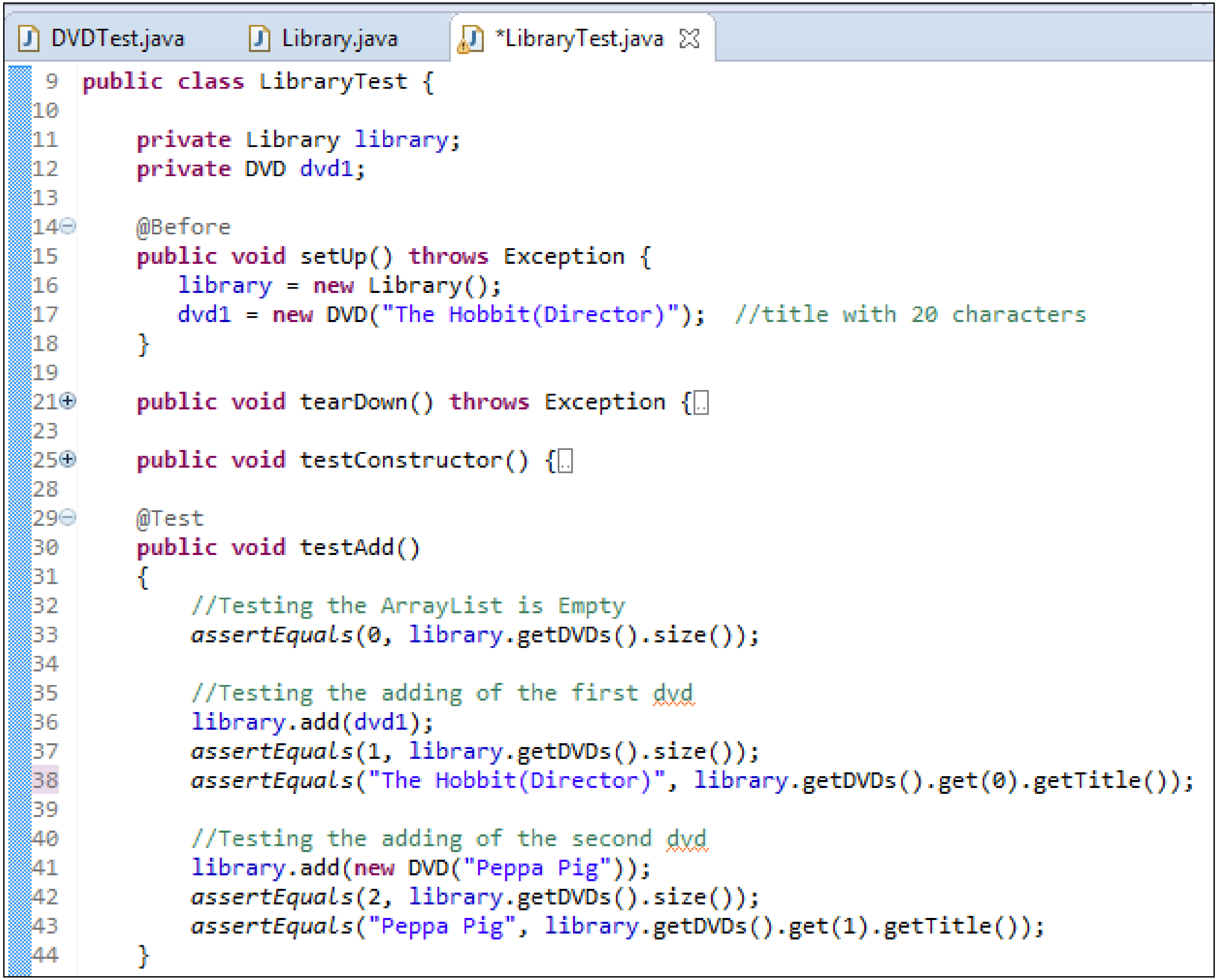
Library.java – testing add(DVD)



- Library.java
  - Library
    - dvds
    - Library()
    - add(DVD) : void
    - getDVDs() : ArrayList<DVD>
    - isValidIndex(int) : boolean
    - listDVDs() : String
    - load() : void
    - save() : void
    - setDVDs(ArrayList<

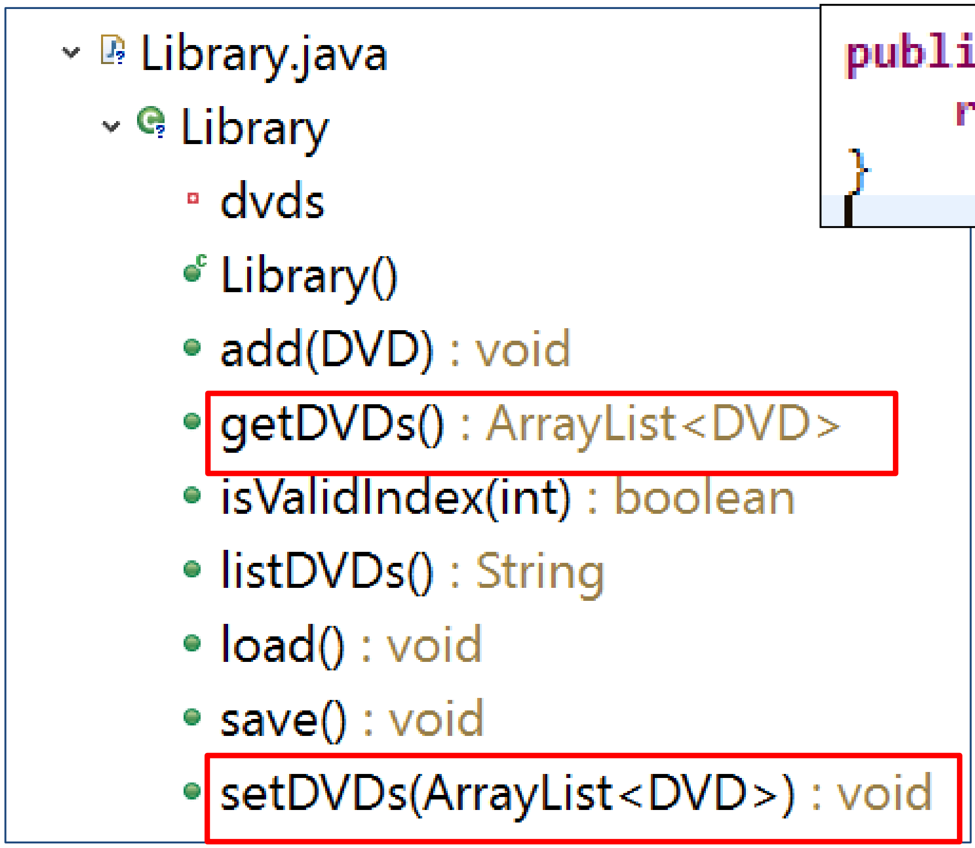


```
public void add(DVD dvd){
    dvds.add(dvd);
}
```

