

# Test Driven Development

## Using JUnit to Test DVD.java

---

Produced      Dr. Siobhán Drohan  
by:            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){
        setTitle(title);
    }

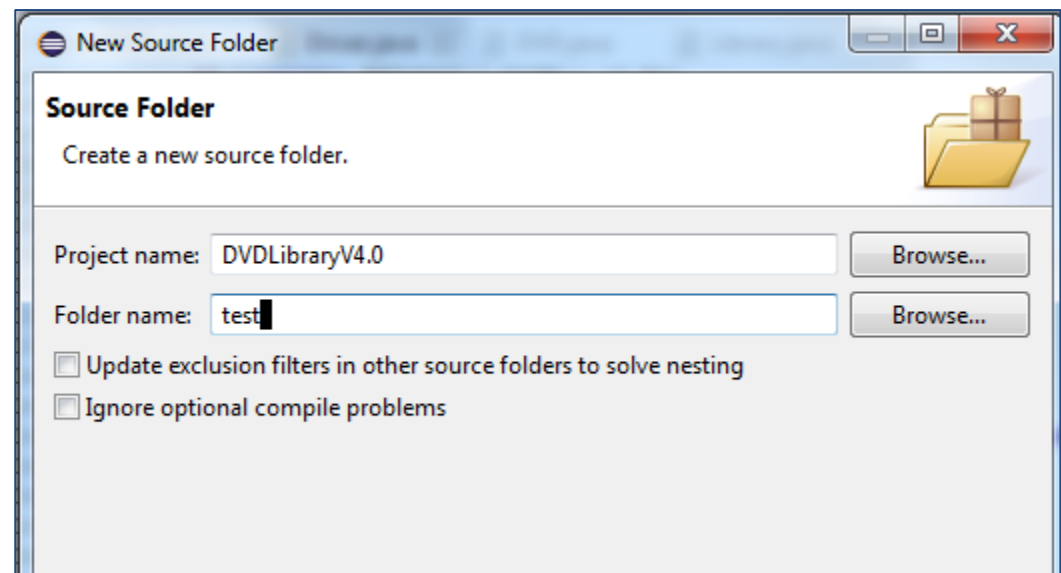
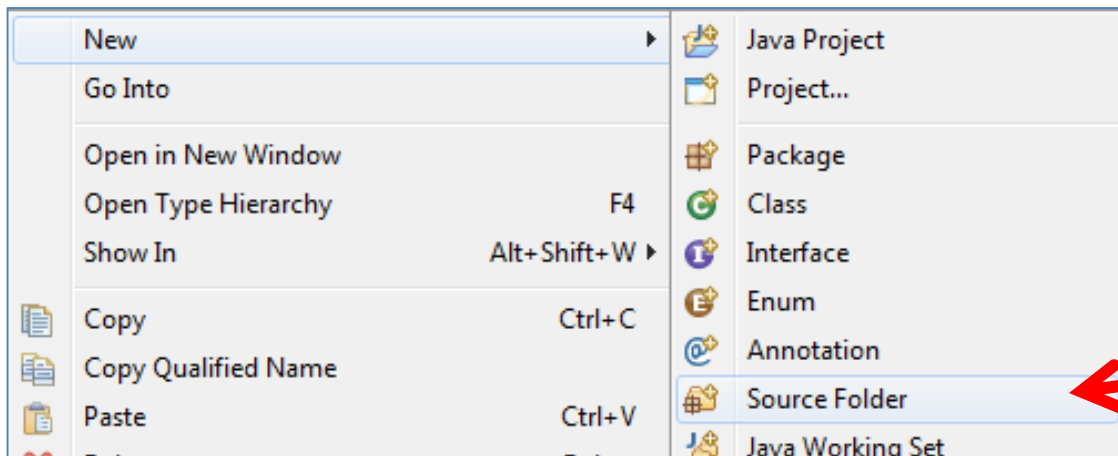
    public DVD(){
    }

