Test Driven Development

Using JUnit to Test DVD.java

Produced by: Dr. Siobhán Drohan
Mairead Meagher

Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

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

Let’s first write a complete test class and then we will discuss the theory behind it.

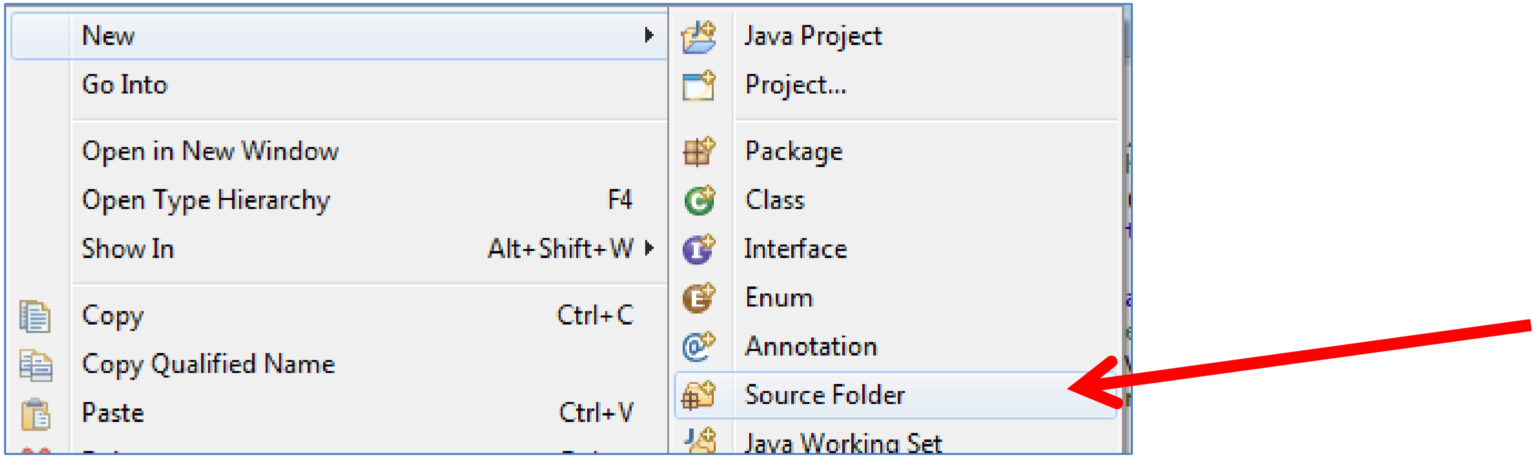
Testing DVD.java

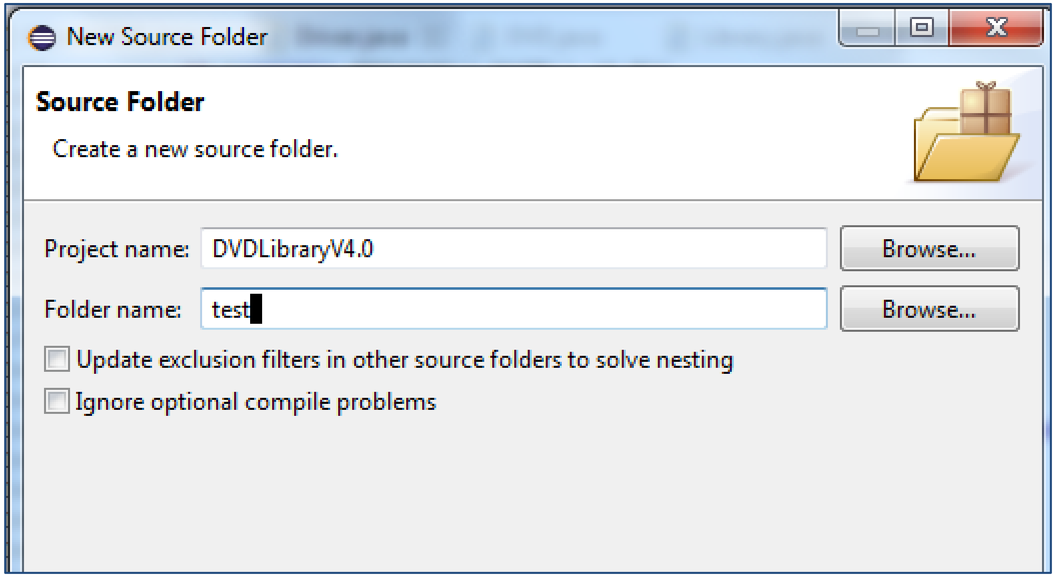
DVD.java

```
public class DVD
{
    private String title;

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

    public DVD(){
    }
```

```
    public void setTitle(String title){
        if (title.length() <= 20){
            this.title = title;
        }
    }

    public String getTitle(){
        return title;
    }

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

In Eclipse, you should create a new “test” folder to hold your test classes




New Source Folder

Source Folder

Create a new source folder.

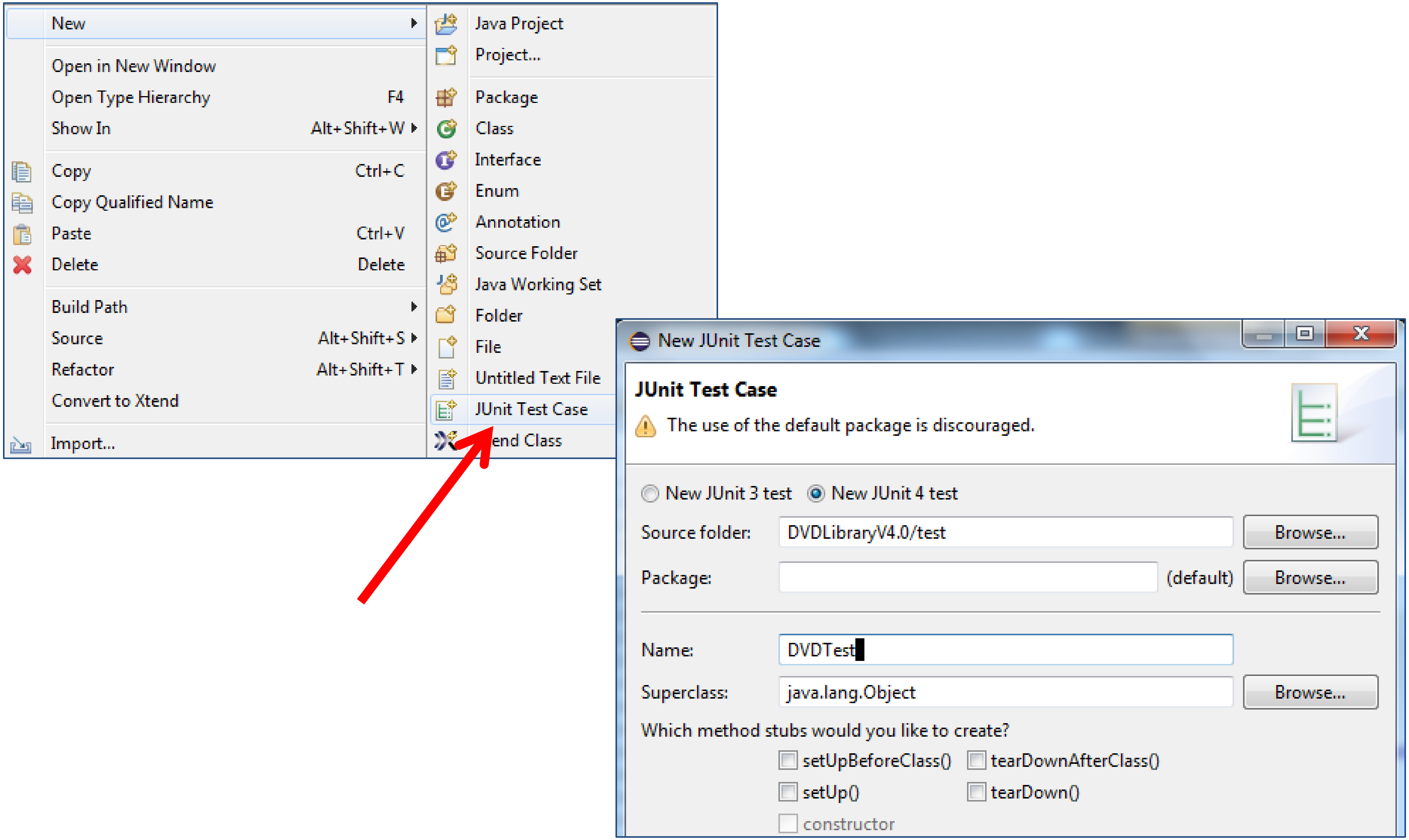
Project name: DVDLibraryV4.0 Browse...

Folder name: test Browse...

□ Update exclusion filters in other source folders to solve nesting

□ Ignore optional compile problems

Within this “test” folder, create a new JUnit Test Case