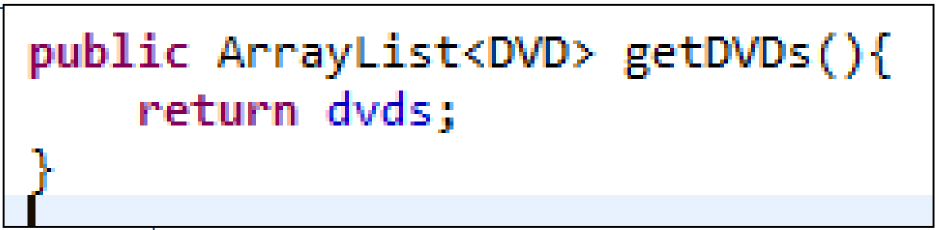


```
public class LibraryTest {

    private Library library;
    private DVD dvd1;

    @Before
    public void setUp() throws Exception {
        library = new Library();
        dvd1 = new DVD("The Hobbit(Director)");  //title with 20 characters
    }

    public void tearDown() throws Exception {

    public void testConstructor() {

    @Test
    public void testAdd()
    {
        //Testing the ArrayList is Empty
        assertEquals(0, library.getDVDs().size());

        //Testing the adding of the first dvd
        library.add(dvd1);
        assertEquals(1, library.getDVDs().size());
        assertEquals("The Hobbit(Director)", library.getDVDs().get(0).getTitle());

        //Testing the adding of the second dvd
        library.add(new DVD("Peppa Pig"));
        assertEquals(2, library.getDVDs().size());
        assertEquals("Peppa Pig", library.getDVDs().get(1).getTitle());
    }
```

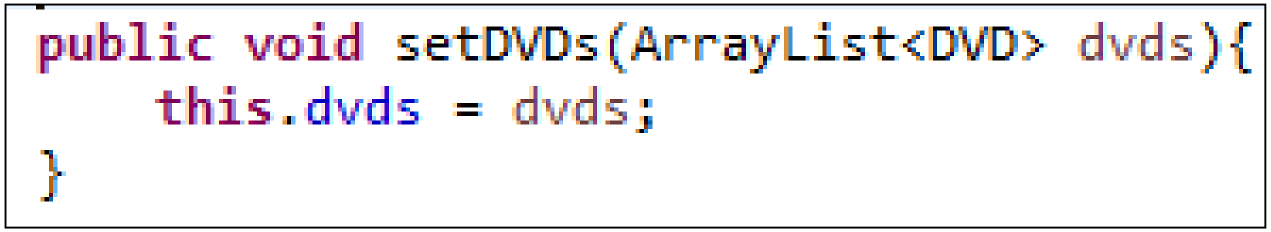
Library.java – testing getDVDs and setDVDs



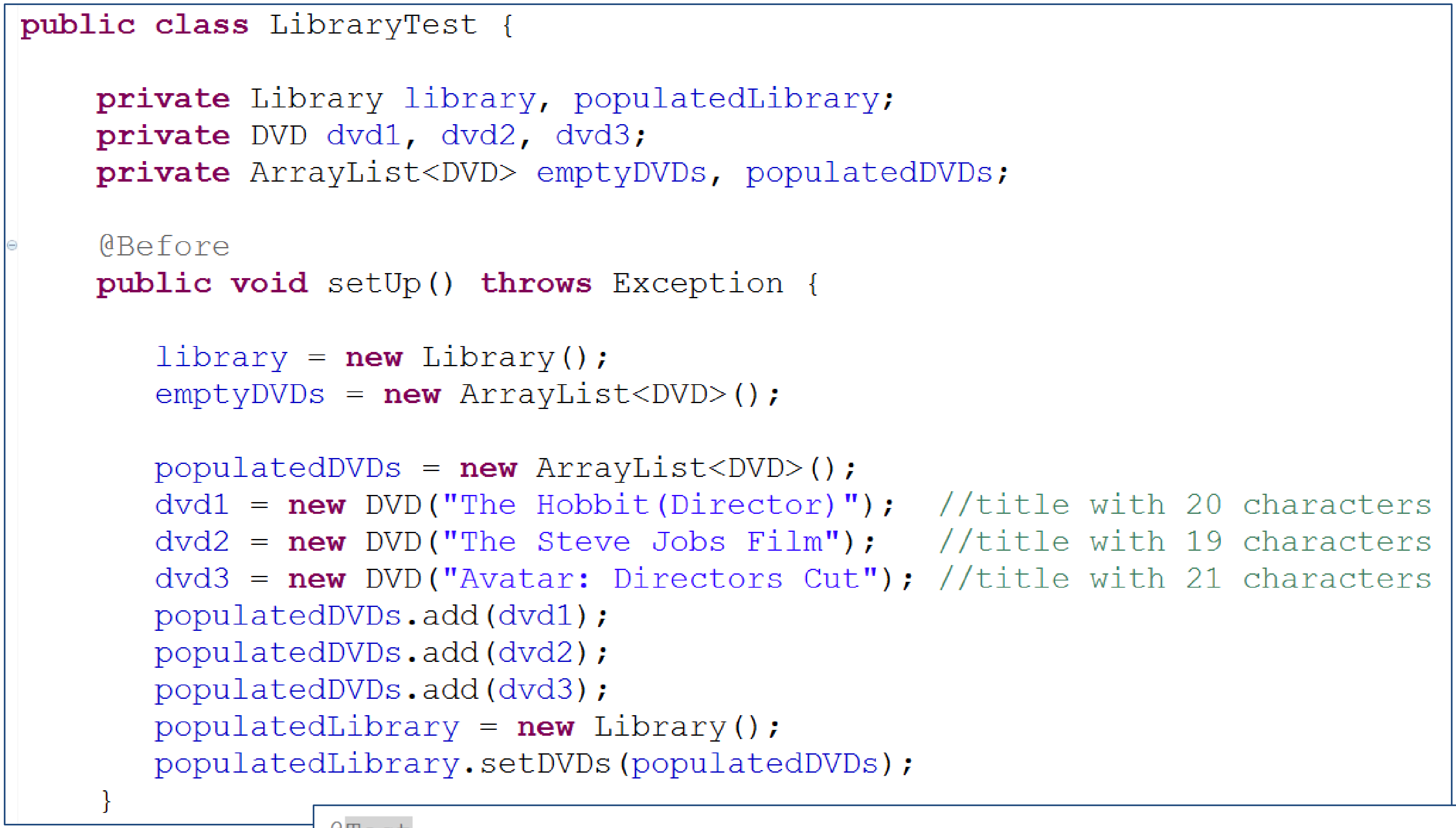
```
Library.java
  Library
    dvds
    Library()
    add(DVD) : void
    getDVDs() : ArrayList<DVD>
    isValidIndex(int) : boolean
    listDVDs() : String
    load() : void
    save() : void
    setDVDs(ArrayList<DVD>) : void
```