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

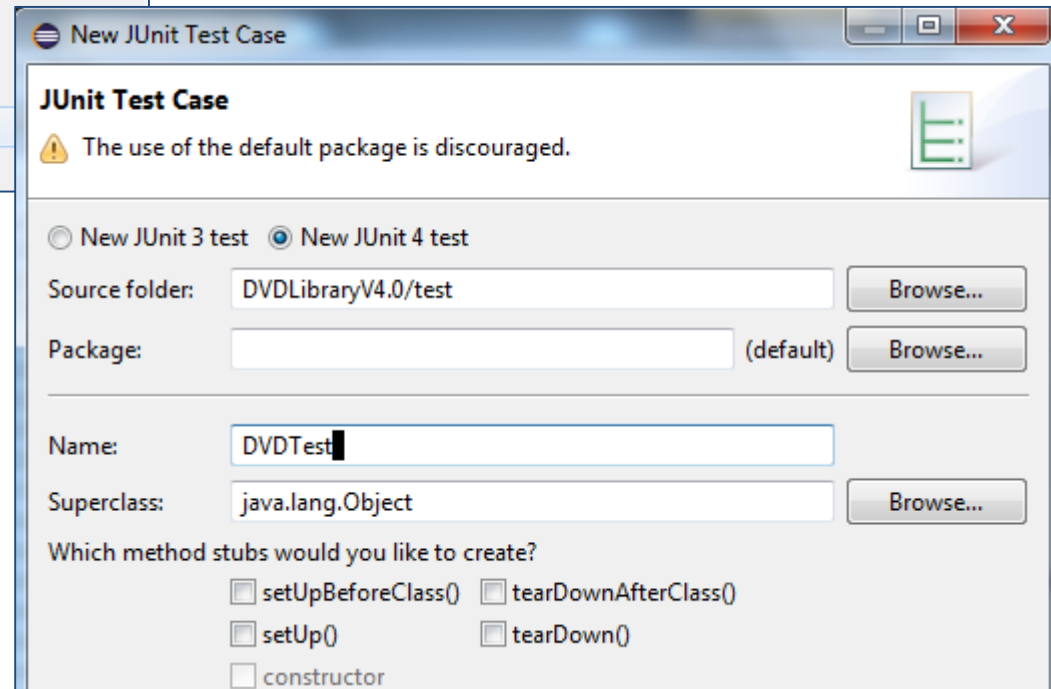
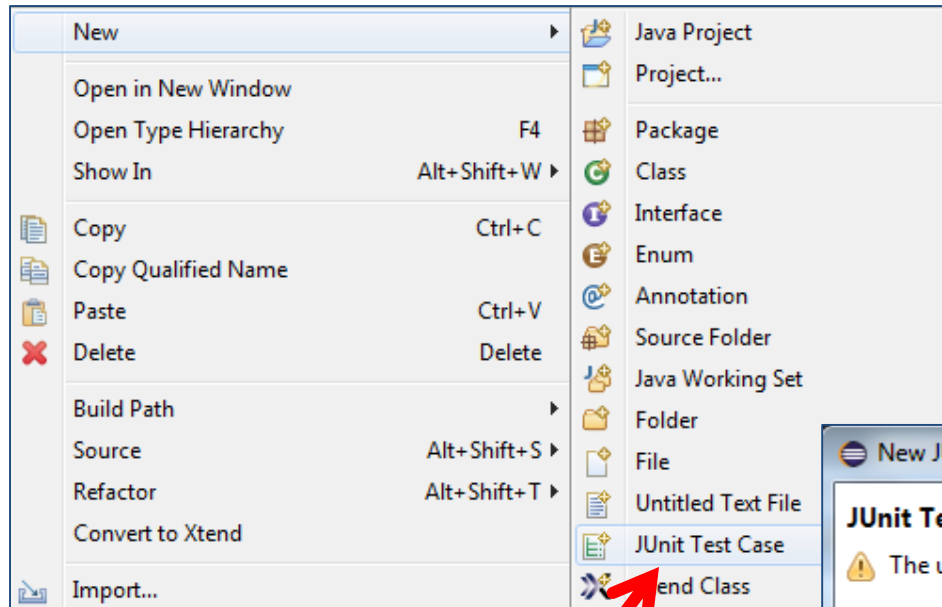
```
    public void setTitle(String title){
        if (title.length() <= 20){
