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

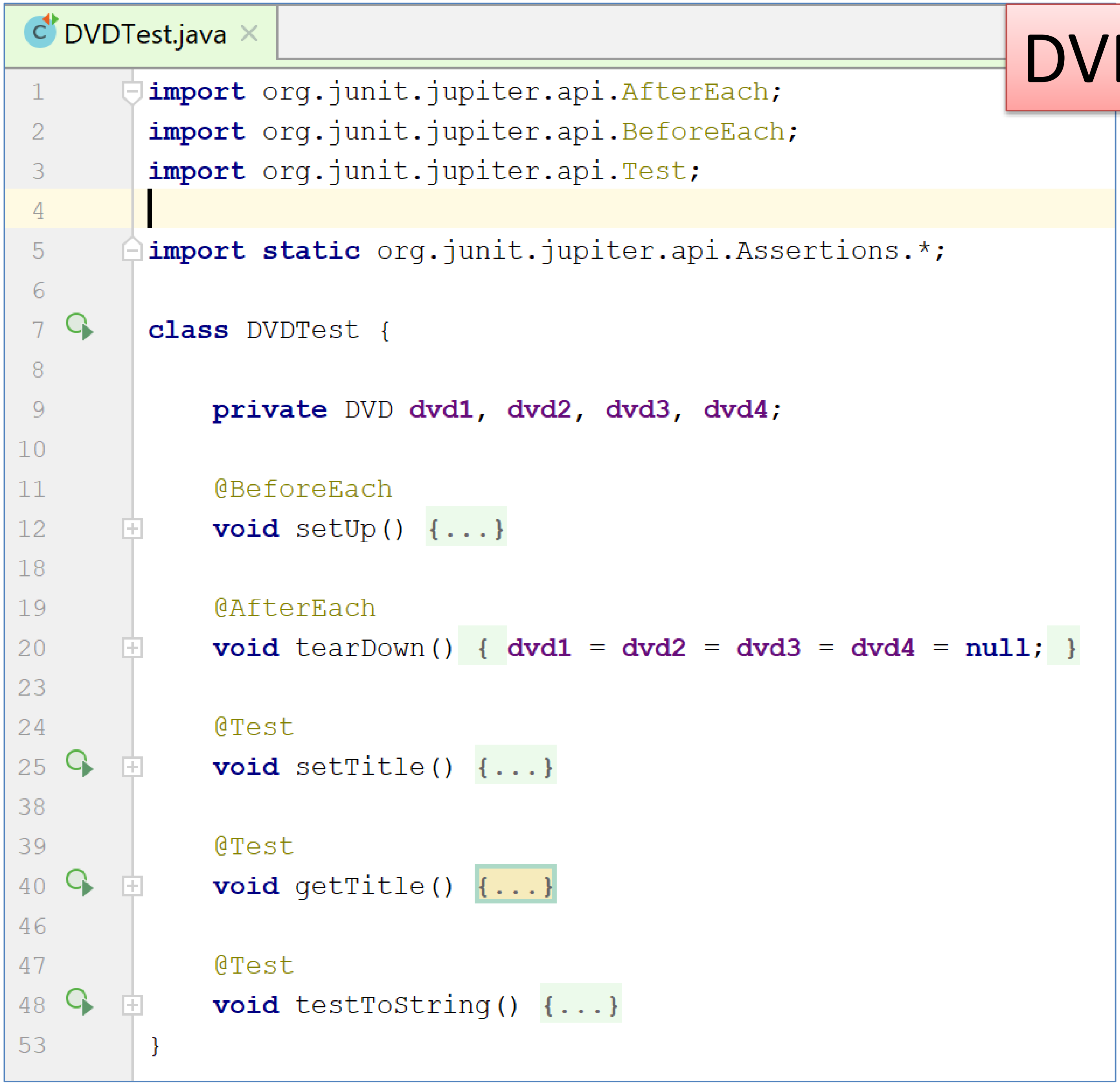
- JUnit Testing of Library.java (which includes testing of XML reading/writing)