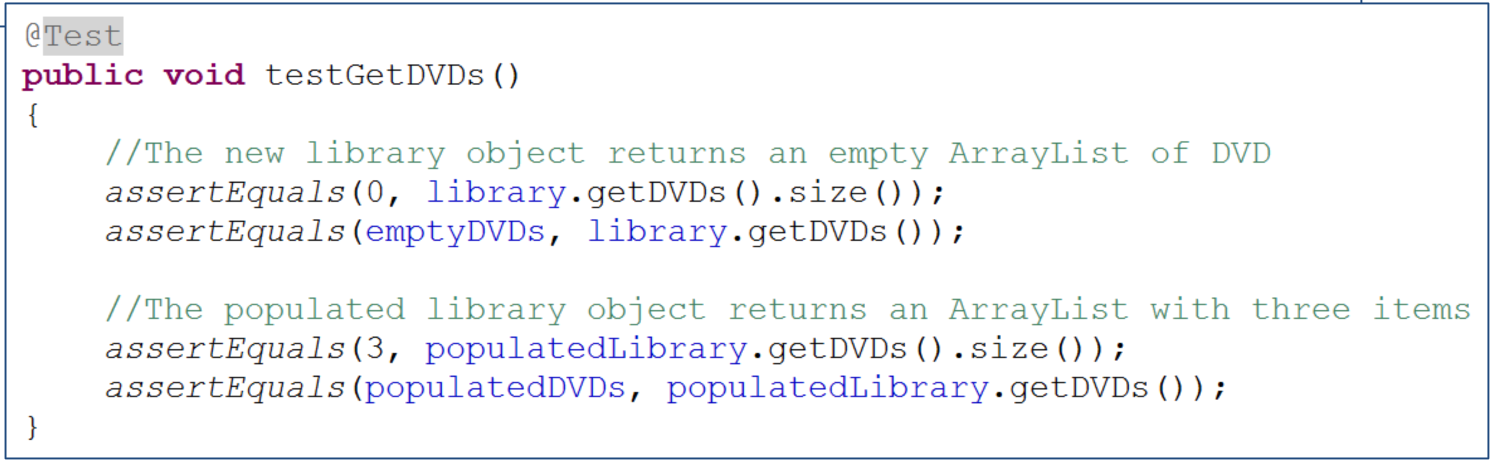
```java
public ArrayList<DVD> getDVDs(){
    return dvds;
}
```



```java
public void setDVDs(ArrayList<DVD> dvds){
    this.dvds = dvds;
}
```

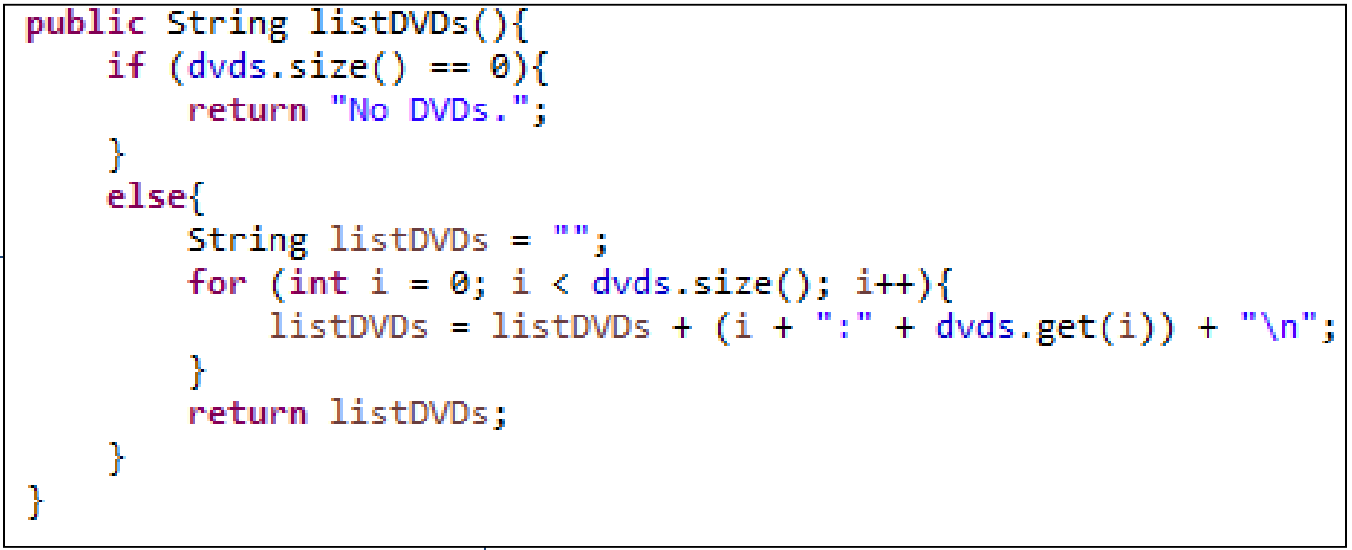


```java
public class LibraryTest {

    private Library library, populatedLibrary;
    private DVD dvd1, dvd2, dvd3;
    private ArrayList<DVD> emptyDVDs, populatedDVDs;

    @Before
    public void setUp() throws Exception {

        library = new Library();
        emptyDVDs = new ArrayList<DVD>();

        populatedDVDs = new ArrayList<DVD>();
        dvd1 = new DVD("The Hobbit(Director)");  //title with 20 characters
        dvd2 = new DVD("The Steve Jobs Film");   //title with 19 characters
        dvd3 = new DVD("Avatar: Directors Cut"); //title with 21 characters
        populatedDVDs.add(dvd1);
        populatedDVDs.add(dvd2);
        populatedDVDs.add(dvd3);
        populatedLibrary = new Library();
        populatedLibrary.setDVDs(populatedDVDs);
    }
```