            this.title = title
        }
        else{
            this.title = title.substring(0,20);
        }
    }

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

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

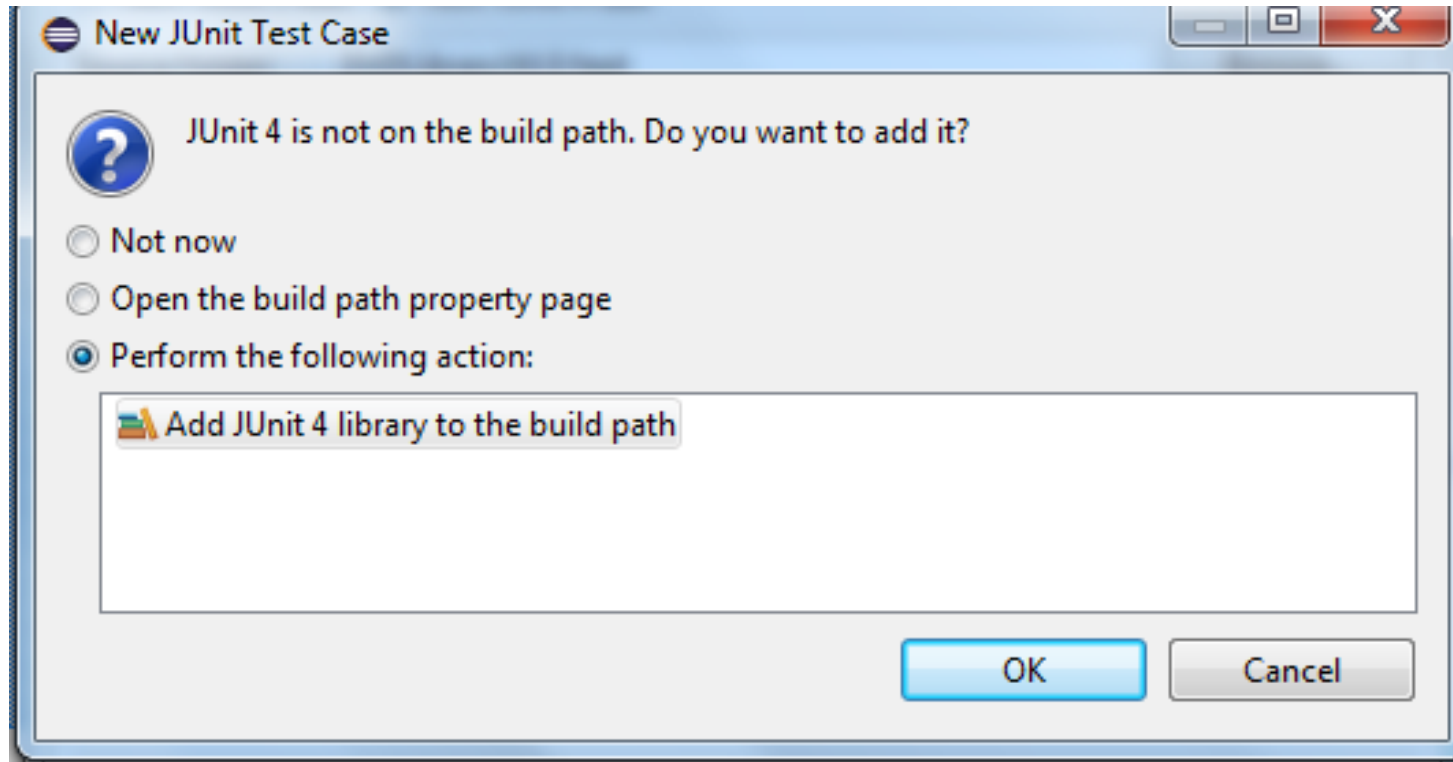


# Within this “test” folder, create a new JUnit Test Case



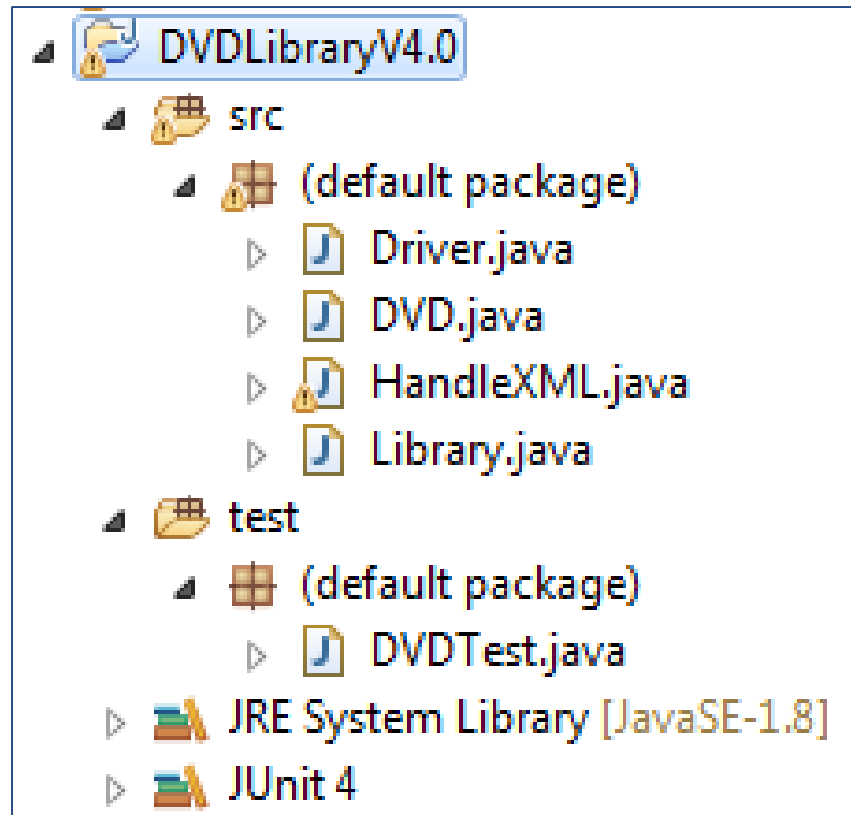
You may be asked to add JUnit 4 to your build path → Click OK.

---