New JUnit Test Case

JUnit Test Case

⚠ The use of the default package is discouraged.

○ New JUnit 3 test ● New JUnit 4 test

Source folder: DVDLibraryV4.0/test

Package: (default)

Name: DVDTest

Superclass: java.lang.Object

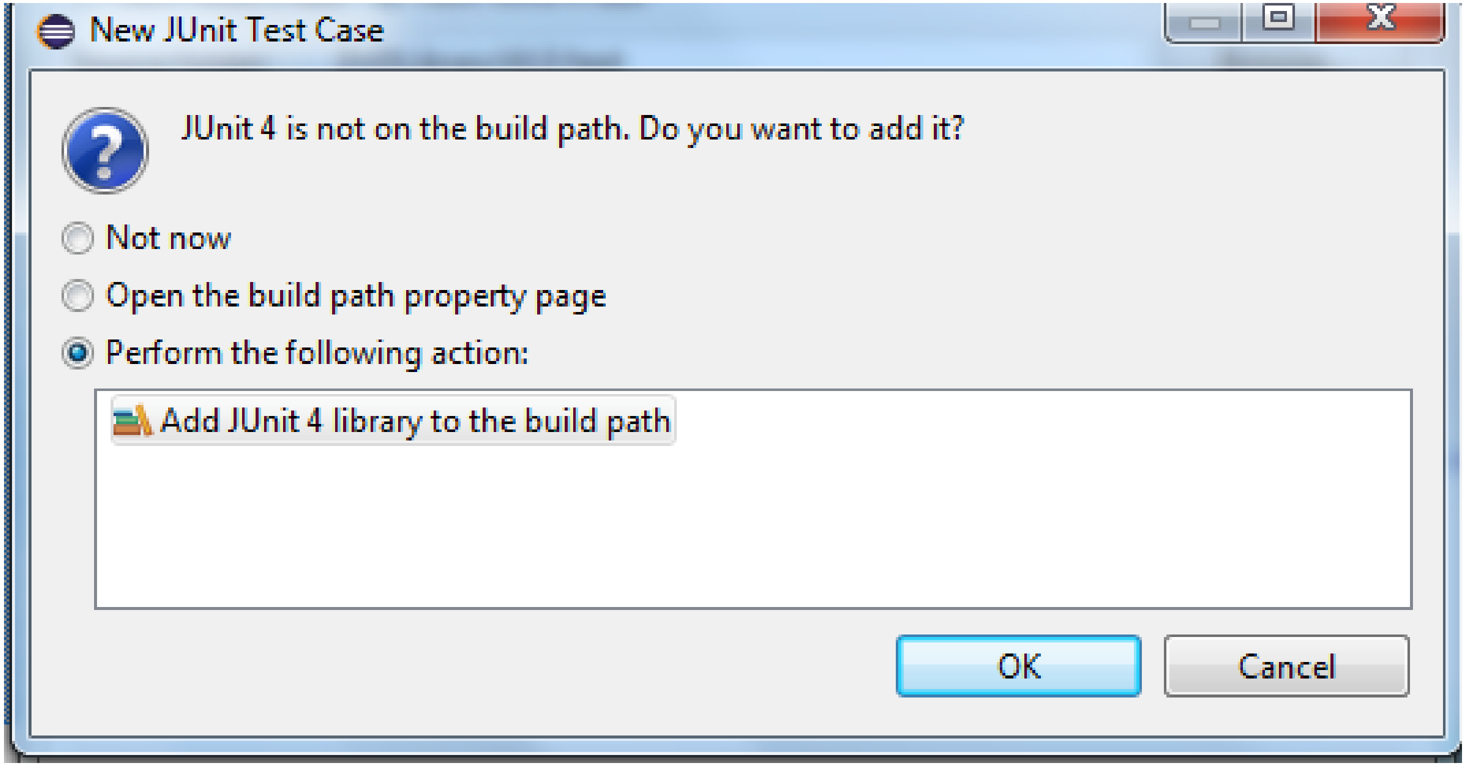
Which method stubs would you like to create?

□ setUpBeforeClass() □ tearDownAfterClass()

□ setUp() □ tearDown()

□ constructor

You may be asked to add JUnit 4 to your build path → Click OK.



New JUnit Test Case

JUnit 4 is not on the build path. Do you want to add it?

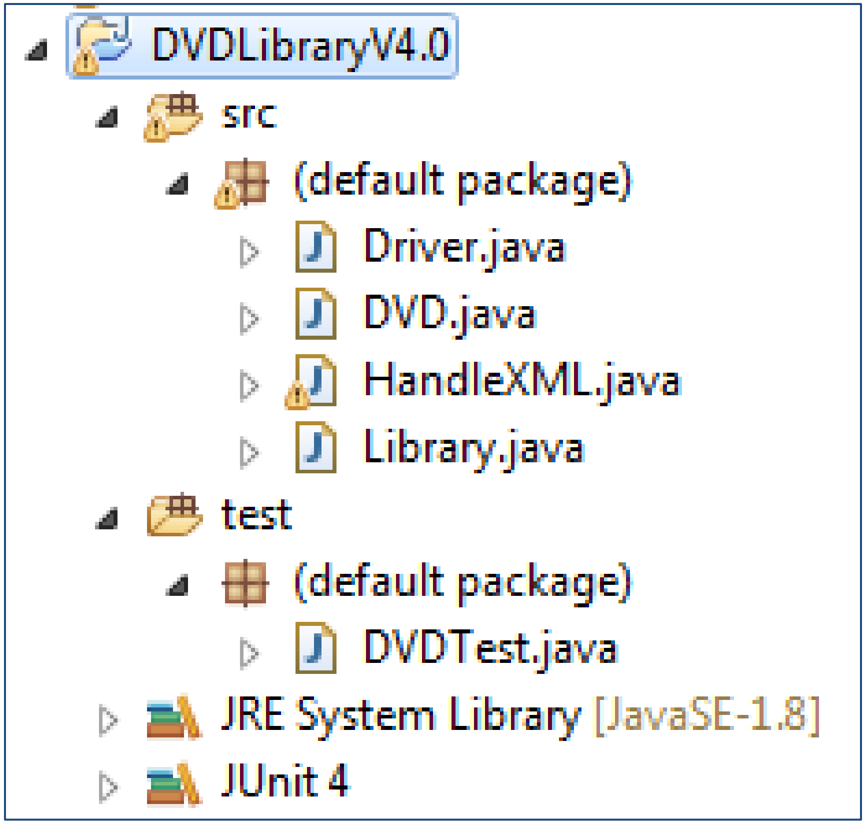
○ Not now

○ Open the build path property page

● Perform the following action:

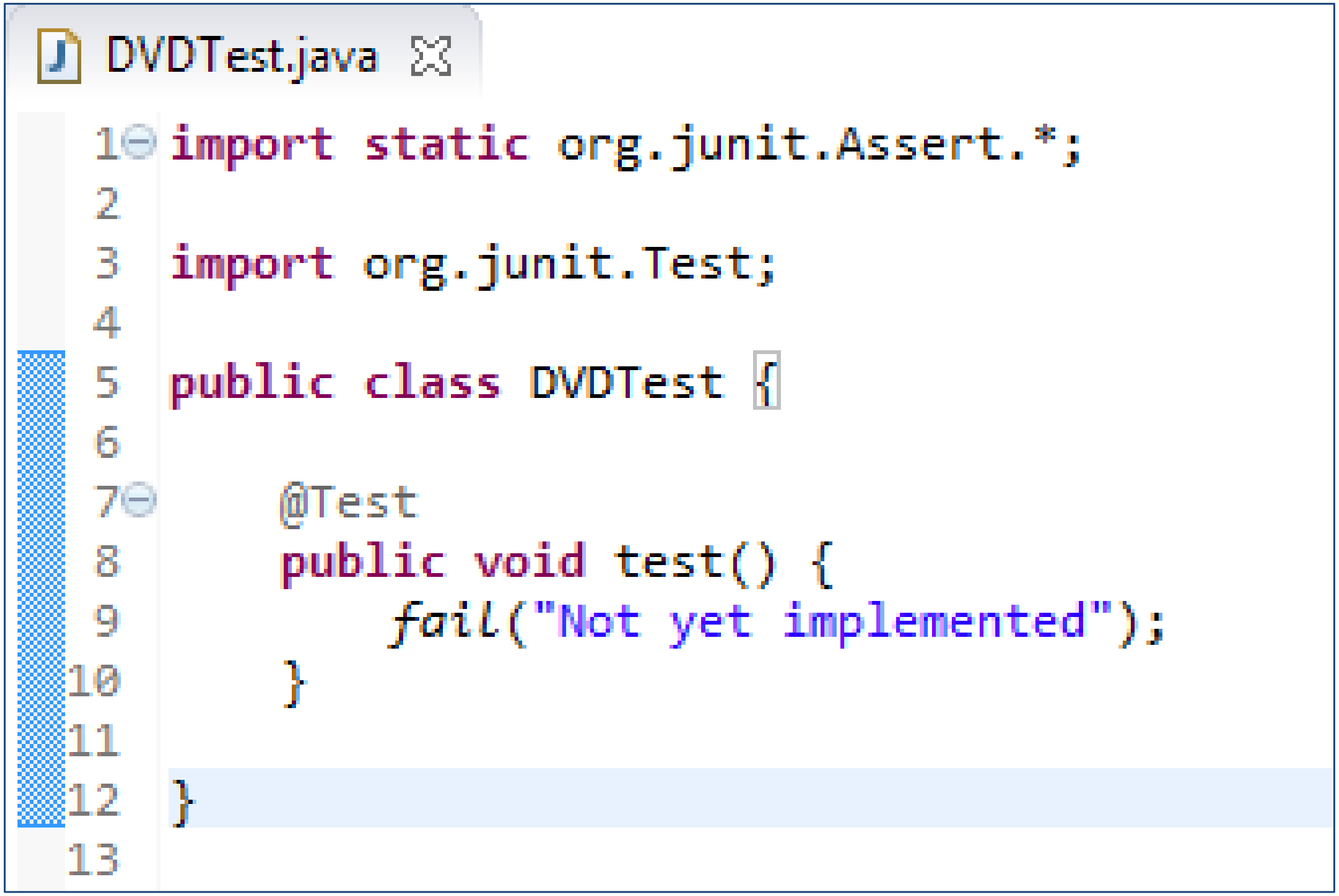
Add JUnit 4 library to the build path

Revised Folder Structure



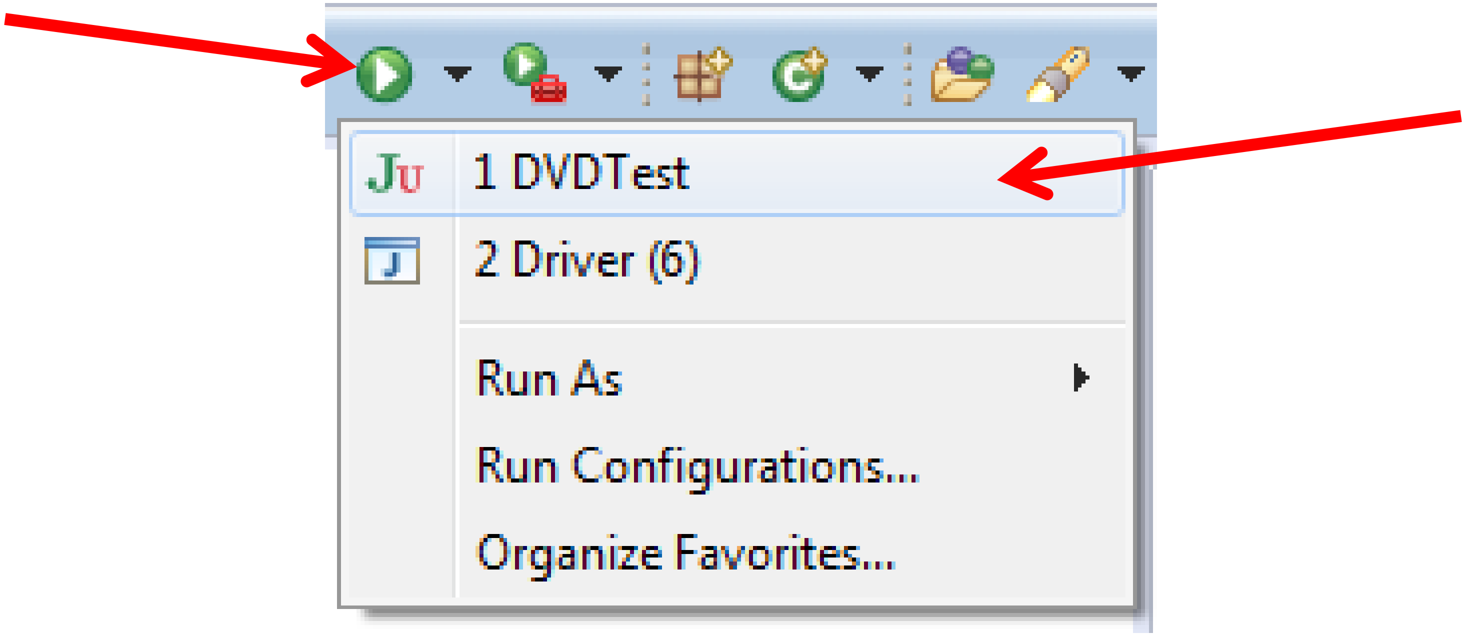
```
DVDLibraryV4.0
    src
        (default package)
            Driver.java
            DVD.java
            HandleXML.java
            Library.java
    test
        (default package)
            DVDTest.java
    JRE System Library [JavaSE-1.8]
    JUnit 4
```

The generated DVDTest.java

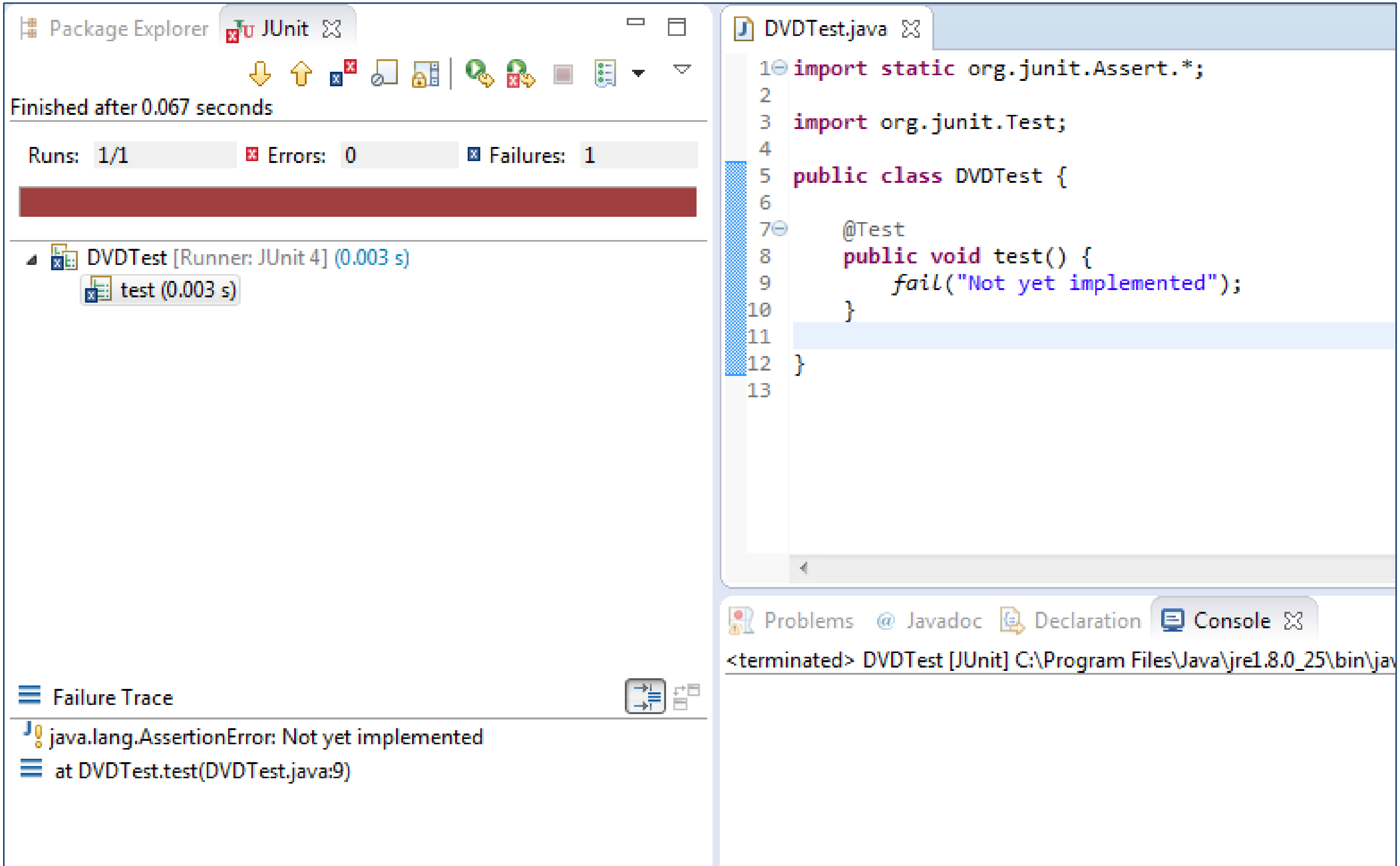