- 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

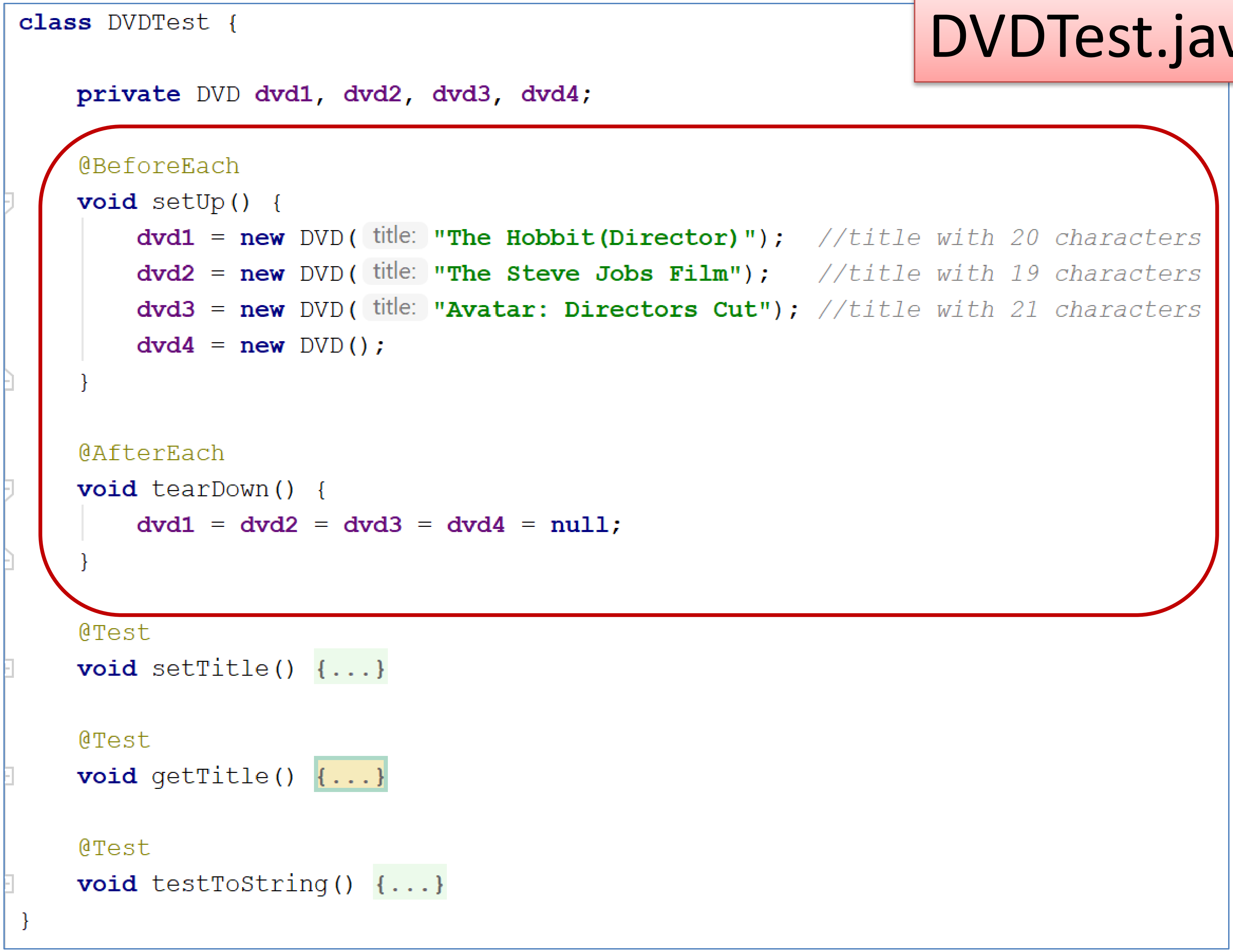
DVDTest.java



DVDTest.java

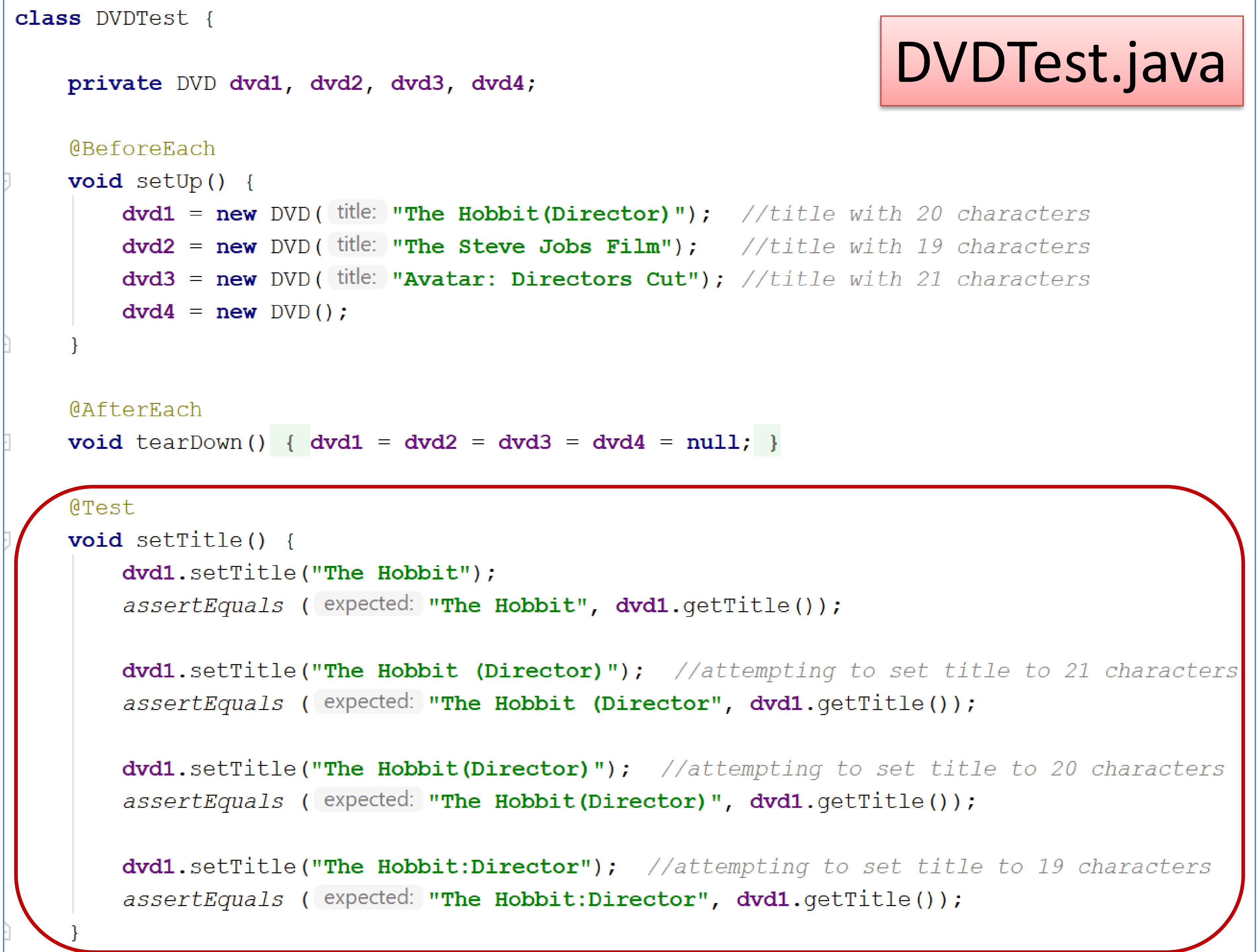
```java
import org.junit.jupiter.api.AfterEach;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;

import static org.junit.jupiter.api.Assertions.*;

class DVDTest {

    private DVD dvd1, dvd2, dvd3, dvd4;

    @BeforeEach
    void setUp() {...}

    @AfterEach
    void tearDown() { dvd1 = dvd2 = dvd3 = dvd4 = null; }

    @Test
    void setTitle() {...}

    @Test
    void getTitle() {...}

    @Test
    void testToString() {...}
}
```

DVDTest.java



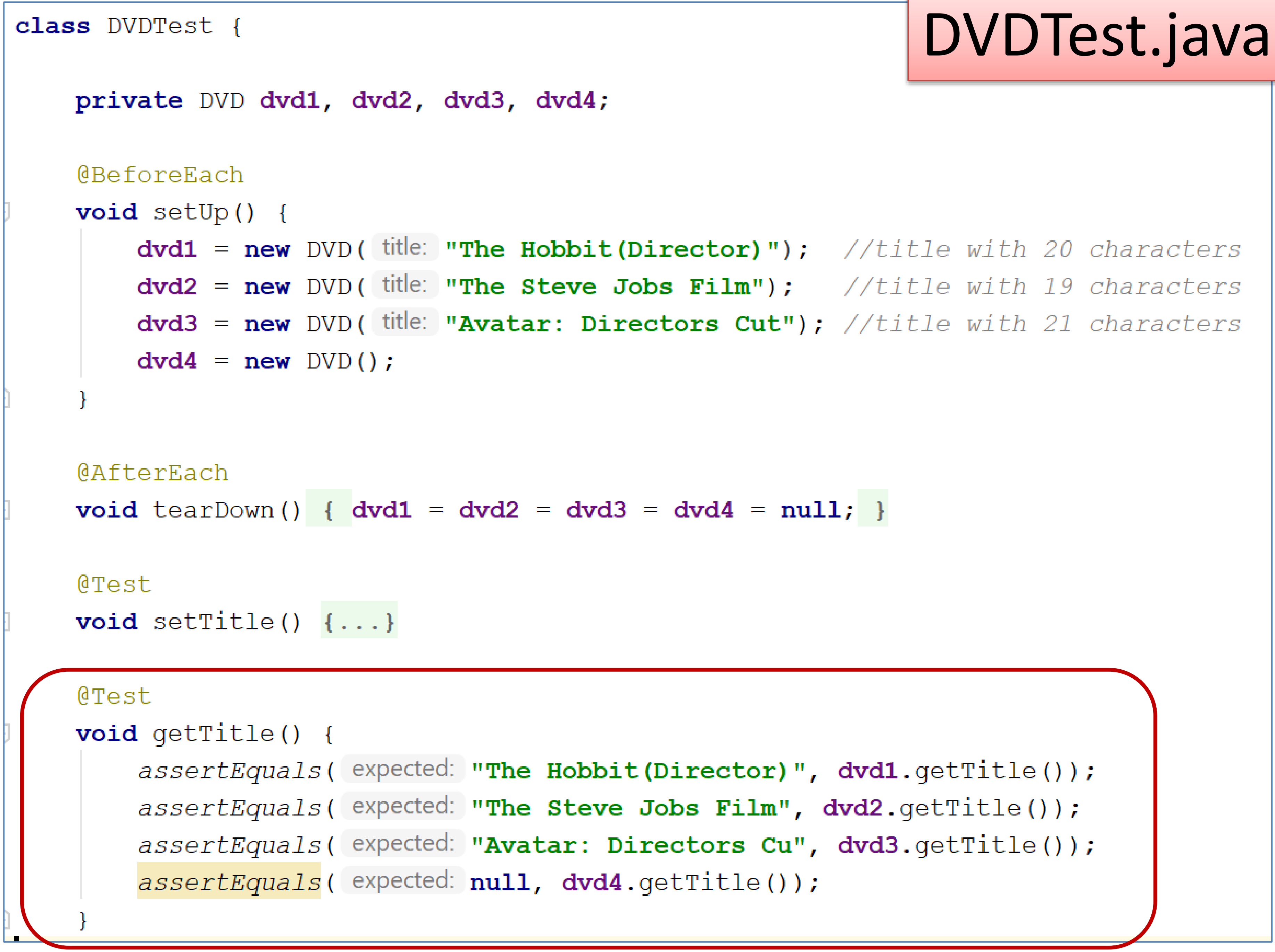
```java
class DVDTest {

    private DVD dvd1, dvd2, dvd3, dvd4;

    @BeforeEach
    void setUp() {
        dvd1 = new DVD( title: "The Hobbit(Director)");  //title with 20 characters
        dvd2 = new DVD( title: "The Steve Jobs Film");   //title with 19 characters
        dvd3 = new DVD( title: "Avatar: Directors Cut"); //title with 21 characters
        dvd4 = new DVD();
    }

    @AfterEach
    void tearDown() {
        dvd1 = dvd2 = dvd3 = dvd4 = null;
    }

    @Test
    void setTitle() {...}

    @Test
    void getTitle() {...}

    @Test
    void testToString() {...}
}
```