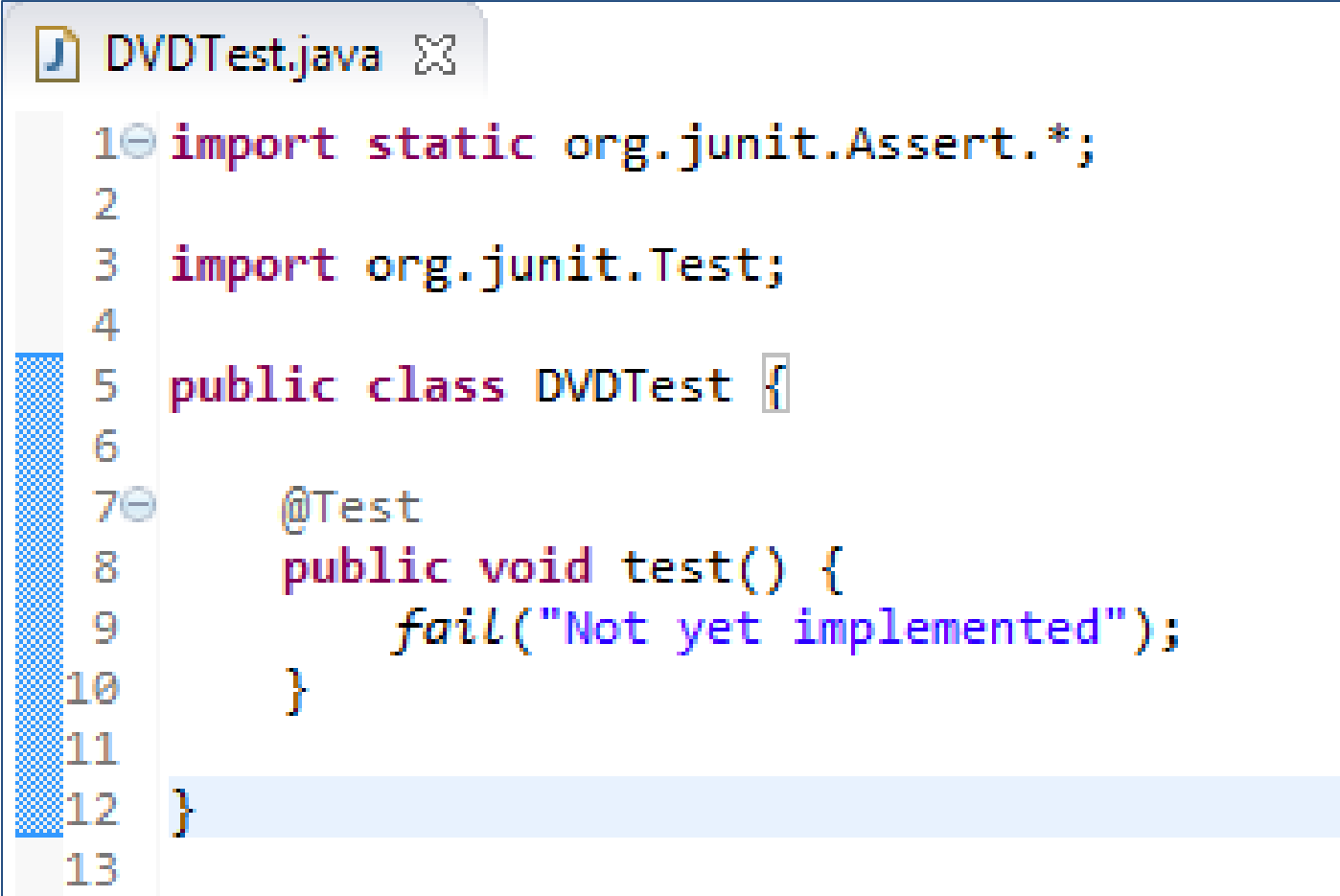
# Revised Folder Structure

---



# The generated DVDTest.java

---



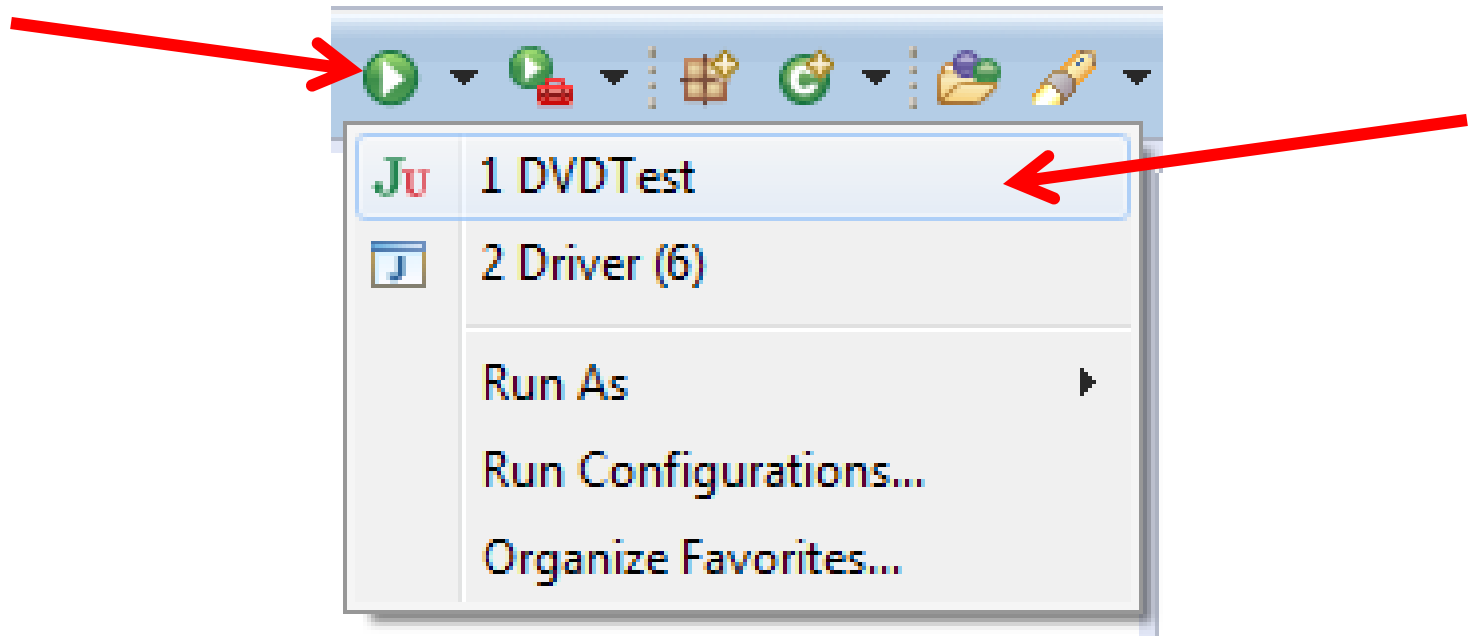
The screenshot shows a code editor window titled "DVDTest.java". The code is as follows:

```
1 import static org.junit.Assert.*;
2
3 import org.junit.Test;
4
5 public class DVDTest {
6
7     @Test
8     public void test() {
9         fail("Not yet implemented");
10    }
11
12 }
13
```



# Running DVDTest.java

---



# JUnit Results Pane (1)

The screenshot displays an IDE interface with the JUnit Results Pane on the left and the source code editor on the right.

**JUnit Results Pane (Left):**

- Package Explorer: JUnit
- Finished after 0.067 seconds
- Runs: 1/1, Errors: 0, Failures: 1
- Test Results: DVDTest [Runner: JUnit 4] (0.003 s) with a sub-item test (0.003 s) marked as failed.
- Failure Trace:
  - java.lang.AssertionError: Not yet implemented
  - at DVDTest.test(DVDTest.java:9)

**Source Code Editor (Right):**

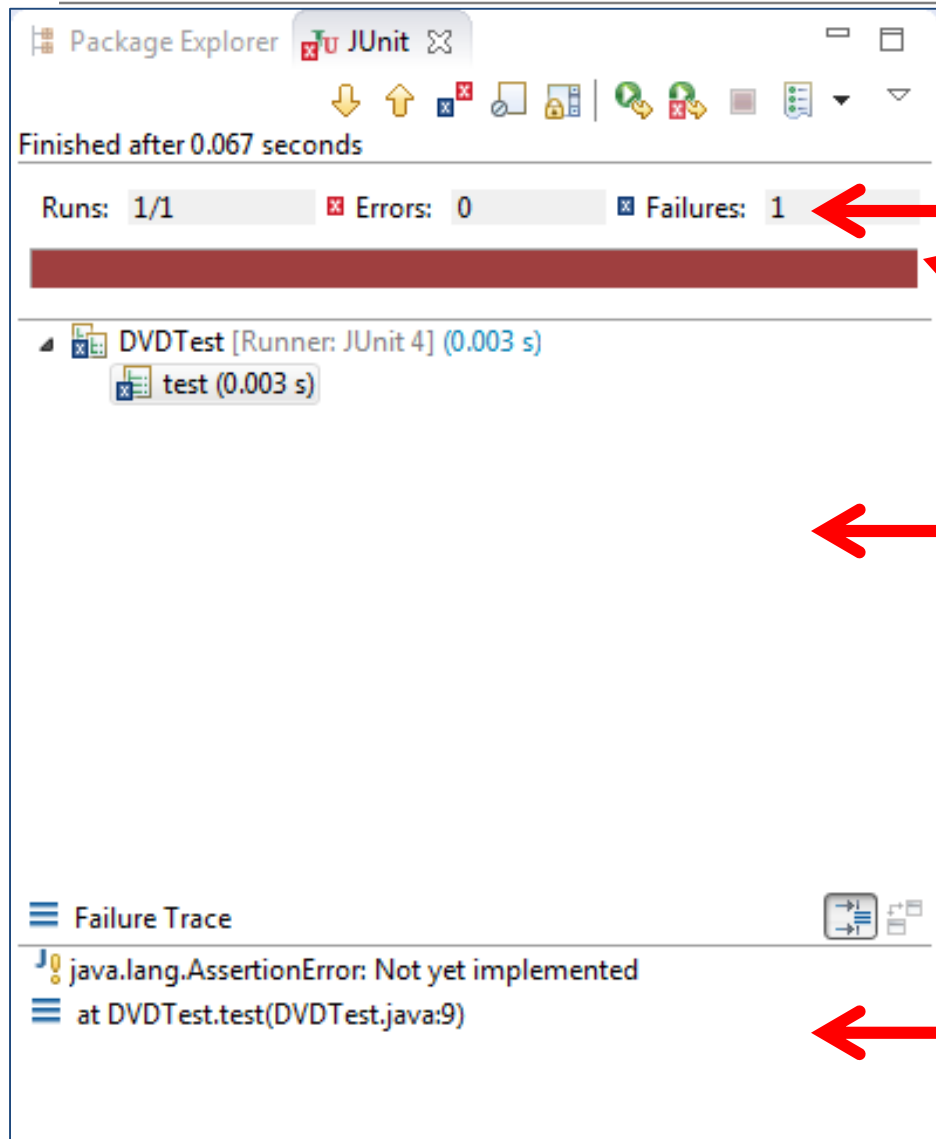
DVDTest.java