```java
import static org.junit.Assert.*;

import org.junit.Test;

public class DVDTest {

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

}

```

Running DVDTest.java



JUnit Results Pane (1)



Finished after 0.067 seconds

Runs: 1/1   Errors: 0   Failures: 1

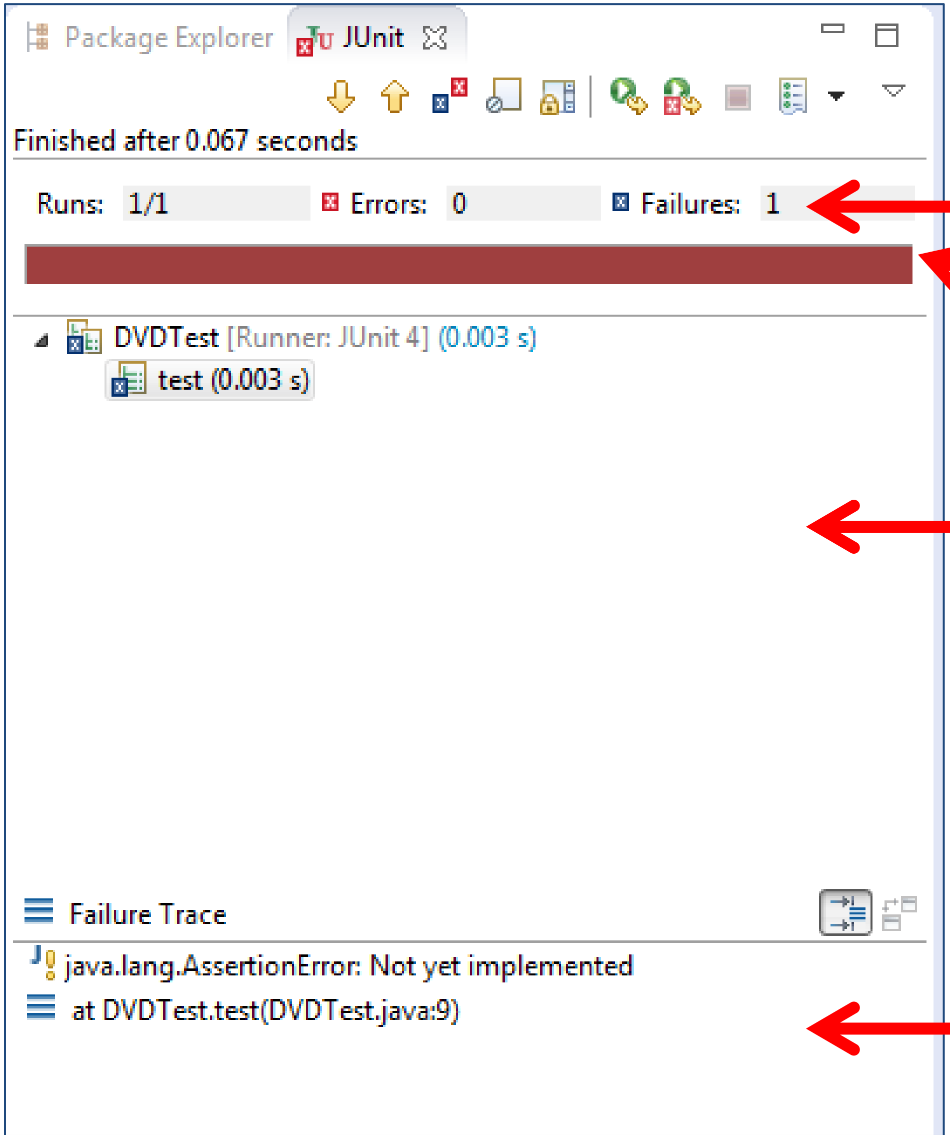
DVDTest [Runner: JUnit 4] (0.003 s)
test (0.003 s)

```
java.lang.AssertionError: Not yet implemented
at DVDTest.test(DVDTest.java:9)
```

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

import org.junit.Test;

public class DVDTest {

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

}

```

```
<terminated> DVDTest [JUnit] C:\Program Files\Java\jre1.8.0_25\bin\jav
```

JUnit Results Pane (2)



Finished after 0.067 seconds

Runs: 1/1 Errors: 0 Failures: 1

DVDTest [Runner: JUnit 4] (0.003 s)
test (0.003 s)

Failure Trace

java.lang.AssertionError: Not yet implemented
at DVDTest.test(DVDTest.java:9)

Number of tests run. Also the number that have errors (e.g. compile) and the number that failed.

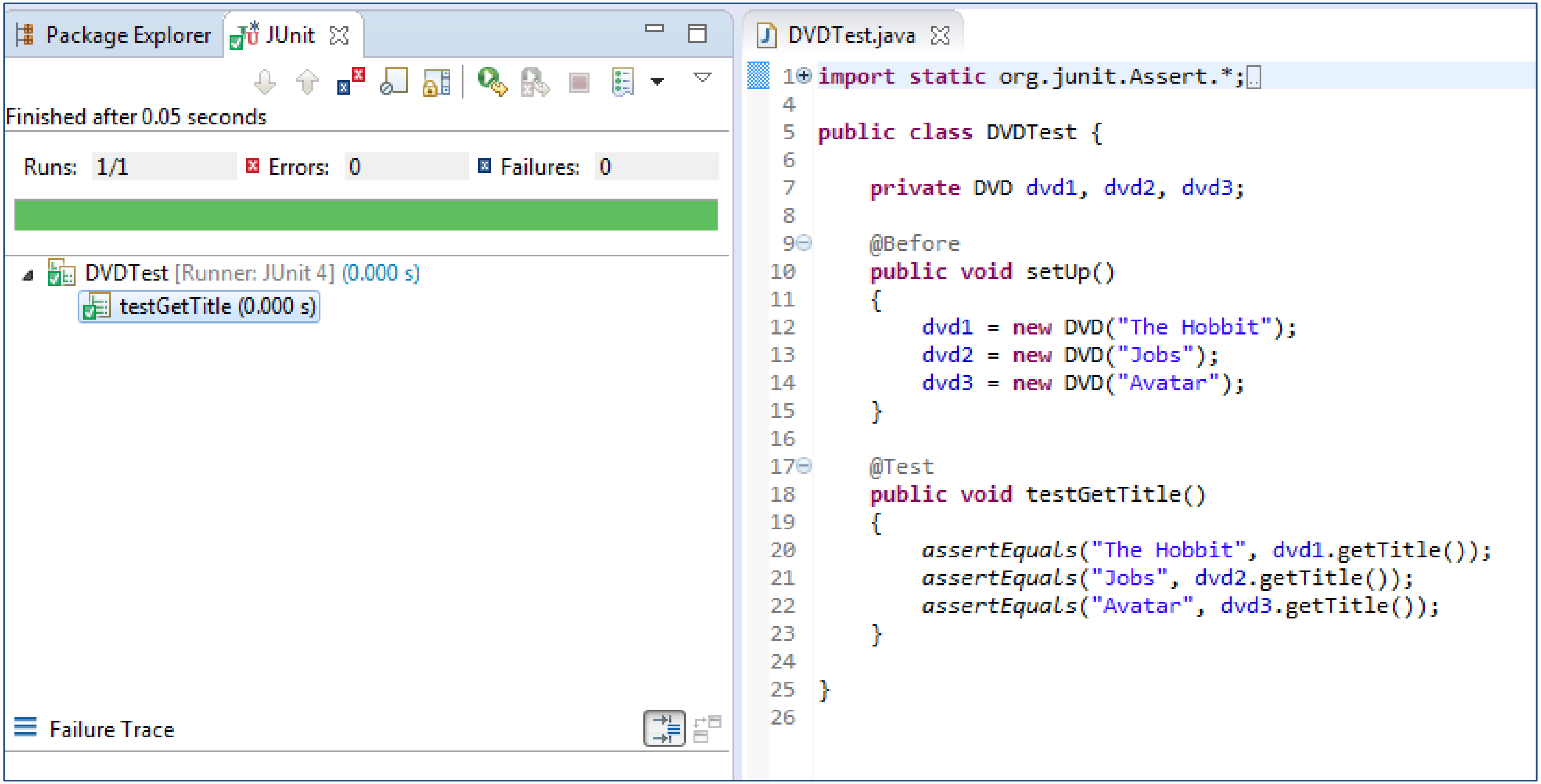
Test Bar indicating failure.