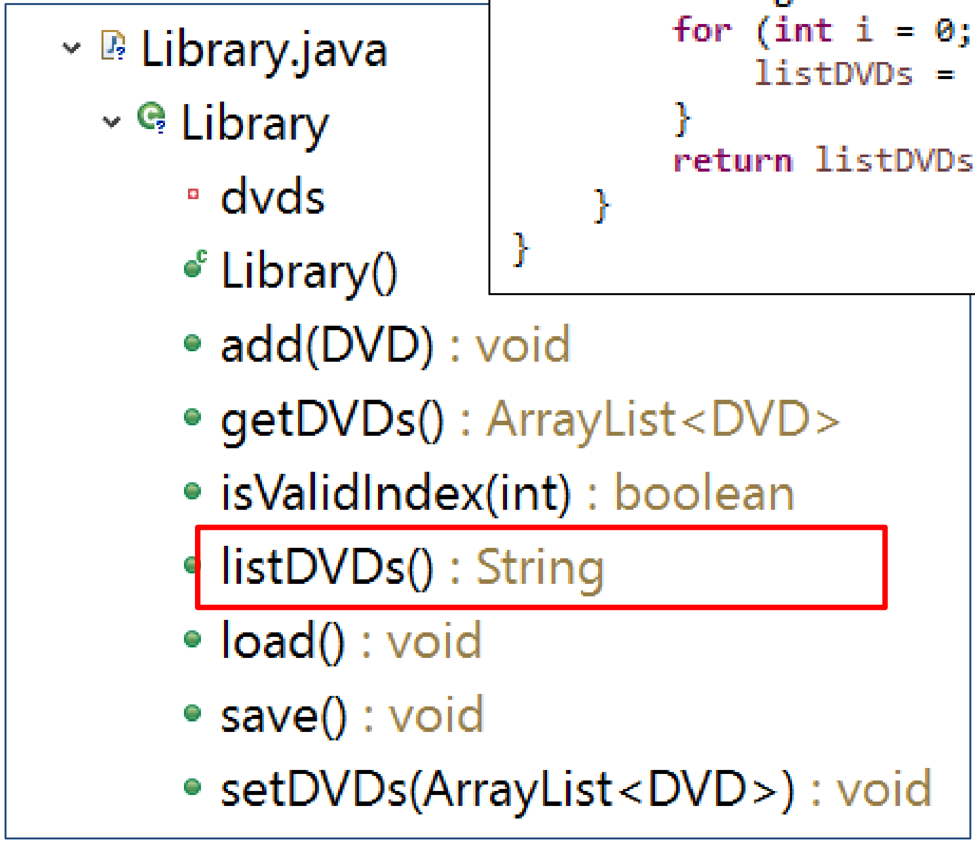


```java
@Test
public void testGetDVDs()
{
    //The new library object returns an empty ArrayList of DVD
    assertEquals(0, library.getDVDs().size());
    assertEquals(emptyDVDs, library.getDVDs());

    //The populated library object returns an ArrayList with three items
    assertEquals(3, populatedLibrary.getDVDs().size());
    assertEquals(populatedDVDs, populatedLibrary.getDVDs());
}
```

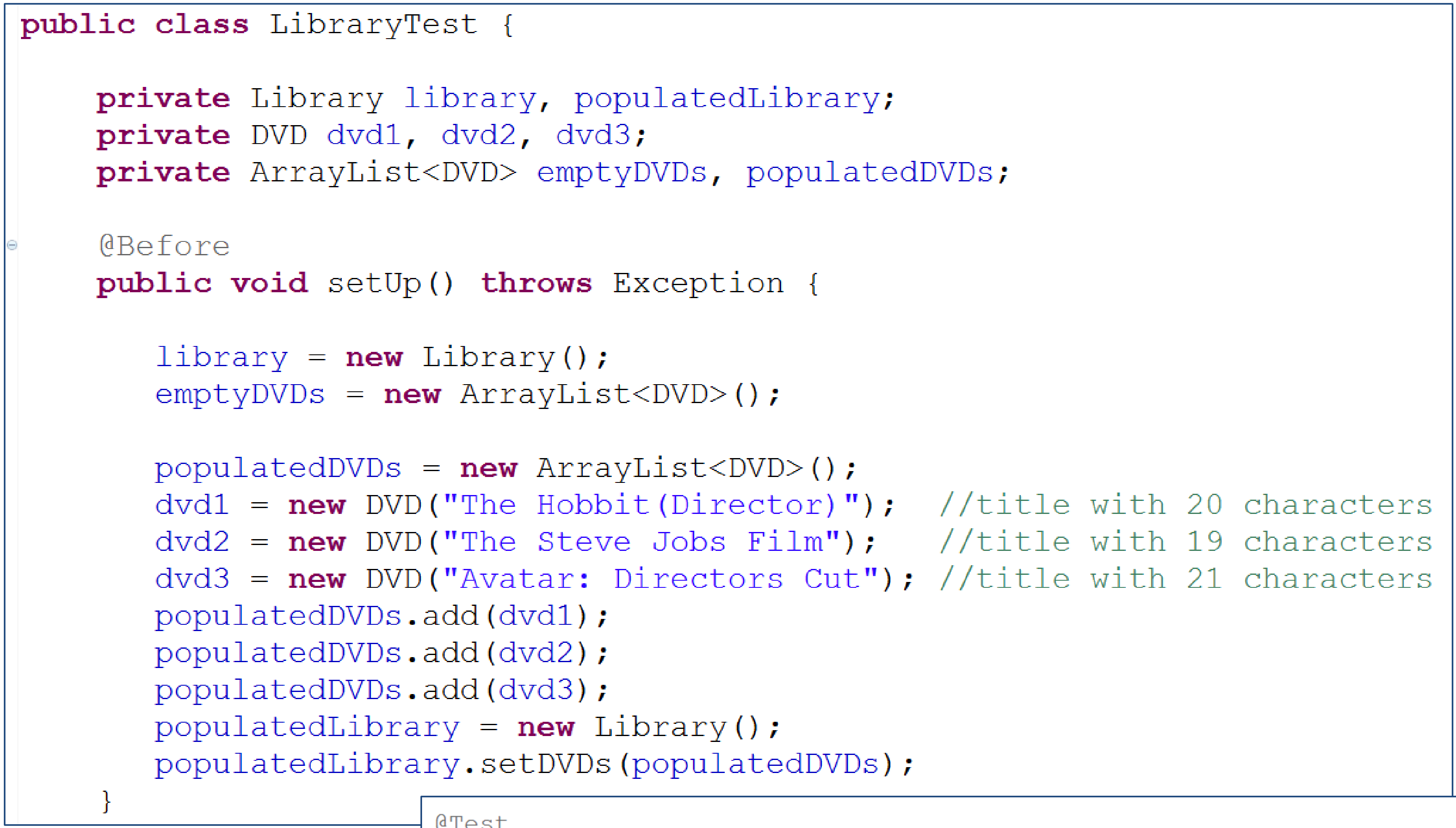
Library.java – testing listDVDs()