```
1 import static org.junit.Assert.*;
2
3 import org.junit.Test;
4
5 public class DVDTest {
6
7     @Test
8     public void test() {
9         fail("Not yet implemented");
10    }
11
12 }
13
```

**Console (Bottom Right):**

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

# JUnit Results Pane (2)



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;  
}
```

The screenshot displays an IDE interface with two main panels. The left panel shows the JUnit test runner results, indicating a successful execution of the `testGetTitle` method. The right panel shows the source code of `DVDTest.java`, which includes a `setUp` method for initializing DVD objects and a `testGetTitle` method that uses `assertEquals` to verify the titles.

**JUnit Runner Results:**

- Package Explorer: JUnit
- Finished after 0.05 seconds
- Runs: 1/1
- Errors: 0
- Failures: 0
- Test Results: DVDTest [Runner: JUnit 4] (0.000 s)
  - testGetTitle (0.000 s)
- Failure Trace

**Source Code (DVDTest.java):**

```
1 import static org.junit.Assert.*;
2
3
4
5 public class DVDTest {
6
7     private DVD dvd1, dvd2, dvd3;
8
9     @Before
10    public void setUp()
11    {
12        dvd1 = new DVD("The Hobbit");
13        dvd2 = new DVD("Jobs");
14        dvd3 = new DVD("Avatar");
15    }
16
17    @Test
18    public void testGetTitle()
19    {
20        assertEquals("The Hobbit", dvd1.getTitle());
21        assertEquals("Jobs", dvd2.getTitle());
22        assertEquals("Avatar", dvd3.getTitle());
23    }
24
25 }
26
```

# Testing Constructors (1)

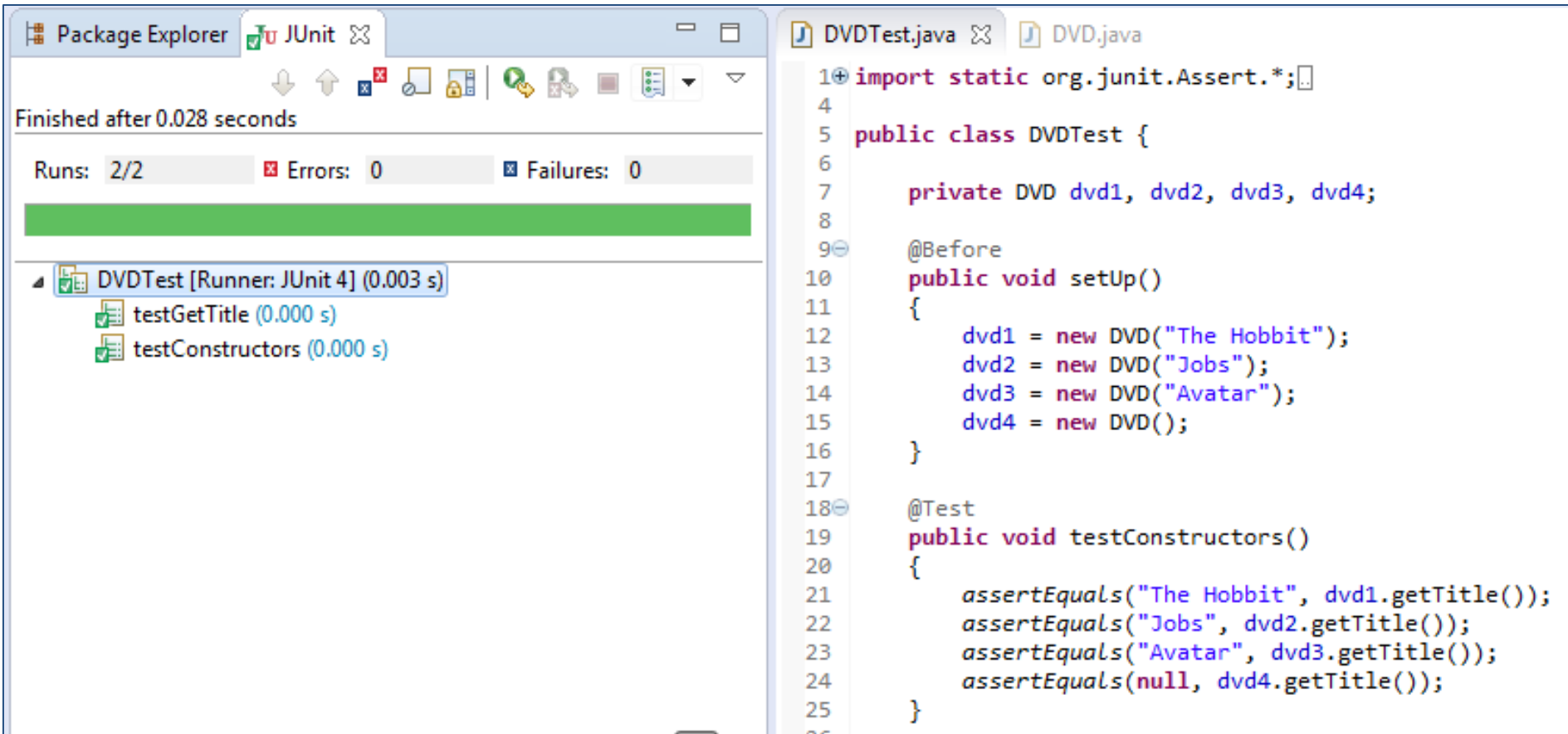
---

```
/**
 * Constructors for objects of class DVD
 */
public DVD(String title)
{
    setTitle(title);
}
```