Itemised tests and their individual result.

When a failed test is selected in the above pane, the reason for failure appears here. Can be double clicked on for more information.

Testing getTitle()

```
public String getTitle() {
    return title;
}
```



Finished after 0.05 seconds

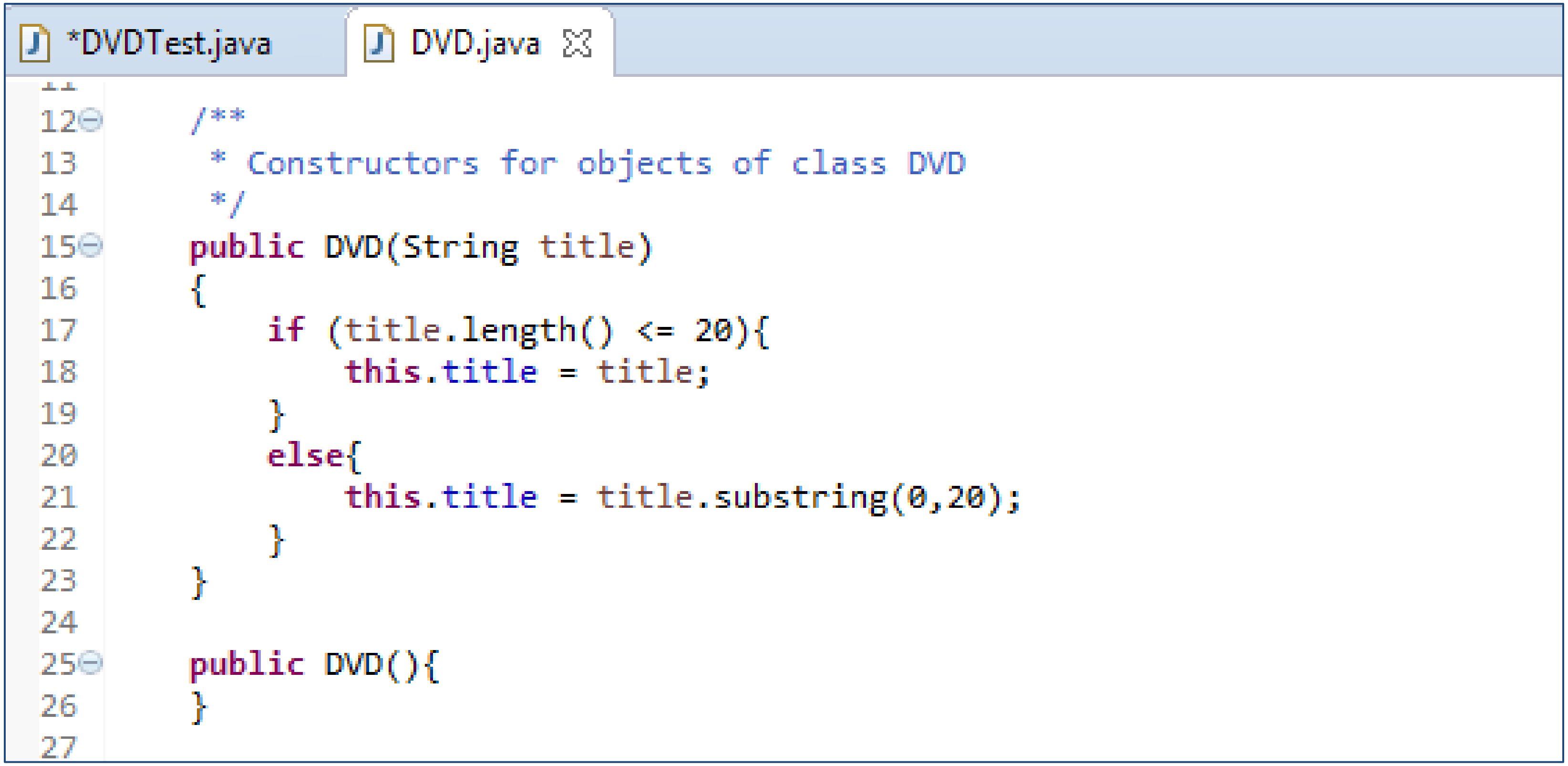
Runs: 1/1 ☒ Errors: 0 ☒ Failures: 0

DVDTest [Runner: JUnit 4] (0.000 s)
testGetTitle (0.000 s)

Failure Trace

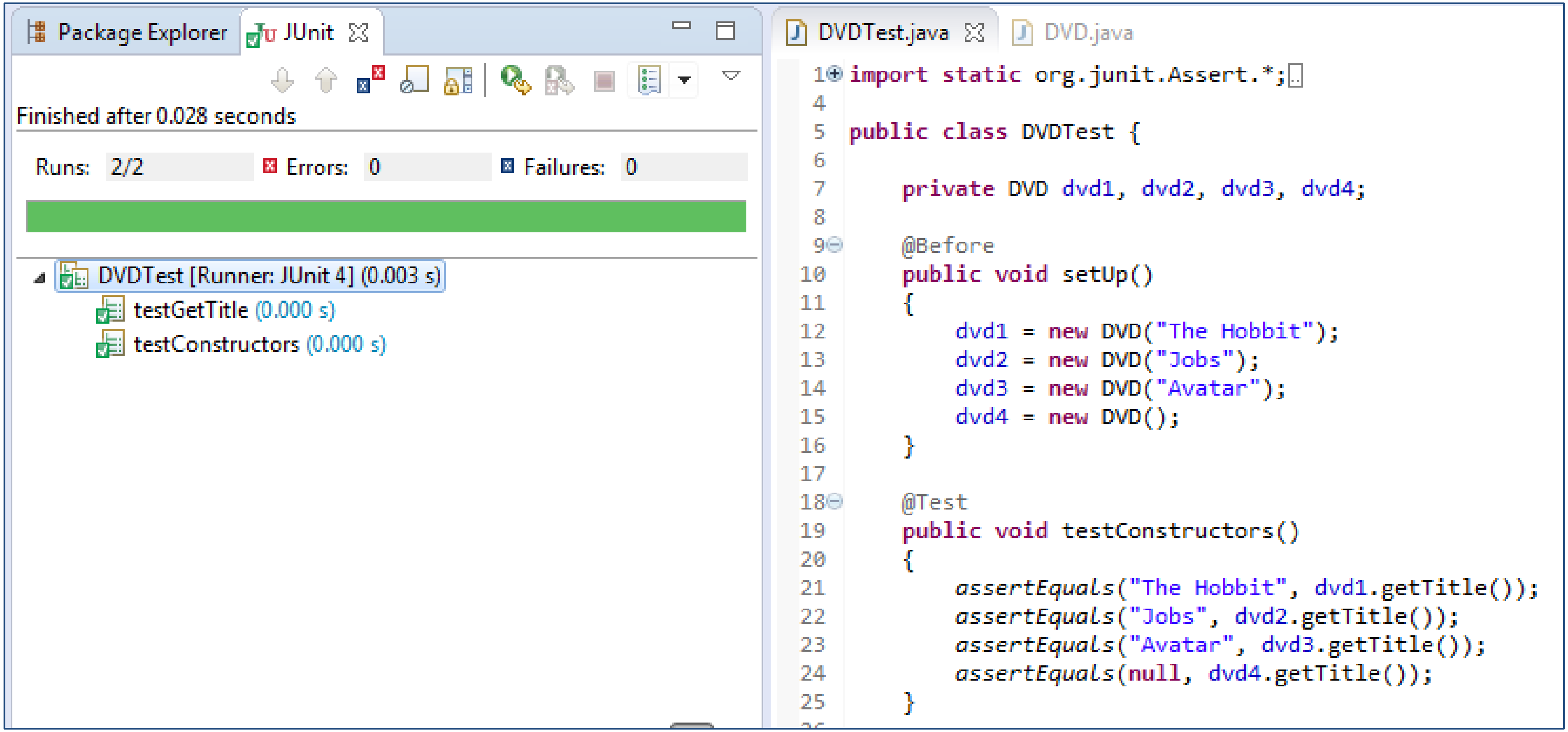
```
import static org.junit.Assert.*;

public class DVDTest {

    private DVD dvd1, dvd2, dvd3;

    @Before
    public void setUp()
    {
        dvd1 = new DVD("The Hobbit");
        dvd2 = new DVD("Jobs");
        dvd3 = new DVD("Avatar");
    }

    @Test
    public void testGetTitle()
    {
        assertEquals("The Hobbit", dvd1.getTitle());
        assertEquals("Jobs", dvd2.getTitle());
        assertEquals("Avatar", dvd3.getTitle());
    }

}

```

Testing Constructors (1)



```
    /**
     * Constructors for objects of class DVD
     */
    public DVD(String title)
    {
        if (title.length() <= 20){
            this.title = title;
        }
        else{
            this.title = title.substring(0,20);
        }
    }

    public DVD(){
    }

```

Testing Constructors (2)



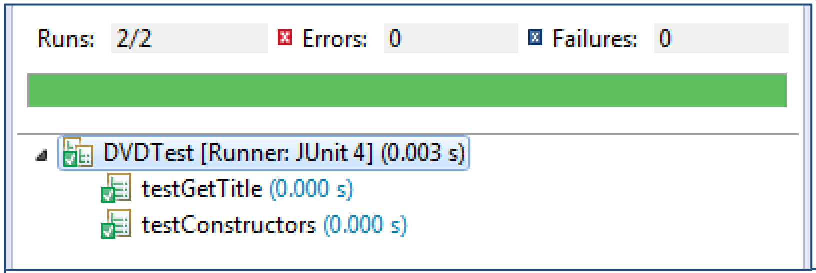
Finished after 0.028 seconds

```
Runs: 2/2    Errors: 0    Failures: 0
```

```
DVDTest [Runner: JUnit 4] (0.003 s)
    testGetTitle (0.000 s)
    testConstructors (0.000 s)
```