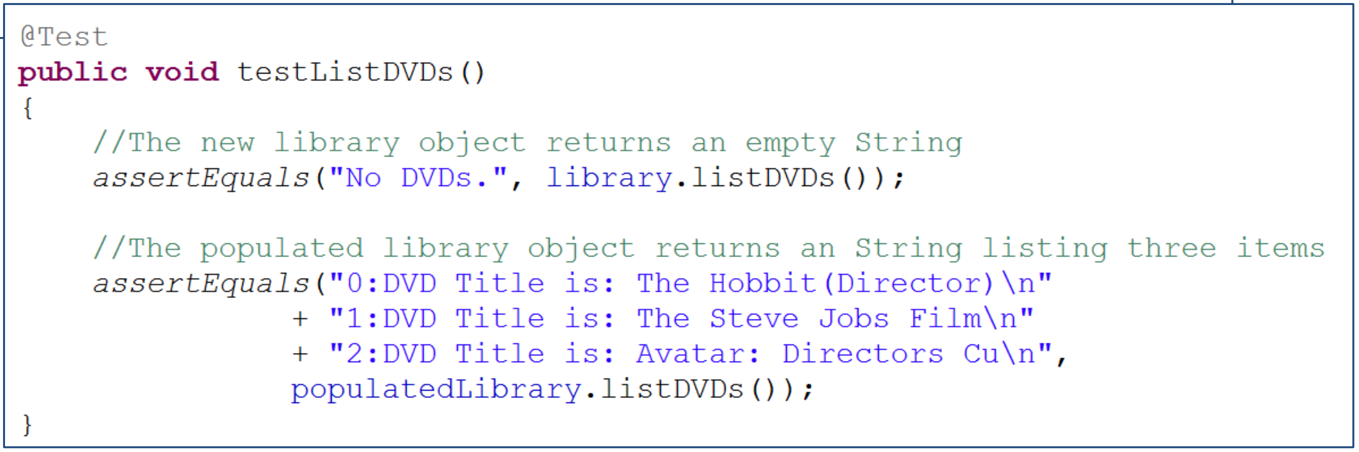
```java
public String listDVDs(){
    if (dvds.size() == 0){
        return "No DVDs.";
    }
    else{
        String listDVDs = "";
        for (int i = 0; i < dvds.size(); i++){
            listDVDs = listDVDs + (i + ":" + dvds.get(i)) + "\n";
        }
        return listDVDs;
    }
}
```



```
Library.java
  Library
    dvds
    Library()
    add(DVD) : void
    getDVDs() : ArrayList<DVD>
    isValidIndex(int) : boolean
    listDVDs() : String
    load() : void
    save() : void
    setDVDs(ArrayList<DVD>) : void
```

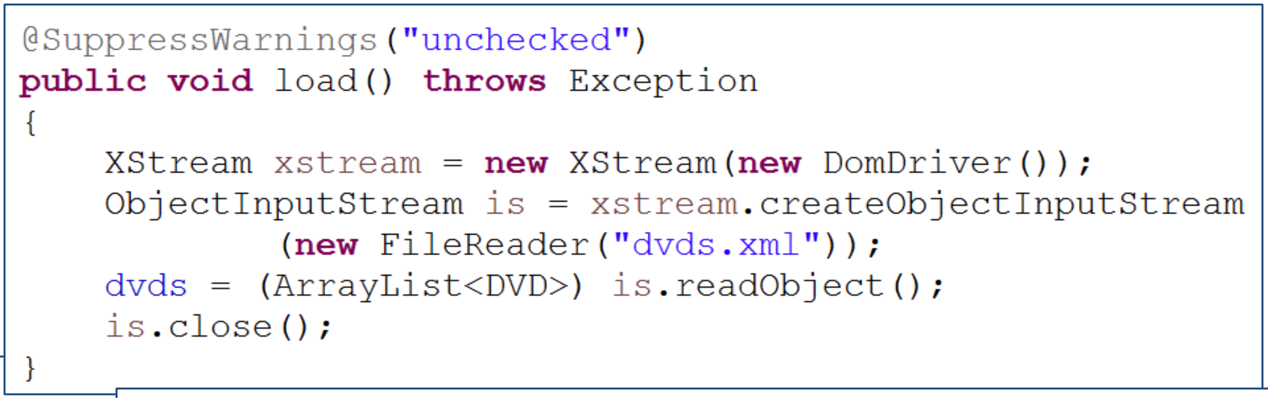


```java
public class LibraryTest {

    private Library library, populatedLibrary;
    private DVD dvd1, dvd2, dvd3;
    private ArrayList<DVD> emptyDVDs, populatedDVDs;

    @Before
    public void setUp() throws Exception {

        library = new Library();
        emptyDVDs = new ArrayList<DVD>();

        populatedDVDs = new ArrayList<DVD>();
        dvd1 = new DVD("The Hobbit(Director)");  //title with 20 characters
        dvd2 = new DVD("The Steve Jobs Film");   //title with 19 characters
        dvd3 = new DVD("Avatar: Directors Cut"); //title with 21 characters
        populatedDVDs.add(dvd1);
        populatedDVDs.add(dvd2);
        populatedDVDs.add(dvd3);
        populatedLibrary = new Library();
        populatedLibrary.setDVDs(populatedDVDs);
    }
```