DVDTest.java



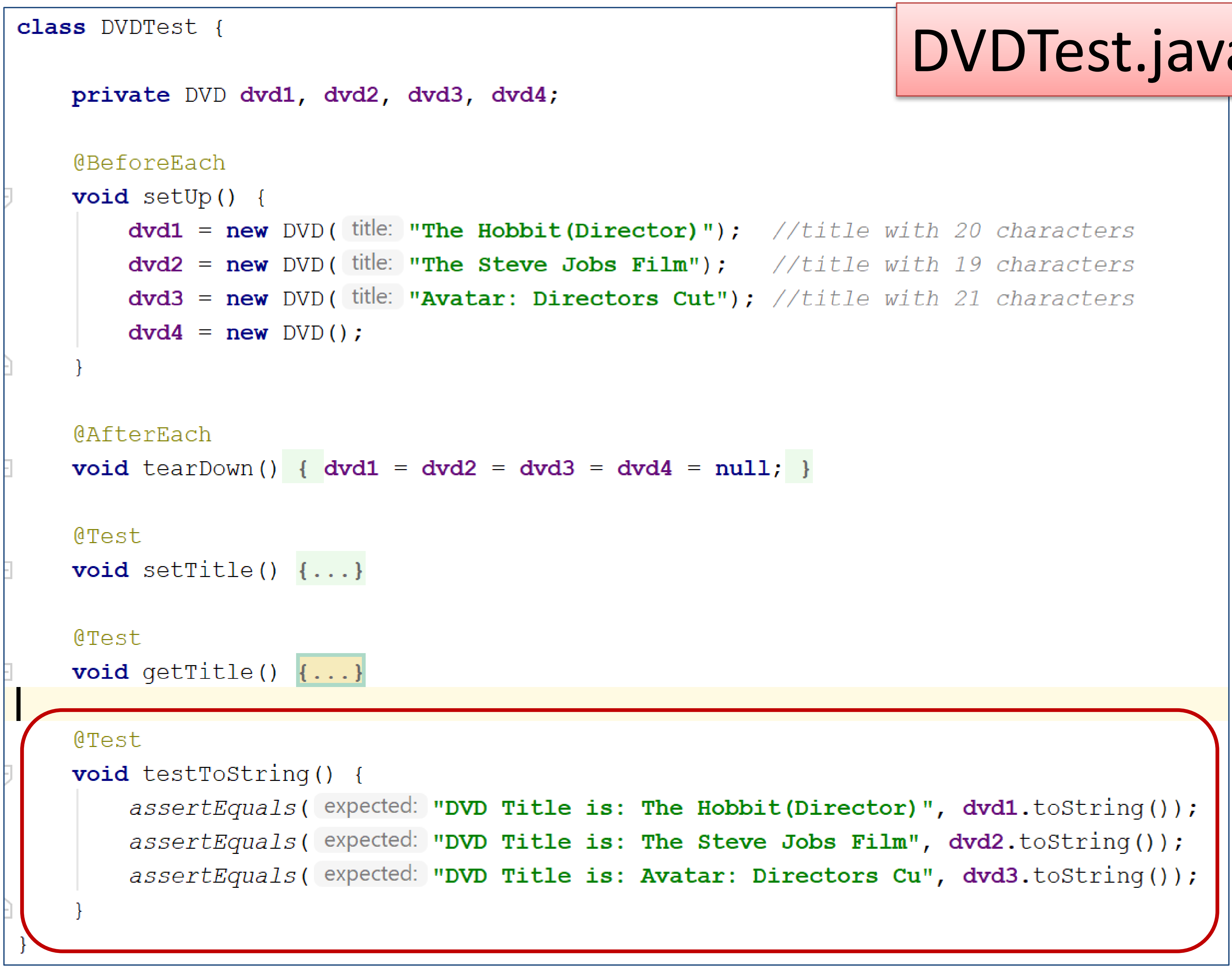
```
class DVDTest {

    private DVD dvd1, dvd2, dvd3, dvd4;

    @BeforeEach
    void setUp() {
        dvd1 = new DVD( title: "The Hobbit(Director)");  //title with 20 characters
        dvd2 = new DVD( title: "The Steve Jobs Film");   //title with 19 characters
        dvd3 = new DVD( title: "Avatar: Directors Cut"); //title with 21 characters
        dvd4 = new DVD();
    }

    @AfterEach
    void tearDown() { dvd1 = dvd2 = dvd3 = dvd4 = null; }

    @Test
    void setTitle() {
        dvd1.setTitle("The Hobbit");
        assertEquals ( expected: "The Hobbit", dvd1.getTitle());

        dvd1.setTitle("The Hobbit (Director)");  //attempting to set title to 21 characters
        assertEquals ( expected: "The Hobbit (Director", dvd1.getTitle());

        dvd1.setTitle("The Hobbit(Director)");  //attempting to set title to 20 characters
        assertEquals ( expected: "The Hobbit(Director)", dvd1.getTitle());

        dvd1.setTitle("The Hobbit:Director");  //attempting to set title to 19 characters
        assertEquals ( expected: "The Hobbit:Director", dvd1.getTitle());
    }
```

DVDTest.java



```java
class DVDTest {

    private DVD dvd1, dvd2, dvd3, dvd4;

    @BeforeEach
    void setUp() {
        dvd1 = new DVD( title: "The Hobbit(Director)");  //title with 20 characters
        dvd2 = new DVD( title: "The Steve Jobs Film");   //title with 19 characters
        dvd3 = new DVD( title: "Avatar: Directors Cut"); //title with 21 characters
        dvd4 = new DVD();
    }

    @AfterEach
    void tearDown() { dvd1 = dvd2 = dvd3 = dvd4 = null; }

    @Test
    void setTitle() {...}

    @Test
    void getTitle() {
        assertEquals( expected: "The Hobbit(Director)", dvd1.getTitle());
        assertEquals( expected: "The Steve Jobs Film", dvd2.getTitle());
        assertEquals( expected: "Avatar: Directors Cu", dvd3.getTitle());
        assertEquals( expected: null, dvd4.getTitle());
    }
```