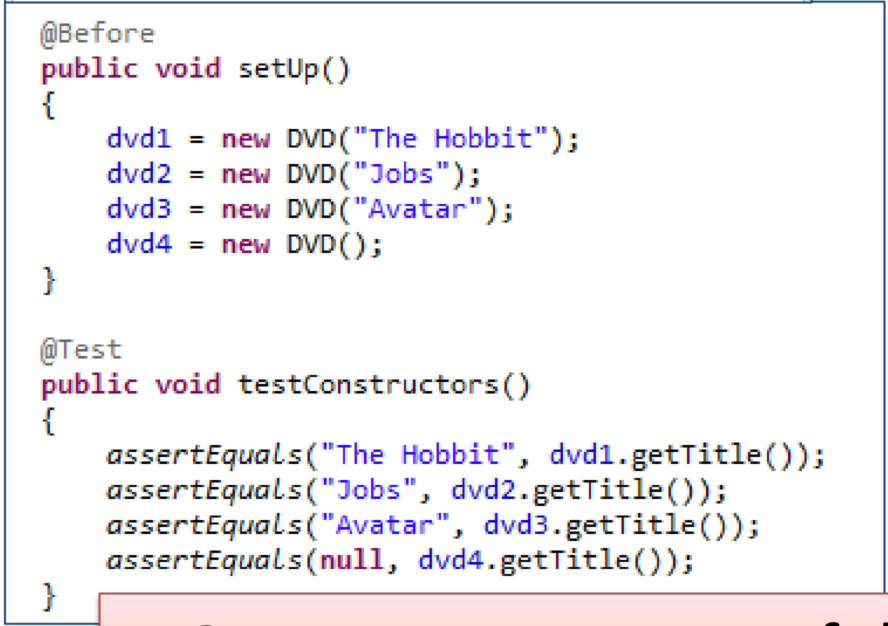
```java
import static org.junit.Assert.*;

public class DVDTest {

    private DVD dvd1, dvd2, dvd3, dvd4;

    @Before
    public void setUp()
    {
        dvd1 = new DVD("The Hobbit");
        dvd2 = new DVD("Jobs");
        dvd3 = new DVD("Avatar");
        dvd4 = new DVD();
    }

    @Test
    public void testConstructors()
    {
        assertEquals("The Hobbit", dvd1.getTitle());
        assertEquals("Jobs", dvd2.getTitle());
        assertEquals("Avatar", dvd3.getTitle());
        assertEquals(null, dvd4.getTitle());
    }
```

Testing Constructors (3)

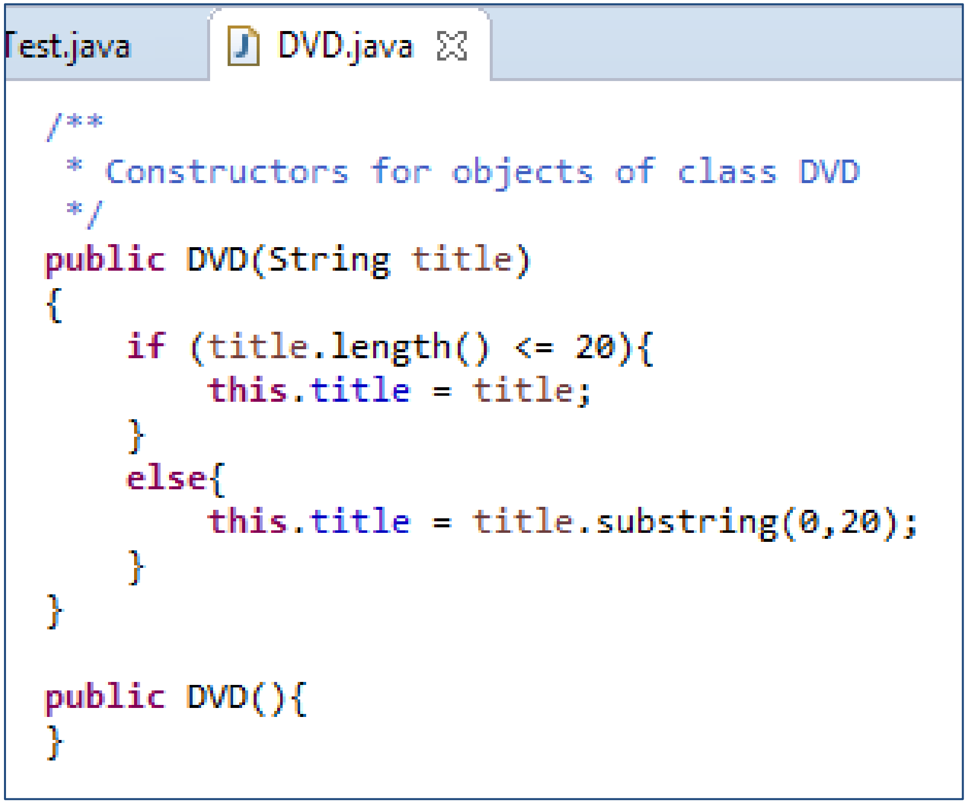


Runs: 2/2 Errors: 0 Failures: 0

```
DVDTest [Runner: JUnit 4] (0.003 s)
    testGetTitle (0.000 s)
    testConstructors (0.000 s)
```



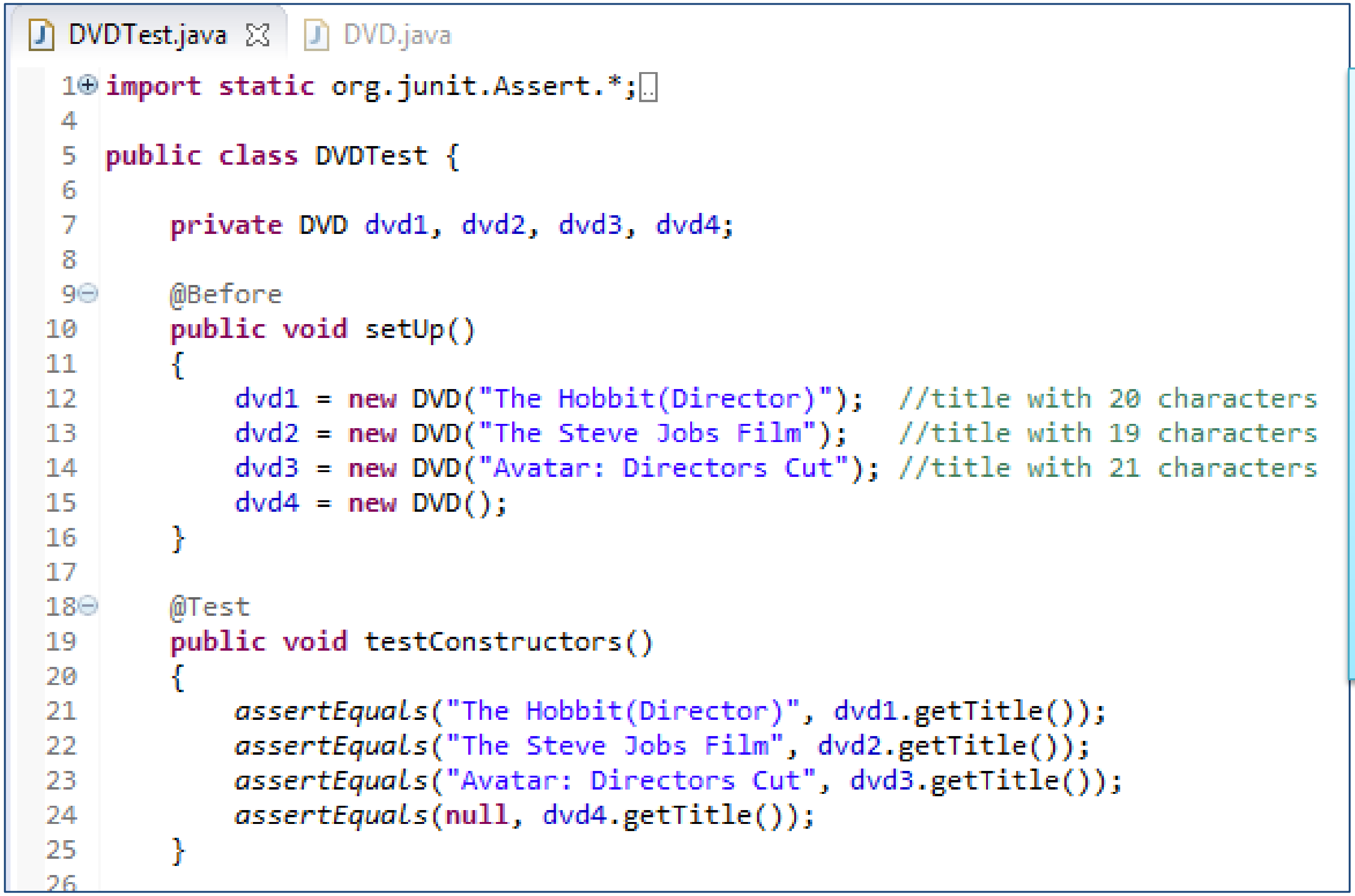
```java
@Before
public void setUp()
{
    dvd1 = new DVD("The Hobbit");
    dvd2 = new DVD("Jobs");
    dvd3 = new DVD("Avatar");
    dvd4 = new DVD();
}

@Test
public void testConstructors()
{
    assertEquals("The Hobbit", dvd1.getTitle());
    assertEquals("Jobs", dvd2.getTitle());
    assertEquals("Avatar", dvd3.getTitle());
    assertEquals(null, dvd4.getTitle());
}
```



```java
/**
 * Constructors for objects of class DVD
 */
public DVD(String title)
{
    if (title.length() <= 20){
        this.title = title;
    }
    else{
        this.title = title.substring(0,20);
    }
}

public DVD(){
}
```

Our tests are successful, BUT have we tested every possible path in the constructors?

Testing Constructors (4)