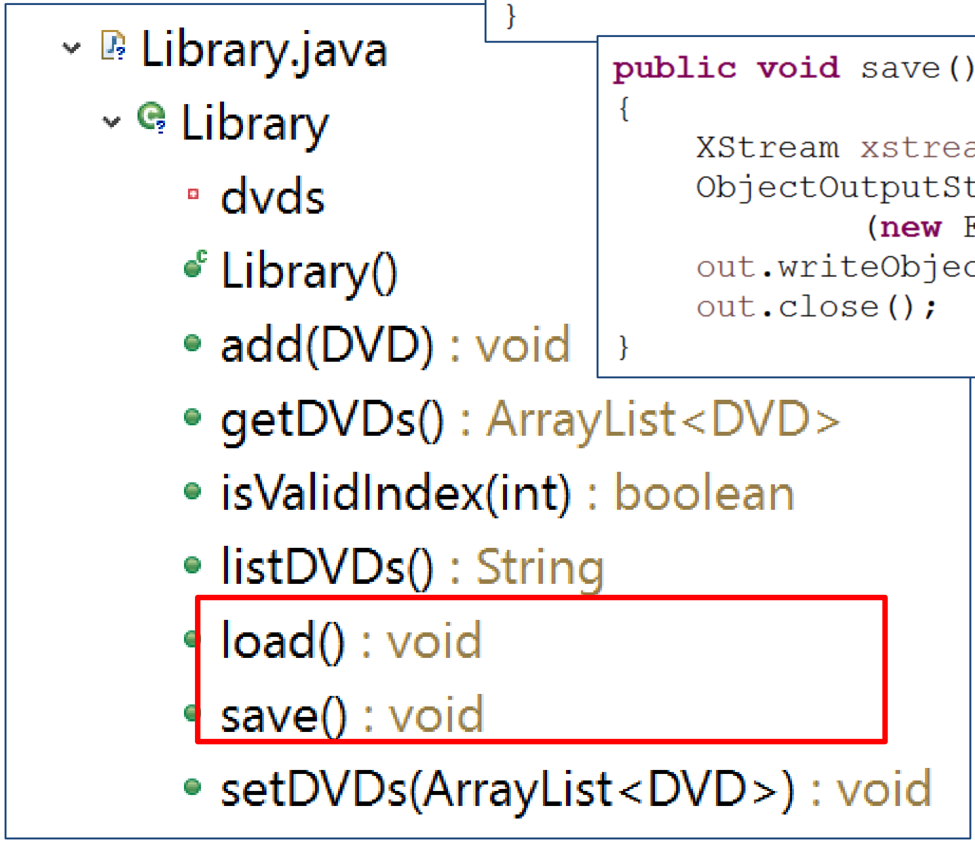


```java
@Test
public void testListDVDs()
{
    //The new library object returns an empty String
    assertEquals("No DVDs.", library.listDVDs());

    //The populated library object returns an String listing three items
    assertEquals("0:DVD Title is: The Hobbit(Director)\n"
               + "1:DVD Title is: The Steve Jobs Film\n"
               + "2:DVD Title is: Avatar: Directors Cu\n",
               populatedLibrary.listDVDs());
}
```

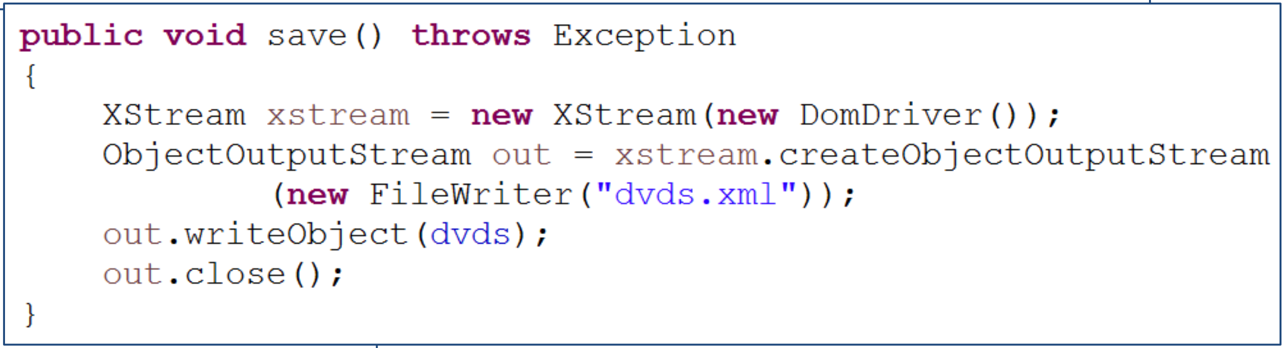
Library.java – testing save and load



```
@SuppressWarnings("unchecked")
public void load() throws Exception
{
    XStream xstream = new XStream(new DomDriver());
    ObjectInputStream is = xstream.createObjectInputStream
            (new FileReader("dvds.xml"));
    dvds = (ArrayList<DVD>) is.readObject();
    is.close();
}
```