DVDTest.java



```java
class DVDTest {

    private DVD dvd1, dvd2, dvd3, dvd4;

    @BeforeEach
    void setUp() {
        dvd1 = new DVD( title: "The Hobbit(Director)");  //title with 20 characters
        dvd2 = new DVD( title: "The Steve Jobs Film");   //title with 19 characters
        dvd3 = new DVD( title: "Avatar: Directors Cut"); //title with 21 characters
        dvd4 = new DVD();
    }

    @AfterEach
    void tearDown() { dvd1 = dvd2 = dvd3 = dvd4 = null; }

    @Test
    void setTitle() {...}

    @Test
    void getTitle() {...}

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

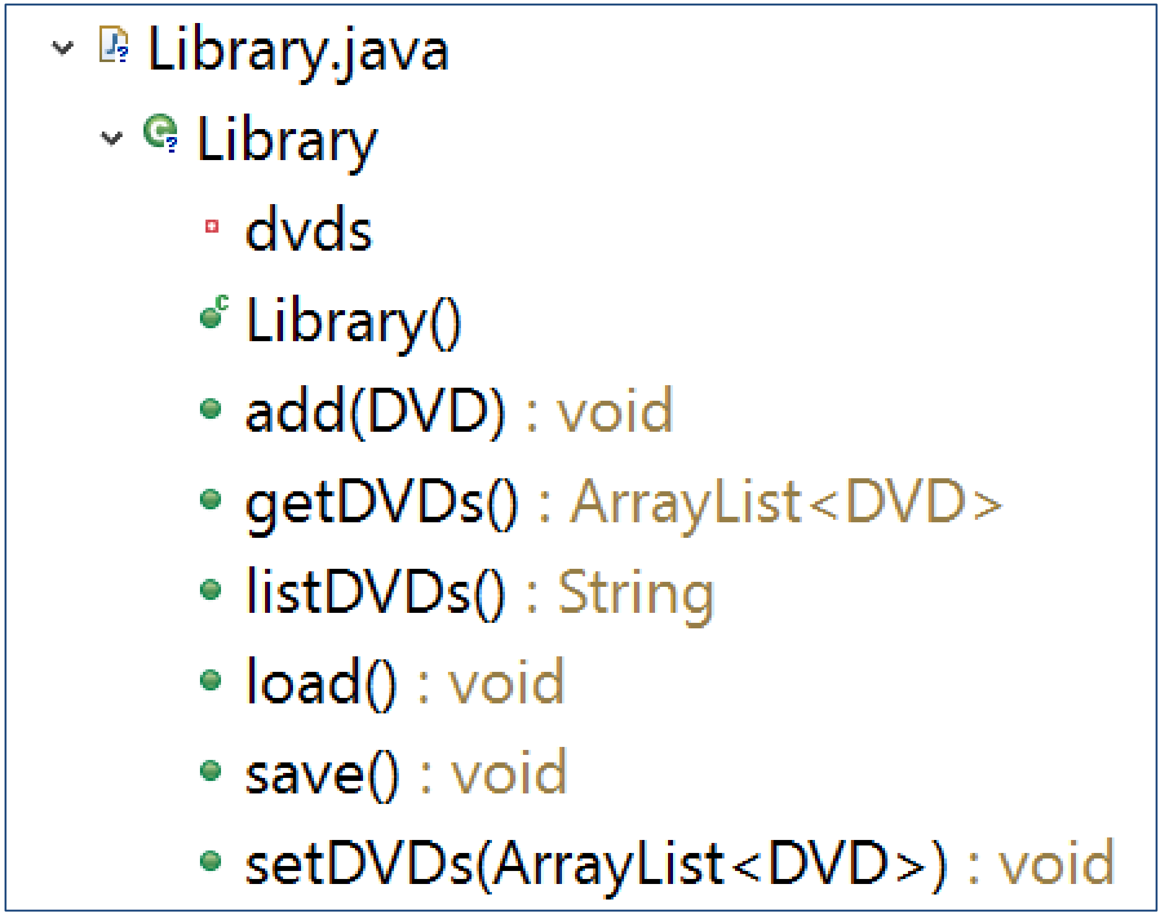
Topic List

- DVD and DVDTest.java

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

- Testing Driver.java

Library.java



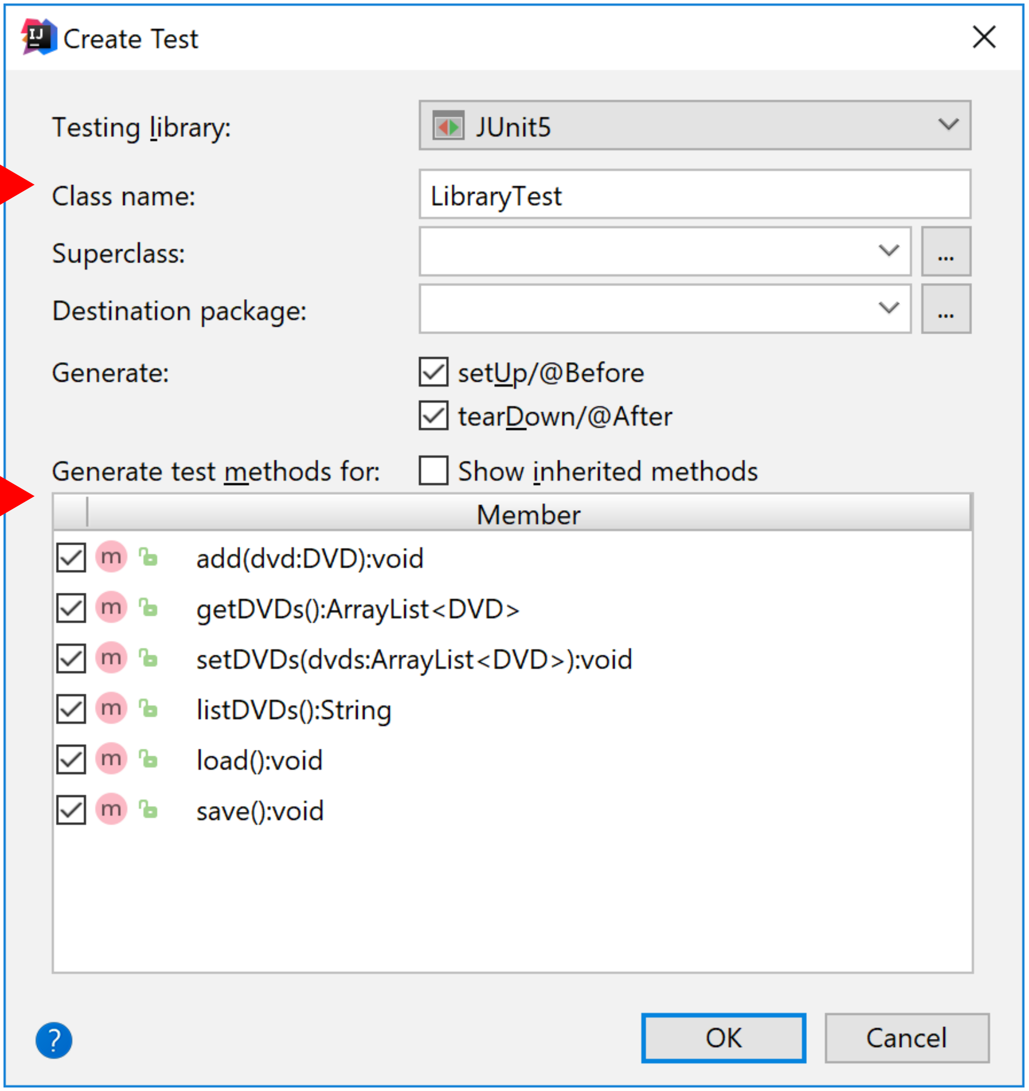
```
Library.java
  Library
    dvds
    Library()
    add(DVD) : void
    getDVDs() : ArrayList<DVD>
    listDVDs() : String
    load() : void
    save() : void
    setDVDs(ArrayList<DVD>) : void
```

We need to write a test for each of these methods.

Open Library.java and call "Create Test"

Call the test class, **LibraryTest**

Generate the default setUp() and tearDown() methods and also generate test methods for all member methods.



Create Test

Testing library: JUnit5

Class name: LibraryTest

Superclass:

Destination package:

Generate: [✓] setUp/@Before [✓] tearDown/@After

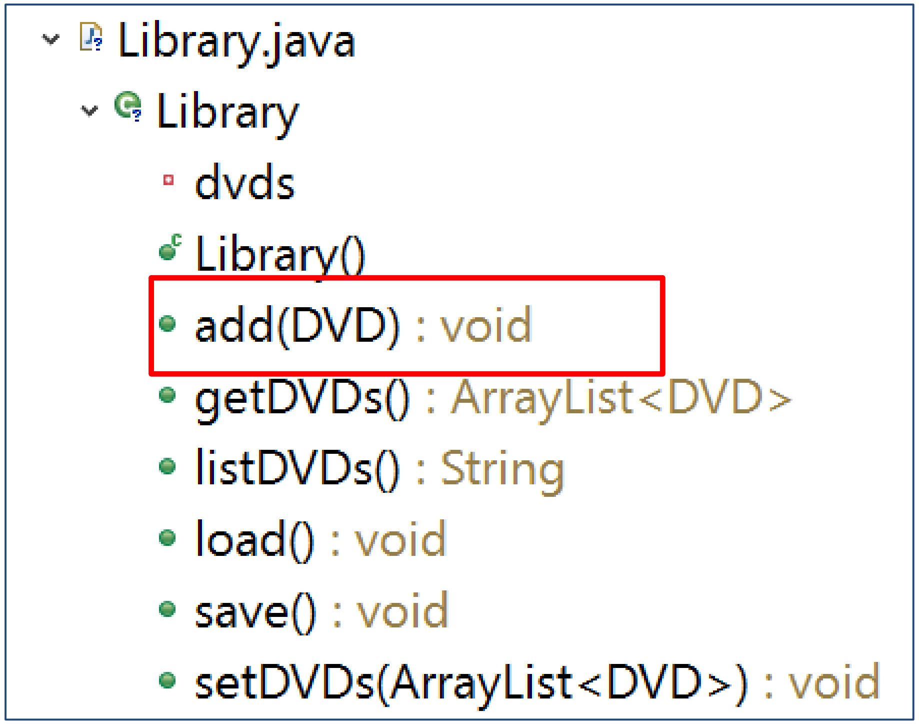
Generate test methods for: [ ] Show inherited methods

	Member
✓	add(dvd:DVD):void
✓	getDVDs():ArrayList<DVD>
✓	setDVDs(dvds:ArrayList<DVD>):void
✓	listDVDs():String
✓	load():void
✓	save():void