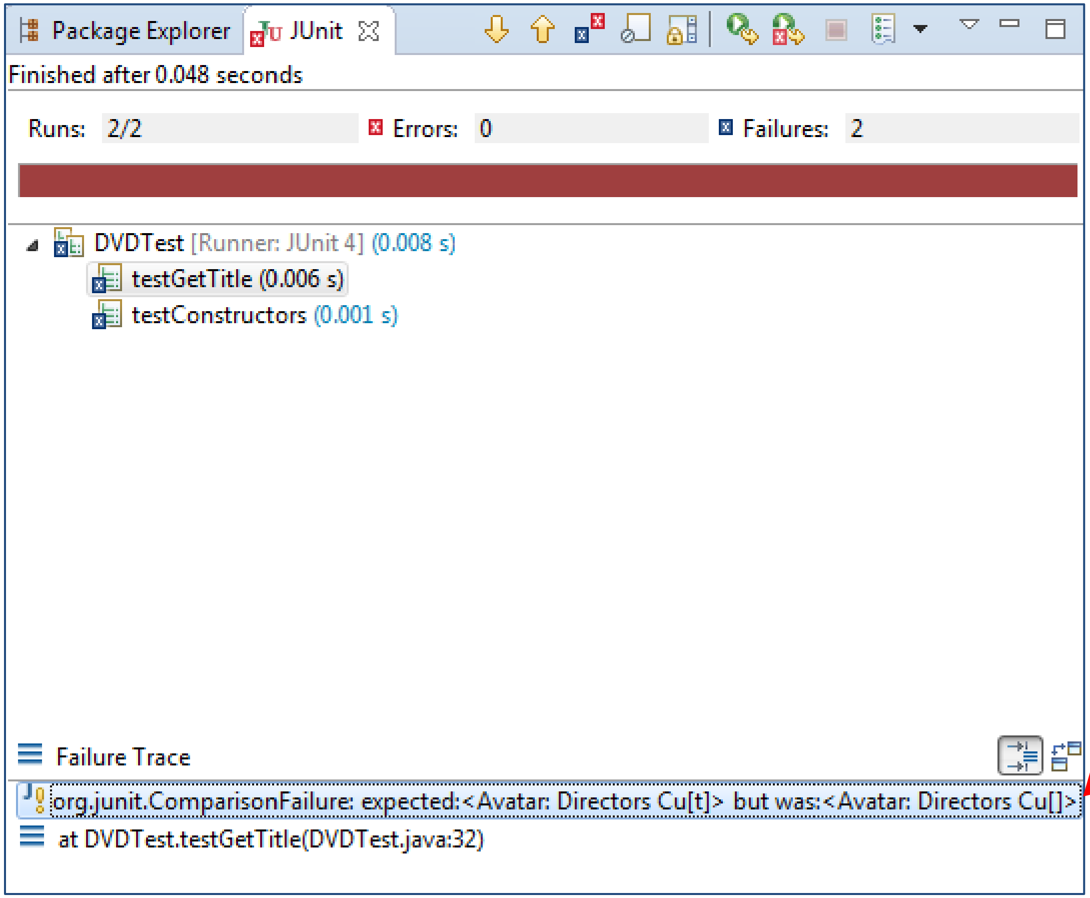
```java
import static org.junit.Assert.*;

public class DVDTest {

    private DVD dvd1, dvd2, dvd3, dvd4;

    @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
        dvd4 = new DVD();
    }

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

What do you think of this approach?

Better?

Will this test be successful?

Testing Constructors (5)



Finished after 0.048 seconds

Runs: 2/2 Errors: 0 Failures: 2

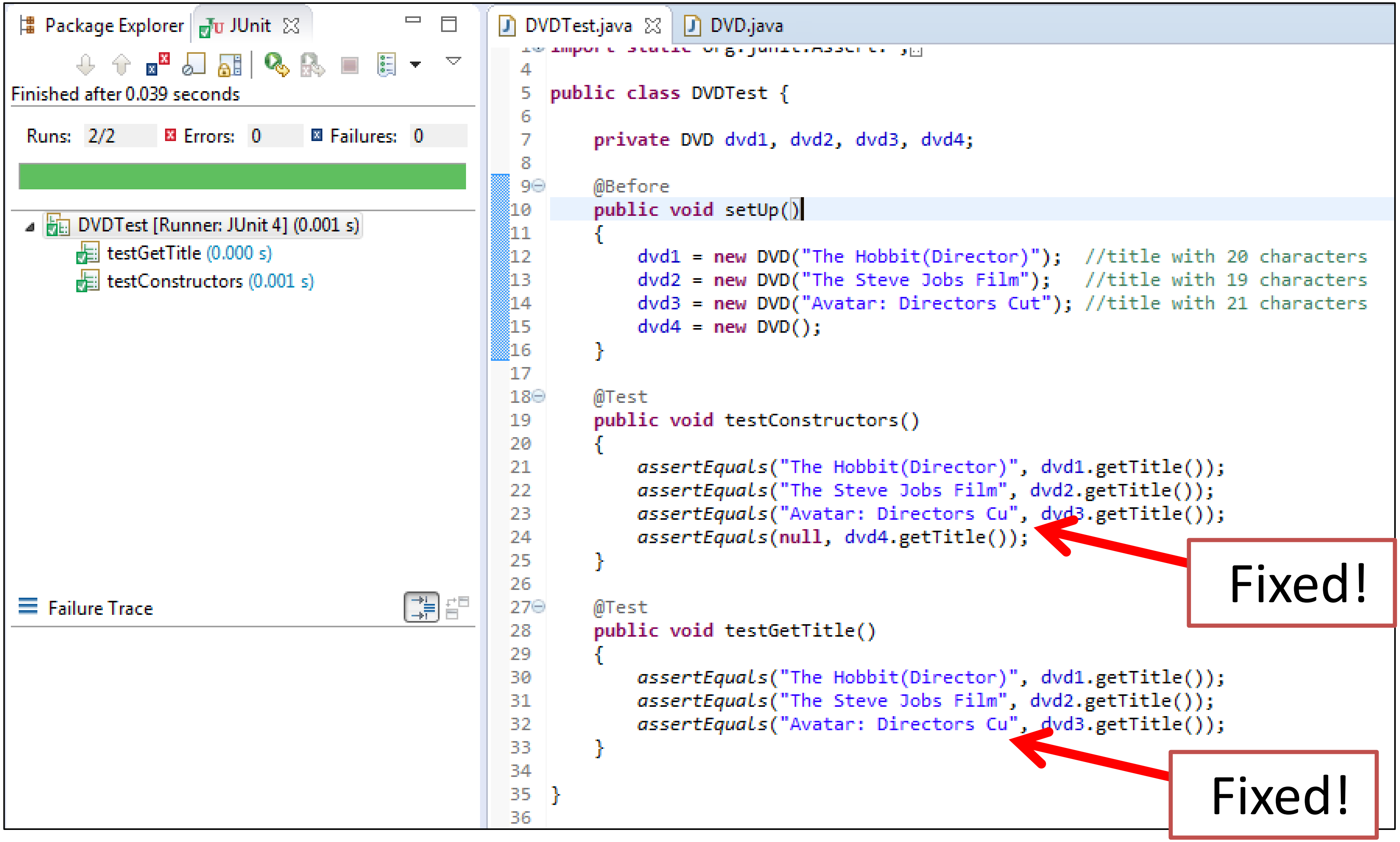
```
DVDTest [Runner: JUnit 4] (0.008 s)
    testGetTitle (0.006 s)
    testConstructors (0.001 s)
```

Failure Trace

```
org.junit.ComparisonFailure: expected:<Avatar: Directors Cu[t]> but was:<Avatar: Directors Cu[]>
at DVDTest.testGetTitle(DVDTest.java:32)
```

Can you see what went wrong?

Testing Constructors (6)



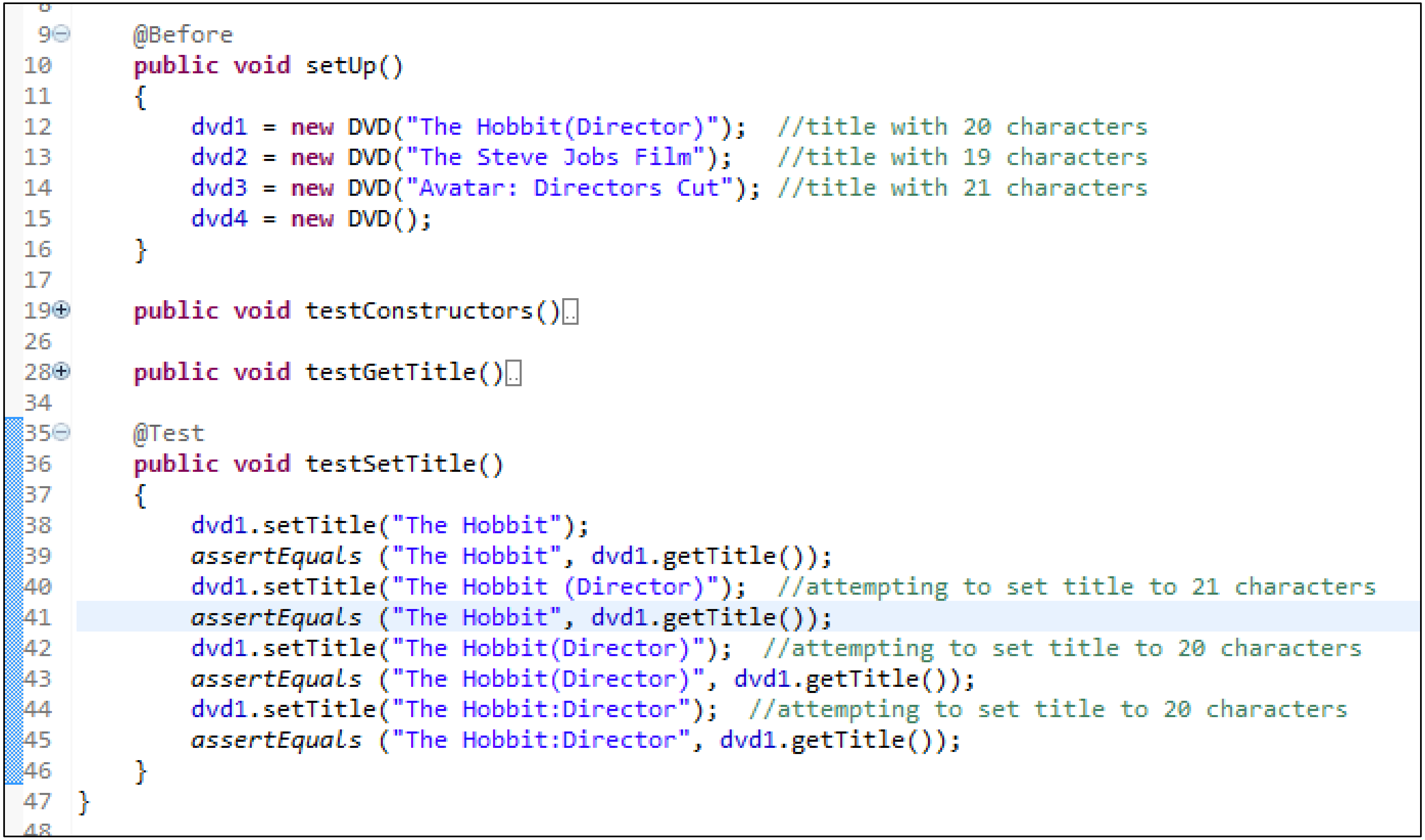
```
Finished after 0.039 seconds

Runs: 2/2    Errors: 0    Failures: 0

DVDTest [Runner: JUnit 4] (0.001 s)
    testGetTitle (0.000 s)
    testConstructors (0.001 s)
```

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

public class DVDTest {

    private DVD dvd1, dvd2, dvd3, dvd4;

    @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
        dvd4 = new DVD();
    }

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

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

}
```

Fixed!