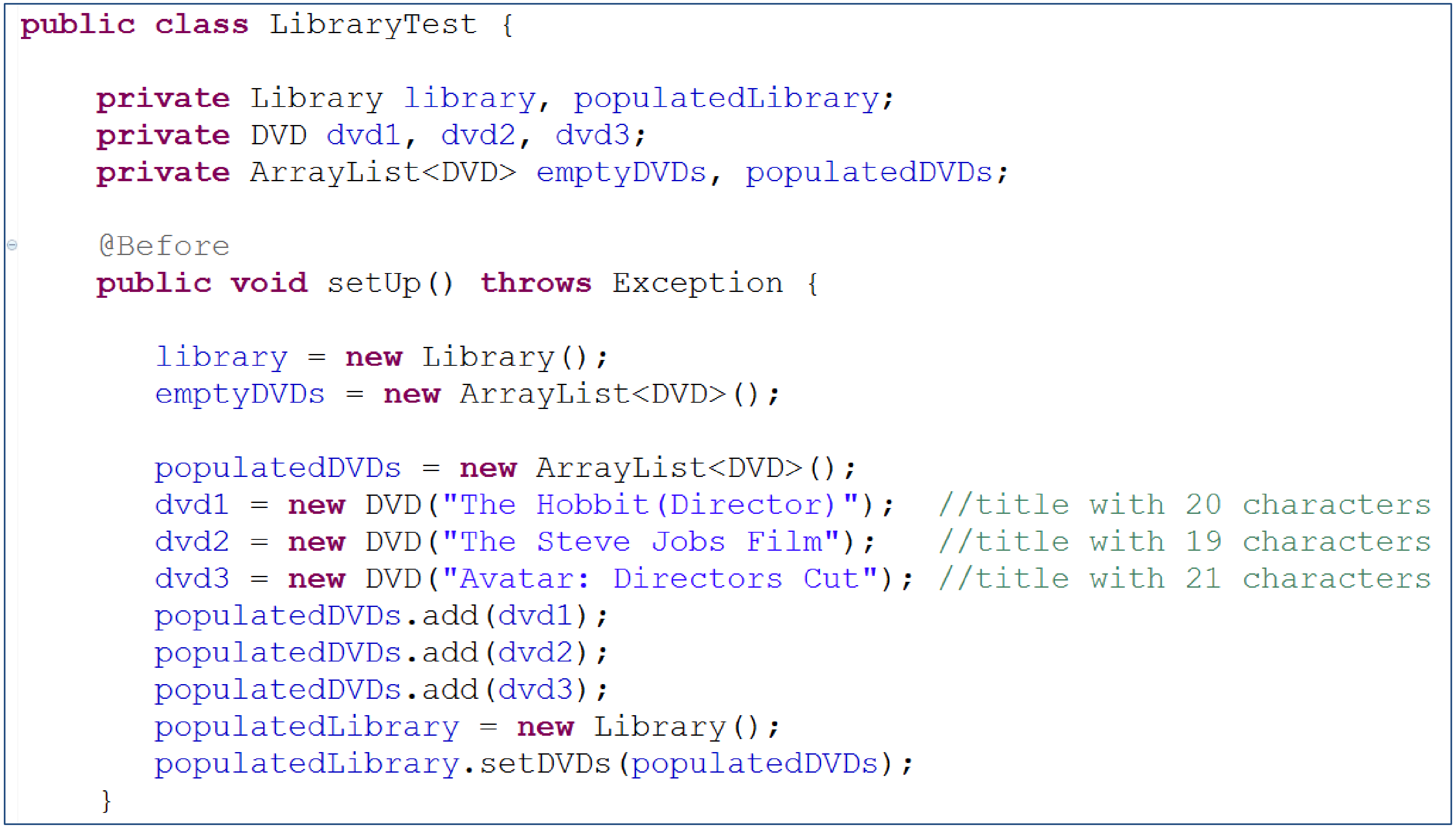


```
Library.java
  Library
    dvds
    Library()
    add(DVD) : void
    getDVDs() : ArrayList<DVD>
    isValidIndex(int) : boolean
    listDVDs() : String
    load() : void
    save() : void
    setDVDs(ArrayList<DVD>) : void
```



```
public void save() throws Exception
{
    XStream xstream = new XStream(new DomDriver());
    ObjectOutputStream out = xstream.createObjectOutputStream
            (new FileWriter("dvds.xml"));
    out.writeObject(dvds);
    out.close();
}
```

Library.java – testing save and load

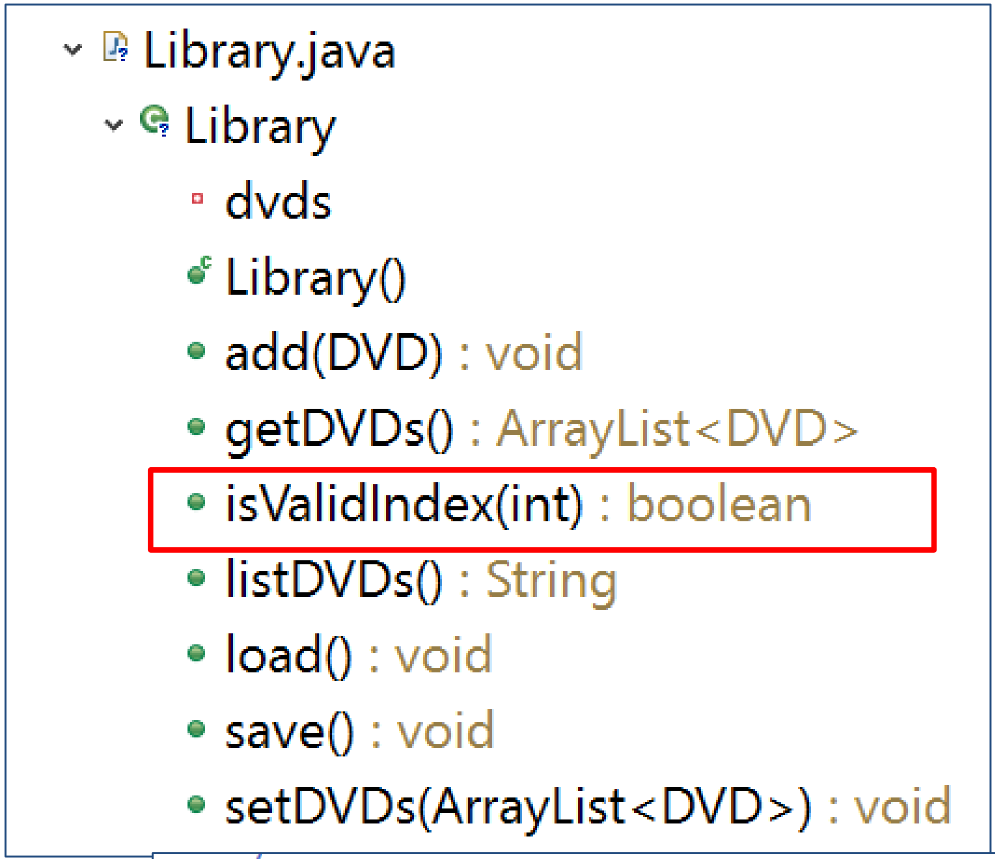


```java
public class LibraryTest {

    private Library library, populatedLibrary;
    private DVD dvd1, dvd2, dvd3;
    private ArrayList<DVD> emptyDVDs, populatedDVDs;

    @Before
    public void setUp() throws Exception {

        library = new Library();
        emptyDVDs = new ArrayList<DVD>();

        populatedDVDs = new ArrayList<DVD>();
        dvd1 = new DVD("The Hobbit(Director)");  //title with 20 characters
        dvd2 = new DVD("The Steve Jobs Film");   //title with 19 characters
        dvd3 = new DVD("Avatar: Directors Cut"); //title with 21 characters
        populatedDVDs.add(dvd1);
        populatedDVDs.add(dvd2);
        populatedDVDs.add(dvd3);
        populatedLibrary = new Library();
        populatedLibrary.setDVDs(populatedDVDs);
    }
```