```java
import org.junit.jupiter.api.AfterEach;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;
import static org.junit.jupiter.api.Assertions.*;

class LibraryTest {

    @BeforeEach
    void setUp() {
    }

    @AfterEach
    void tearDown() {
    }

    @Test
    void add() {
    }

    @Test
    void getDVDs() {
    }

    @Test
    void setDVDs() {
    }

    @Test
    void listDVDs() {
    }

    @Test
    void load() {
    }

    @Test
    void save() {
    }
}
```

Generated LibraryTest.java

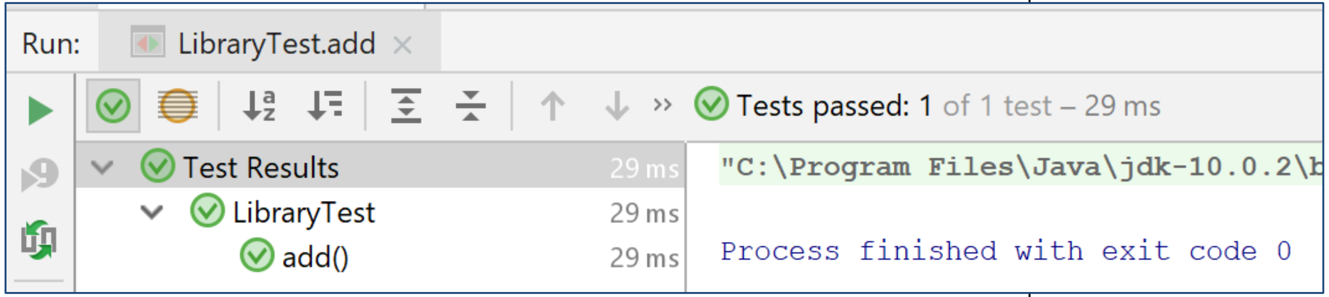
Library.java – testing add(DVD)



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

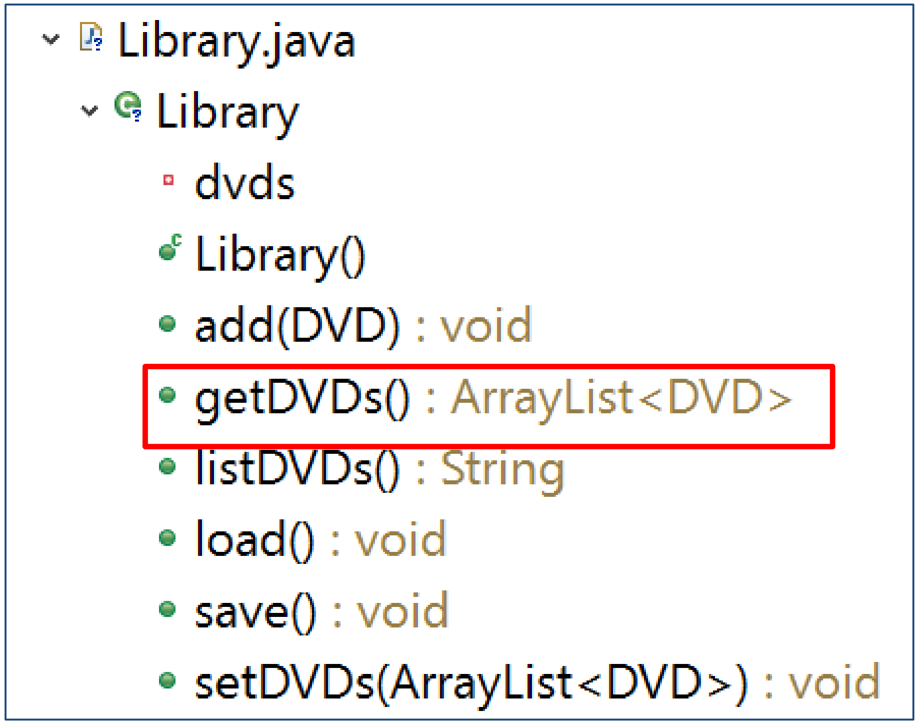
```
public void add(DVD dvd){
    dvds.add(dvd);
}
```

```java
class LibraryTest {

    private Library library;

    @BeforeEach
    void setUp() {
        library = new Library();
    }

    @AfterEach
    void tearDown() {
        library = null;
    }

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

        //Testing the adding of the first dvd and making sure the title
        //was setup correctly.
        library.add(new DVD("The Avengers"));
        assertEquals(1, library.getDVDs().size());
        assertEquals("The Avengers", 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());
    }
}
```

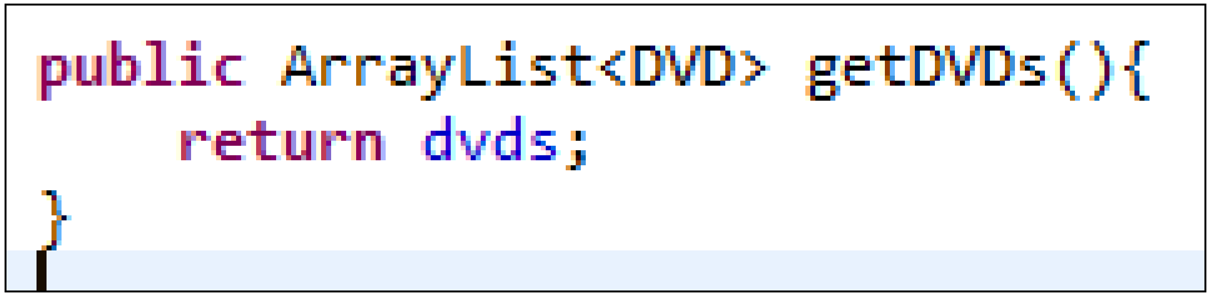


```
Run:  LibraryTest.add ×
✓ Tests passed: 1 of 1 test – 29 ms
✓ Test Results      29 ms
  ✓ LibraryTest     29 ms
    ✓ add()         29 ms

"C:\Program Files\Java\jdk-10.0.2\b

Process finished with exit code 0
```

Library.java – testing getDVDs()



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



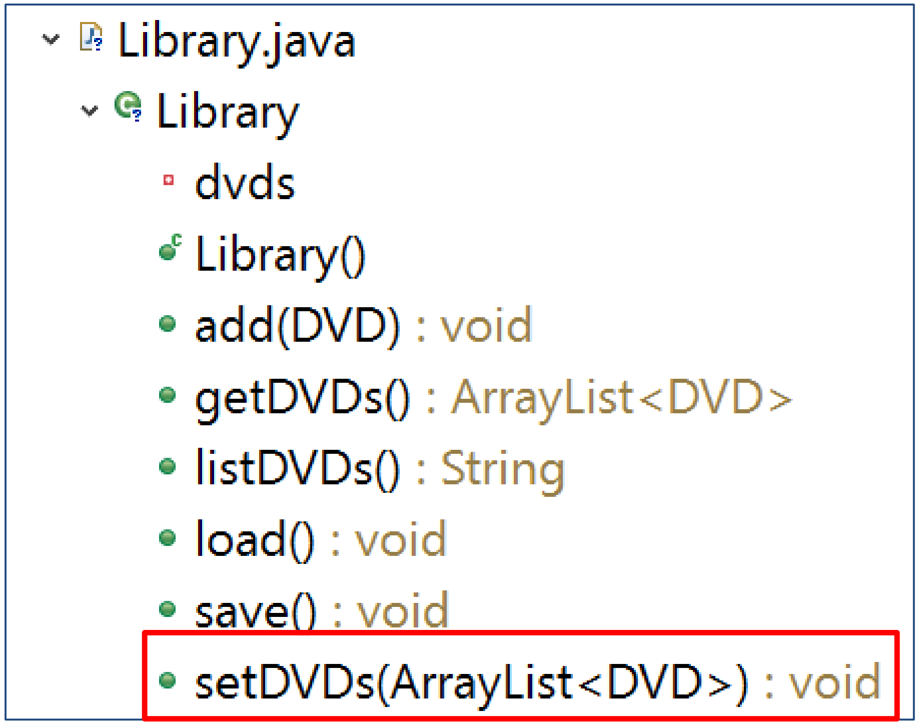
```
public ArrayList<DVD> getDVDs(){
    return dvds;
}
```

```java
class LibraryTest {

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

    @BeforeEach
    void setUp() {
        //A Library object that will be empty at the beginning of each test
        library = new Library();

        //An empty ArrayList of DVDs created independently of the Library class.
        //This will be used to compare with the ArrayList created in Library.
        emptyDVDs = new ArrayList<DVD>();

        //A populated ArrayList of DVDs created independently of the Library class.
        //This will be used to compare with the ArrayList created in Library.
        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);

        //A Library object that will be populated with three DVDs at the beginning of each test
        populatedLibrary = new Library();
        populatedLibrary.setDVDs(populatedDVDs);
    }