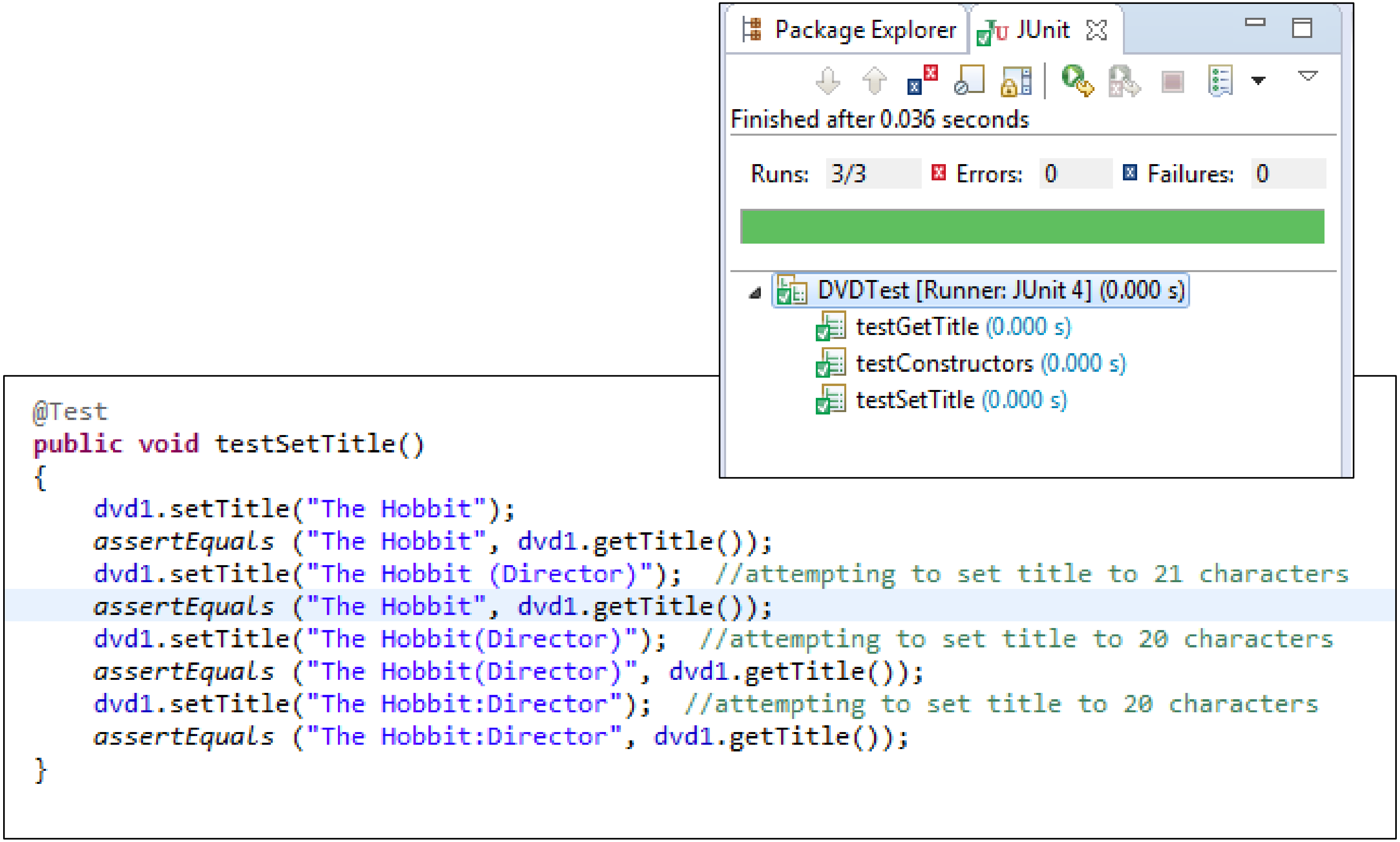
Fixed!

Testing setTitle method (1)



```
    @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
        dvd4 = new DVD();
    }

    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", 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());
    }
}
```

Testing setTitle method (2)



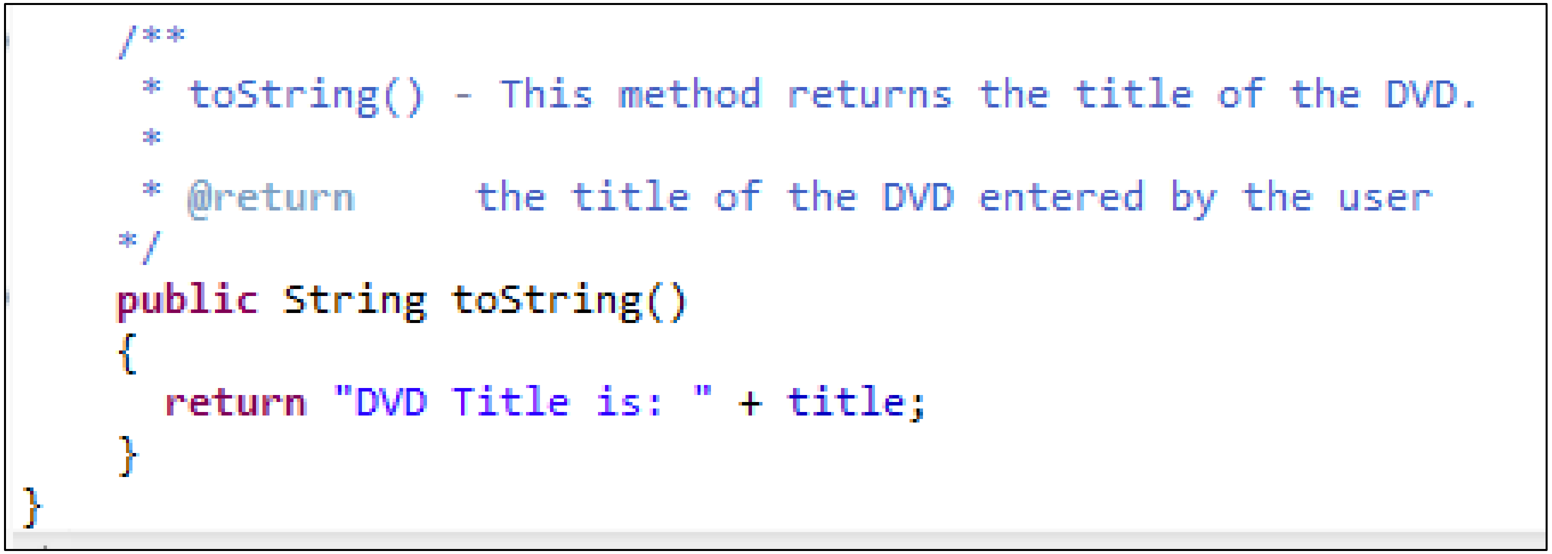
Finished after 0.036 seconds

Runs: 3/3 ☒ Errors: 0 ☒ Failures: 0

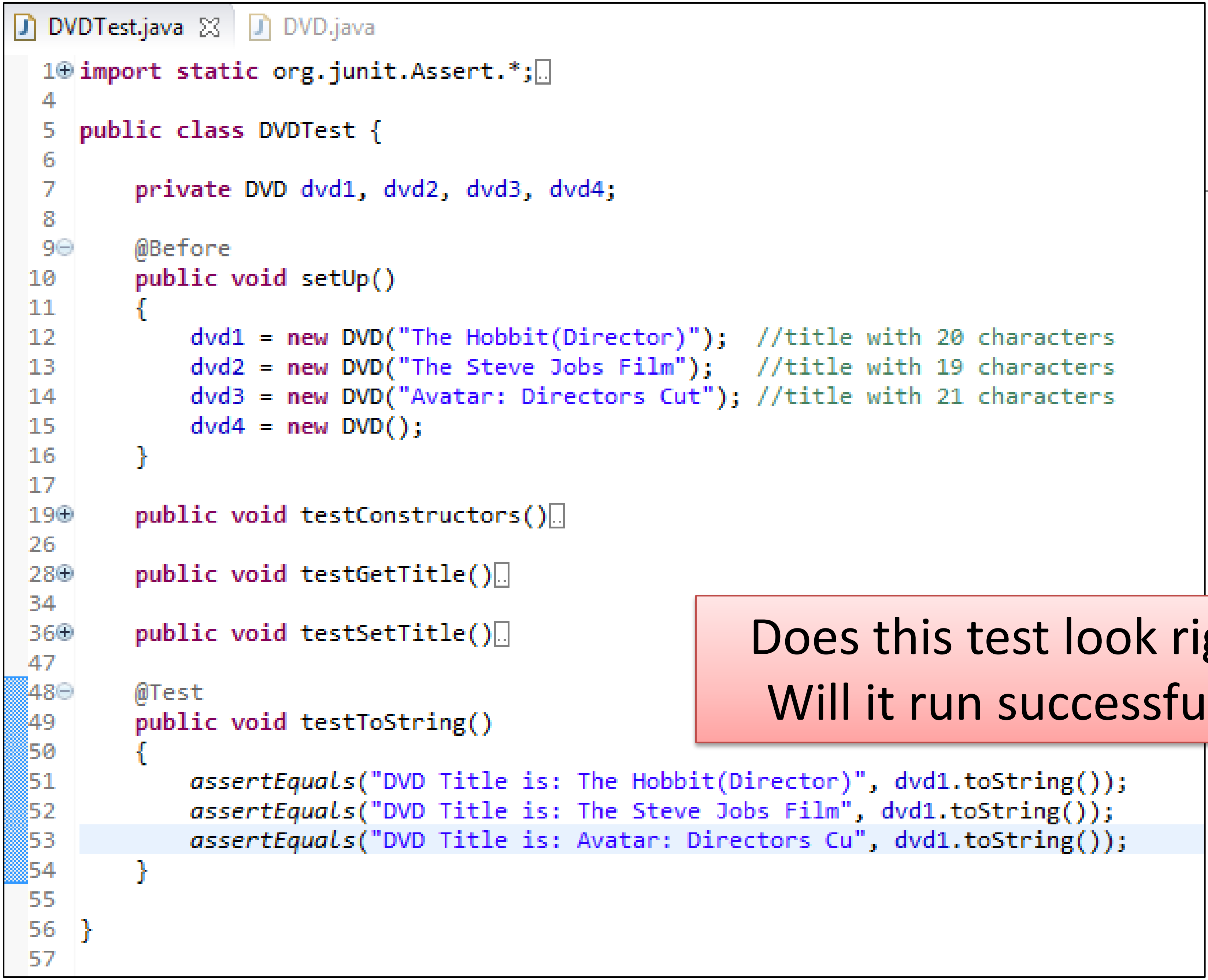
- DVDTest [Runner: JUnit 4] (0.000 s)
  - testGetTitle (0.000 s)
  - testConstructors (0.000 s)
  - testSetTitle (0.000 s)

```
@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", 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());
}
```

Testing toString method (1)