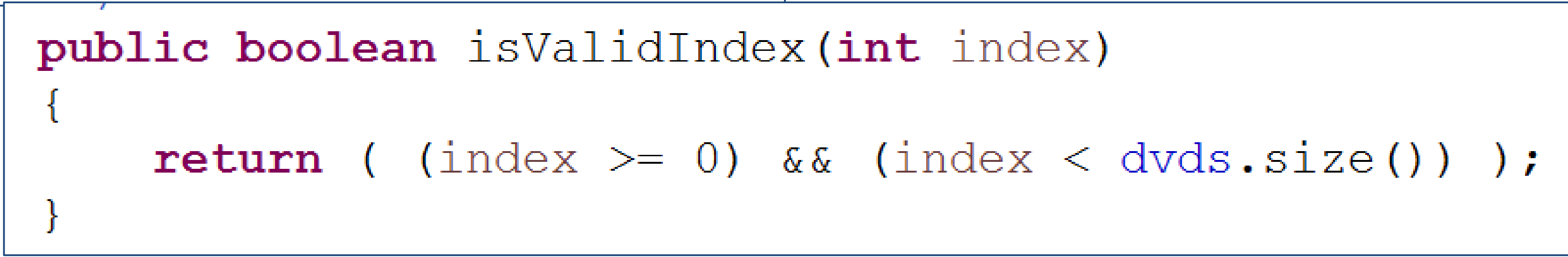
Library.java – testing save and load

```
@Test
public void testSaveAndLoad() throws Exception
{
    //TESTING AN EMPTY ARRAYLIST
    //--------------------------
    //Saving a new library object with an empty ArrayList of DVD
    assertEquals(0, library.getDVDs().size());
    assertEquals(emptyDVDs, library.getDVDs());
    library.save();
    //Load the file into another library object and compare it to emptyDVDs
    Library library2 = new Library();
    library2.load();
    assertEquals(library2.getDVDs().size(), library.getDVDs().size());

    //TESTING A POPULATED ARRAYLIST
    //-----------------------------
    //Saving a library object with a populated ArrayList of DVD
    assertEquals(3, populatedLibrary.getDVDs().size());
    assertEquals(populatedDVDs, populatedLibrary.getDVDs());
    populatedLibrary.save();
    //Load the file into another library object and compare it to populatedLibrary
    Library library3 = new Library();
    library3.load();
    assertEquals(library3.getDVDs().size(), populatedLibrary.getDVDs().size());
    assertEquals(library3.getDVDs().get(1).getTitle(), populatedLibrary.getDVDs().get(1).getTitle());
    assertEquals(library3.getDVDs().get(2).getTitle(), populatedLibrary.getDVDs().get(2).getTitle());
}
```

Library.java – testing isValidIndex(int)



```
Library.java
  Library
    dvds
    Library()
    add(DVD) : void
    getDVDs() : ArrayList<DVD>
    isValidIndex(int) : boolean
    listDVDs() : String
    load() : void
    save() : void
    setDVDs(ArrayList<DVD>) : void
```



```java
public boolean isValidIndex(int index)
{
    return ( (index >= 0) && (index < dvds.size()) );
}
```

Library.java – testing isValidIndex(int)

```java
@Test
public void testIsValidIndex() {
    //Boundary testing the library object, which has an empty ArrayList of DVDs.
    assertEquals(0, library.getDVDs().size());
    assertEquals(false, library.isValidIndex(-1));
    assertEquals(false, library.isValidIndex(0));
    assertEquals(false, library.isValidIndex(1));

    //Boundary testing the library object which has an ArrayList of three DVDs.
    //lower boundary tests / fence post error testing
    assertEquals(3, populatedLibrary.getDVDs().size());
    assertEquals(false, populatedLibrary.isValidIndex(-1));
    assertEquals(true, populatedLibrary.isValidIndex(0));
    assertEquals(true, populatedLibrary.isValidIndex(1));
    //upper boundary tests / fence post error testing
    assertEquals(true, populatedLibrary.isValidIndex(2));
    assertEquals(false, populatedLibrary.isValidIndex(3));
}
```

```java
public boolean isValidIndex(int index)
{
    return ( (index >= 0) && (index < dvds.size()) );
}
```

Topic List

- DVD and DVDTest.java
- JUnit Testing of Library.java (which includes testing of XML reading/writing)
- Testing Driver.java

Driver.java

- JUnit is not used to test the class that takes input from the console.
- Why do you think this is?

Any Questions?