    @AfterEach
    void tearDown() {
        library = populatedLibrary = null;
        dvd1 = dvd2 = dvd3 = null;
        populatedDVDs = emptyDVDs = null;
    }
```

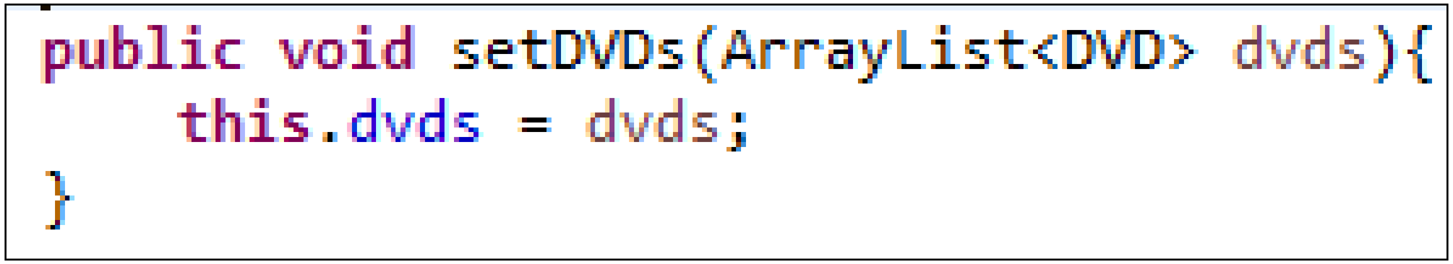
```java
class LibraryTest {

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

    @BeforeEach
    void setUp() {
        //A Library object that will be empty at the beginning of each test
        library = new Library();

        //An empty ArrayList of DVDs created independently of the Library class.
        //This will be used to compare with the ArrayList created in Library.
        emptyDVDs = new ArrayList<DVD>();

        //A populated ArrayList of DVDs created independently of the Library class.
        //This will be used to compare with the ArrayList created in Library.
        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);

        //A Library object that will be populated with three DVDs at the beginning of each test
        populatedLibrary = new Library();
        populatedLibrary.setDVDs(populatedDVDs);
    }

    @AfterEach
    void tearDown() {
        library = populatedLibrary = null;
        dvd1 = dvd2 = dvd3 = null;
        populatedDVDs = emptyDVDs = null;
    }
```

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

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

Library.java – testing setDVDs()

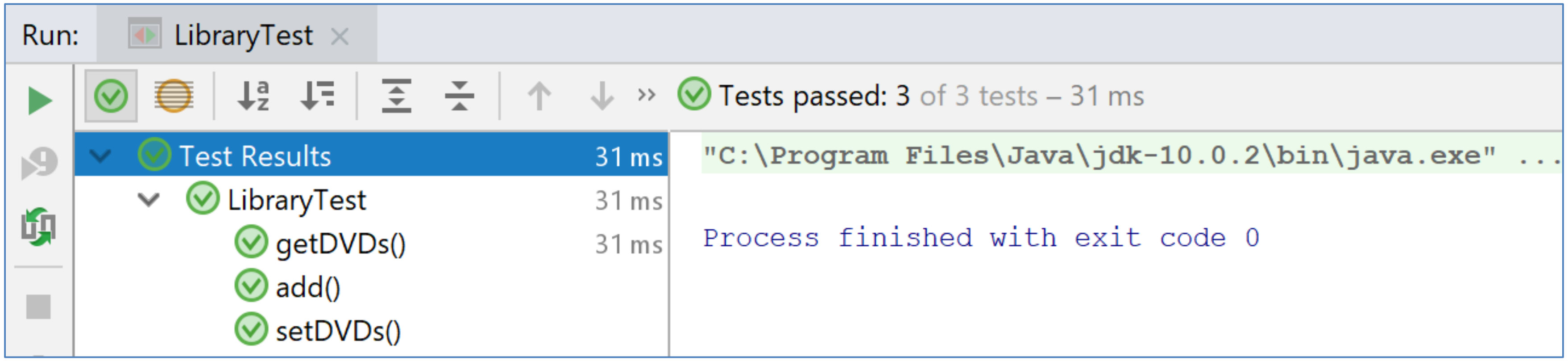


```
Library.java
  Library
    dvds
    Library()
    add(DVD) : void
    getDVDs() : ArrayList<DVD>
    listDVDs() : String
    load() : void
    save() : void
    setDVDs(ArrayList<DVD>) : void
```



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

```java
@Test
void setDVDs() {
    //The new library object size is zero.  Set the DVDs
    //ArrayList to an empty list; zero should still be returned
    assertEquals(0, library.getDVDs().size());
    library.setDVDs(emptyDVDs);
    assertEquals(0, library.getDVDs().size());

    //Now set the DVDs ArrayList with the populatedDVDs.
    //3 should still be returned
    library.setDVDs(populatedDVDs);
    assertEquals(3, library.getDVDs().size());
    //The contents of the library DVDs should also be the
    //same as the populatedDVDs.
    assertEquals(populatedDVDs, library.getDVDs());
}
```



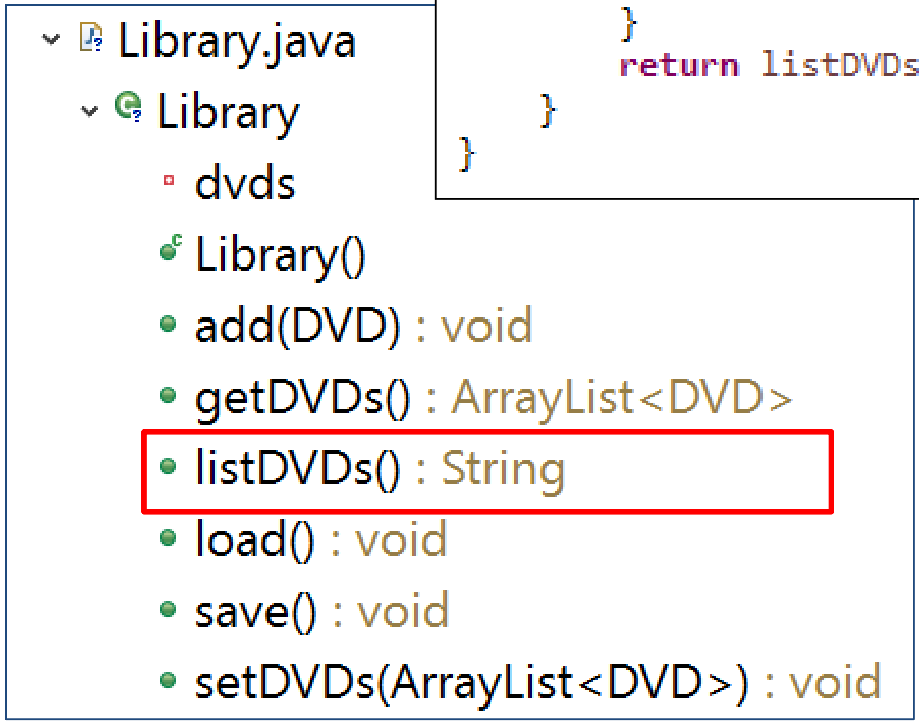
Run: LibraryTest

Tests passed: 3 of 3 tests – 31 ms

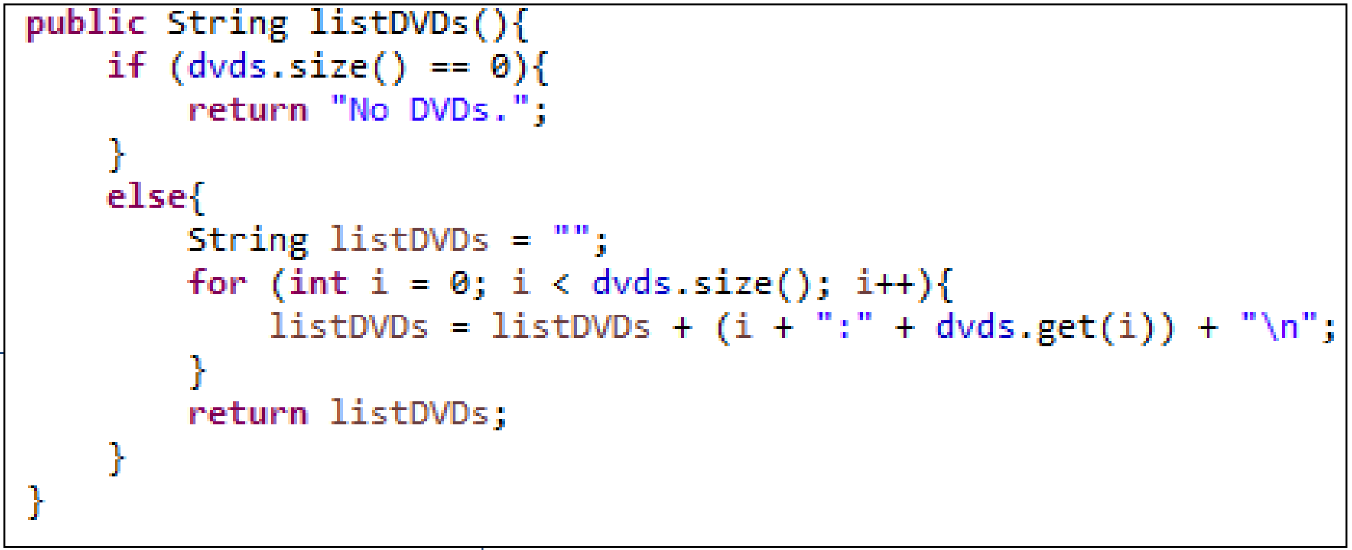
```
Test Results       31 ms
  LibraryTest      31 ms
    getDVDs()      31 ms
    add()
    setDVDs()
```

```
"C:\Program Files\Java\jdk-10.0.2\bin\java.exe" ...