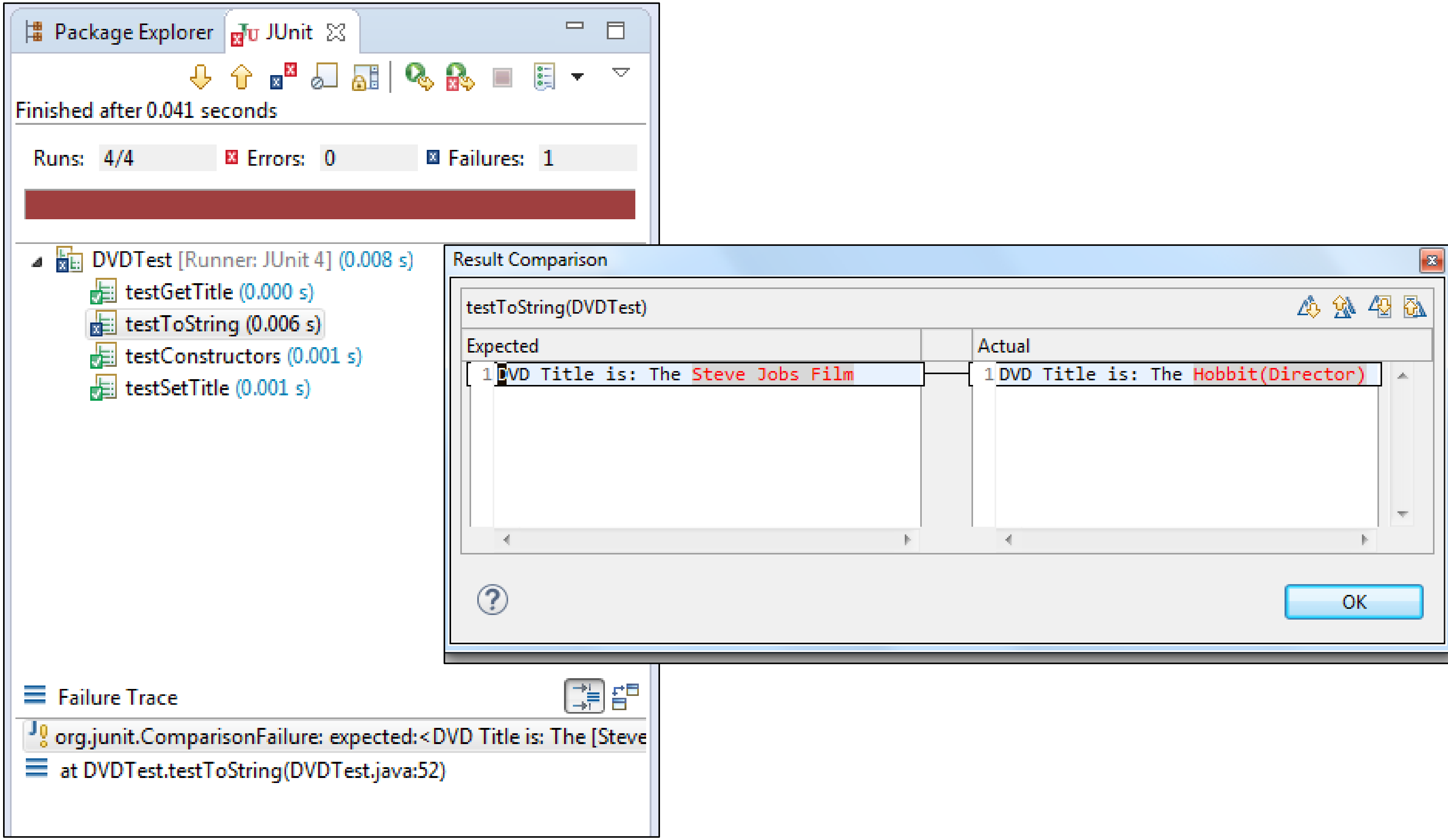
```
    /**
     * toString() - This method returns the title of the DVD.
     *
     * @return     the title of the DVD entered by the user
    */
    public String toString()
    {
      return "DVD Title is: " + title;
    }
}
```



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

public class DVDTest {

    private DVD dvd1, dvd2, dvd3, dvd4;

    @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
        dvd4 = new DVD();
    }

    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", dvd1.toString());
        assertEquals("DVD Title is: Avatar: Directors Cu", dvd1.toString());
    }

}
```

Does this test look right?
Will it run successfully?

Testing toString method (3)



Finished after 0.041 seconds

Runs: 4/4 Errors: 0 Failures: 1

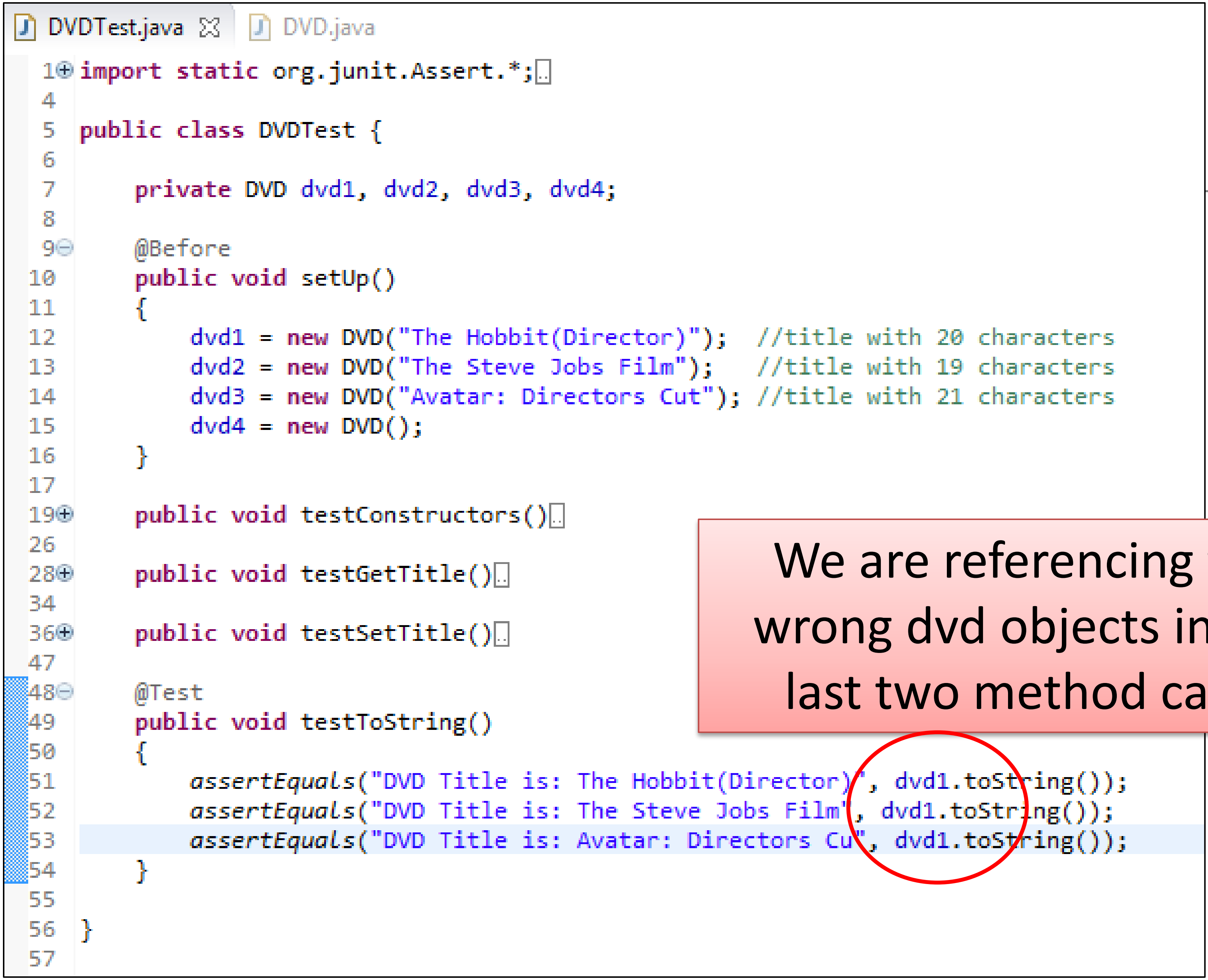
```
DVDTest [Runner: JUnit 4] (0.008 s)
    testGetTitle (0.000 s)
    testToString (0.006 s)
    testConstructors (0.001 s)
    testSetTitle (0.001 s)
```

testToString(DVDTest)

Expected	Actual
1 DVD Title is: The Steve Jobs Film	1 DVD Title is: The Hobbit(Director)

Failure Trace

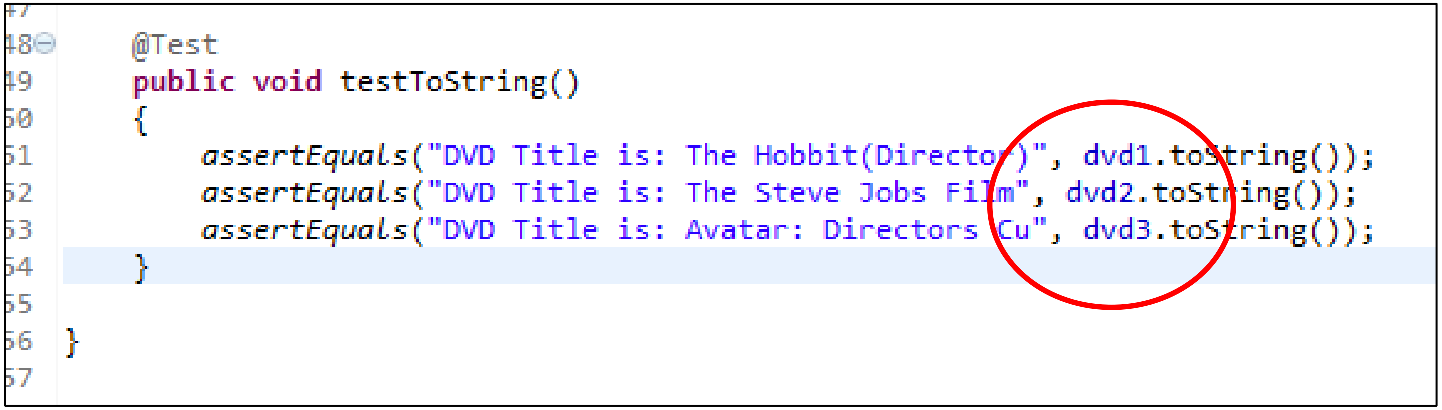
```
org.junit.ComparisonFailure: expected:<DVD Title is: The [Steve
at DVDTest.testToString(DVDTest.java:52)
```



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

public class DVDTest {

    private DVD dvd1, dvd2, dvd3, dvd4;

    @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
        dvd4 = new DVD();
    }

    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", dvd1.toString());
        assertEquals("DVD Title is: Avatar: Directors Cu", dvd1.toString());
    }

}
```

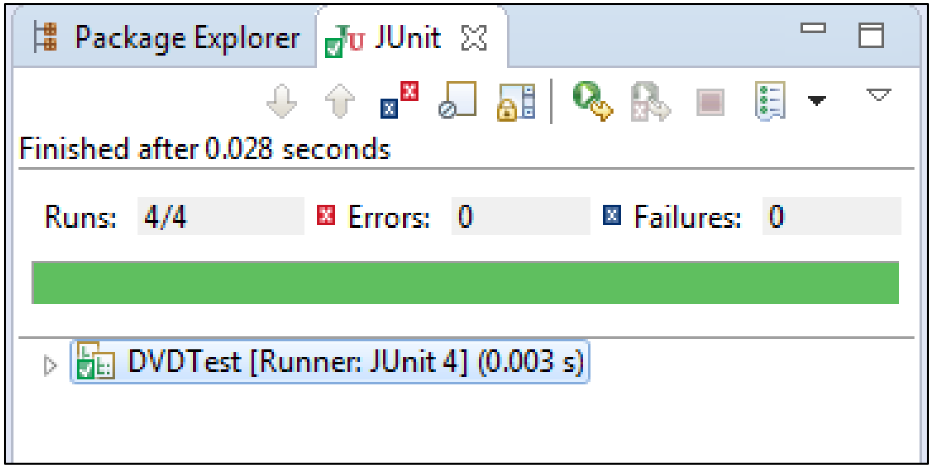
We are referencing the wrong dvd objects in the last two method calls.

Testing toString method (5)



```
    @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());
    }

}
```

We are now referencing the correct dvd objects, so our test is now successful.



```
Finished after 0.028 seconds

Runs: 4/4    Errors: 0    Failures: 0

DVDTest [Runner: JUnit 4] (0.003 s)
```

We have now successfully automated the testing of the entire DVD class

```
public class DVD
{
    private String title;

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

    public DVD(){
    }
```

```
    public void setTitle(String title){
        if (title.length() <= 20){
             this.title = title;
        }
    }

    public String getTitle(){
          return title;
    }

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

Any Questions?