Process finished with exit code 0
```

Library.java – testing listDVDs()

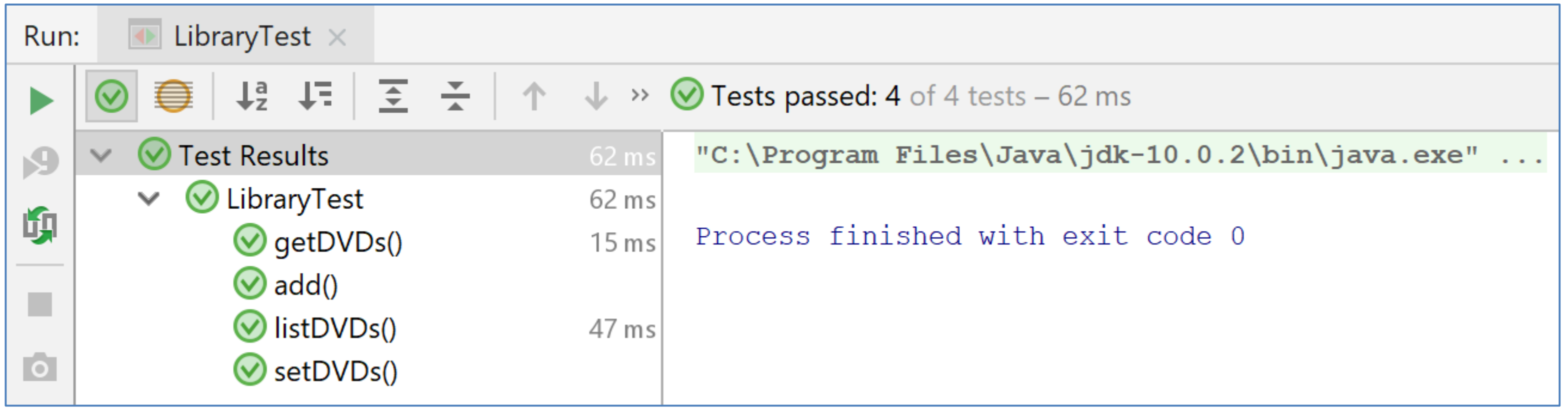


```
Library.java
  Library
    dvds
    Library()
    add(DVD) : void
    getDVDs() : ArrayList<DVD>
    listDVDs() : String
    load() : void
    save() : void
    setDVDs(ArrayList<DVD>) : void
```



```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;
    }
}
```

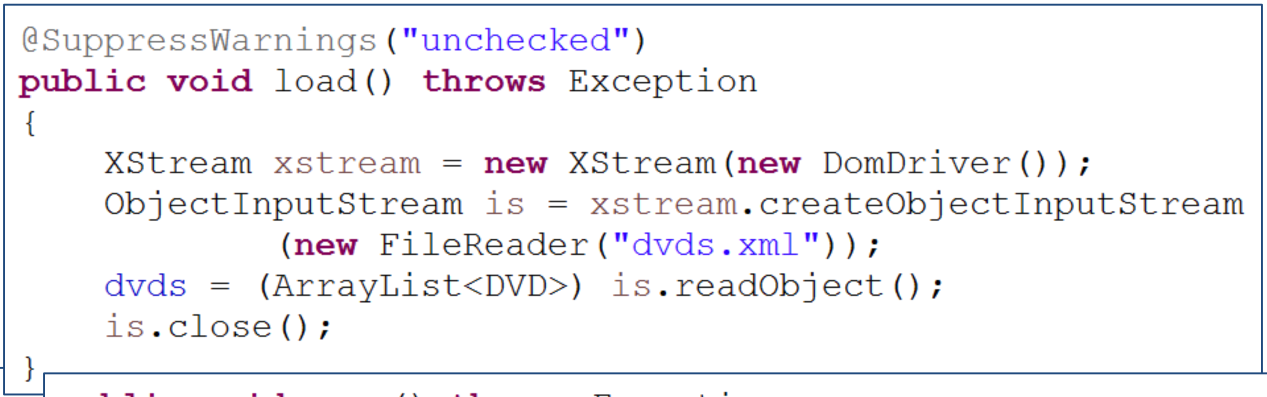
```java
@Test
void listDVDs() {
    //The new library object returns an empty String
    assertEquals("No DVDs.", library.listDVDs());

    //The populated library object returns an String listing three DVDs
    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());
}
```



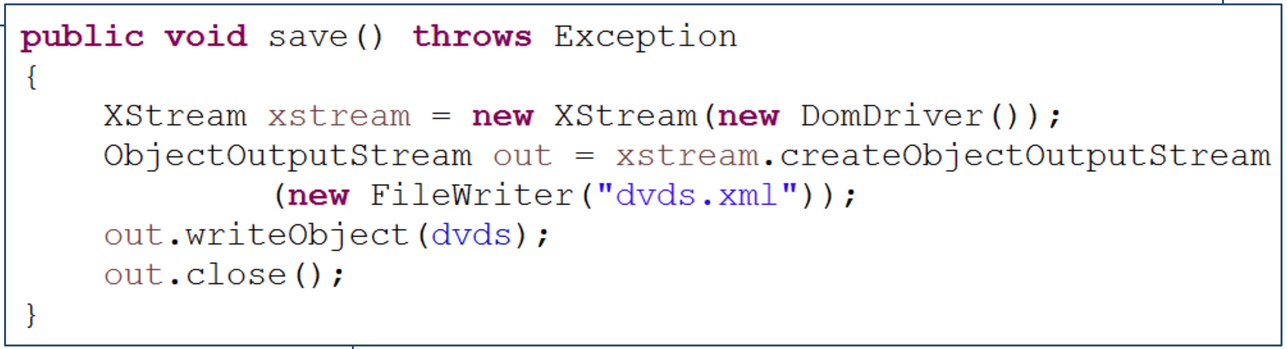
```
Run:  LibraryTest

Tests passed: 4 of 4 tests – 62 ms

Test Results          62 ms
  LibraryTest         62 ms
    getDVDs()         15 ms
    add()
    listDVDs()        47 ms
    setDVDs()
```

```
"C:\Program Files\Java\jdk-10.0.2\bin\java.exe" ...

Process finished with exit code 0
```

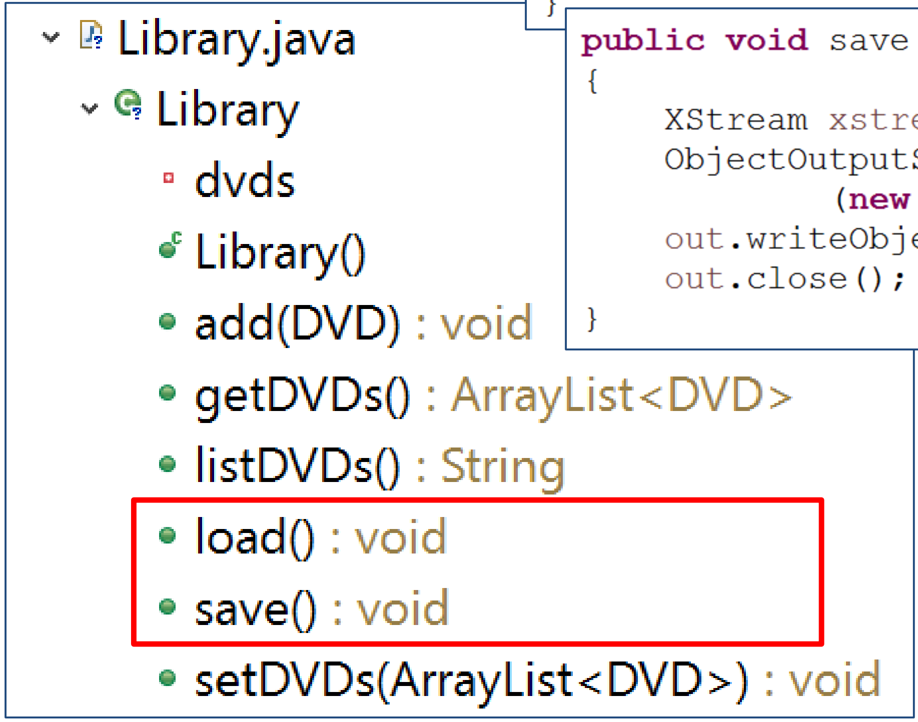
Library.java – testing save and load



```java
@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();
}
```



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

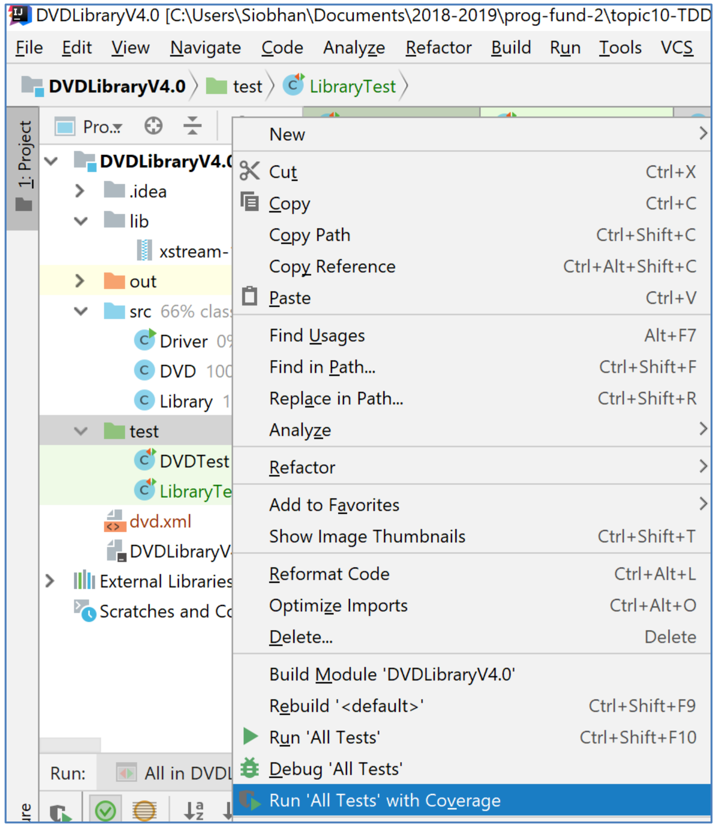
Library.java – testing save and load

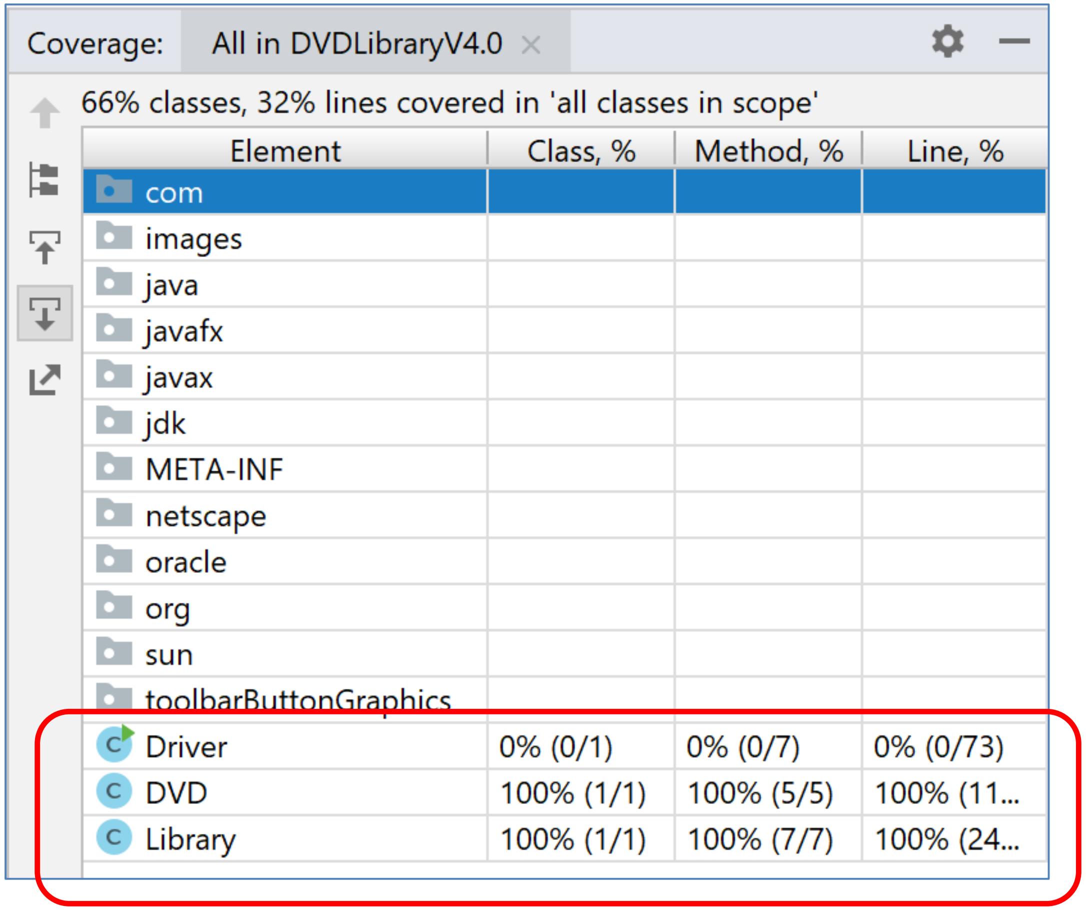
```java
private Library library, populatedLibrary;
private DVD dvd1, dvd2, dvd3;
private ArrayList<DVD> emptyDVDs, populatedDVDs;

@BeforeEach
void setUp() {
    //A Library object that will be empty at the beginning of each test
    library = new Library();

    //An empty ArrayList of DVDs created independently of the Library class.
    //This will be used to compare with the ArrayList created in Library.
    emptyDVDs = new ArrayList<DVD>();

    //A populated ArrayList of DVDs created independently of the Library class.
    //This will be used to compare with the ArrayList created in Library.
    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);

    //A Library object that will be populated with three DVDs at the beginning of each test
    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());
}
```

What's our code coverage?


DVDLibraryV4.0 [C:\Users\Siobhan\Documents\2018-2019\prog-fund-2\topic10-TDD
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS
DVDLibraryV4.0 › test › LibraryTest
1: Project
Pro.
DVDLibraryV4.0
.idea
lib
xstream-
out
src 66% class
Driver 0%
DVD 100
Library 1
test
DVDTest
LibraryTe
dvd.xml
DVDLibraryV
External Libraries
Scratches and Co
Run: All in DVDL
New
Cut Ctrl+X
Copy Ctrl+C
Copy Path Ctrl+Shift+C
Copy Reference Ctrl+Alt+Shift+C
Paste Ctrl+V
Find Usages Alt+F7
Find in Path... Ctrl+Shift+F
Replace in Path... Ctrl+Shift+R
Analyze
Refactor
Add to Favorites
Show Image Thumbnails Ctrl+Shift+T
Reformat Code Ctrl+Alt+L
Optimize Imports Ctrl+Alt+O
Delete... Delete
Build Module 'DVDLibraryV4.0'
Rebuild '<default>' Ctrl+Shift+F9
Run 'All Tests' Ctrl+Shift+F10
Debug 'All Tests'
Run 'All Tests' with Coverage



66% classes, 32% lines covered in 'all classes in scope'

Element	Class, %	Method, %	Line, %
com			
images			
java			
javafx			
javax			
jdk			
META-INF			
netscape			
oracle			
org			
sun			
toolbarButtonGraphics			
Driver	0% (0/1)	0% (0/7)	0% (0/73)
DVD	100% (1/1)	100% (5/5)	100% (11...
Library	100% (1/1)	100% (7/7)	100% (24...

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?