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

public DVD(){
}
```

# Testing Constructors (2)



The screenshot displays an IDE interface with two main panels. The left panel shows the JUnit test results, and the right panel shows the source code for `DVDTest.java`.

**JUnit Test Results (Left Panel):**

- Package Explorer: JUnit
- Finished after 0.028 seconds
- Runs: 2/2, Errors: 0, Failures: 0
- Test Results:
  - DVDTest [Runner: JUnit 4] (0.003 s)
    - testGetTitle (0.000 s)
    - testConstructors (0.000 s)

**Source Code (Right Panel):**

```
1 import static org.junit.Assert.*;
4
5 public class DVDTest {
6
7     private DVD dvd1, dvd2, dvd3, dvd4;
8
9     @Before
10    public void setUp()
11    {
12        dvd1 = new DVD("The Hobbit");
13        dvd2 = new DVD("Jobs");
14        dvd3 = new DVD("Avatar");
15        dvd4 = new DVD();
16    }
17
18    @Test
19    public void testConstructors()
20    {
21        assertEquals("The Hobbit", dvd1.getTitle());
22        assertEquals("Jobs", dvd2.getTitle());
23        assertEquals("Avatar", dvd3.getTitle());
24        assertEquals(null, dvd4.getTitle());
25    }
26 }
```

# 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)

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

```
/**
 * Constructors for objects of class DVD
 */
public DVD(String title)
{
    setTitle(title);
}
```

```
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)

```
DVDTest.java  DVD.java

1+ import static org.junit.Assert.*;
4
5 public class DVDTest {
6
7     private DVD dvd1, dvd2, dvd3, dvd4;
8
9-    @Before
10    public void setUp()
11    {
12        dvd1 = new DVD("The Hobbit(Director)"); //title with 20 characters
13        dvd2 = new DVD("The Steve Jobs Film"); //title with 19 characters
14        dvd3 = new DVD("Avatar: Directors Cut"); //title with 21 characters
15        dvd4 = new DVD();
16    }
17
18-    @Test
19    public void testConstructors()
20    {
21        assertEquals("The Hobbit(Director)", dvd1.getTitle());
22        assertEquals("The Steve Jobs Film", dvd2.getTitle());
23        assertEquals("Avatar: Directors Cut", dvd3.getTitle());
24        assertEquals(null, dvd4.getTitle());
25    }
26
```

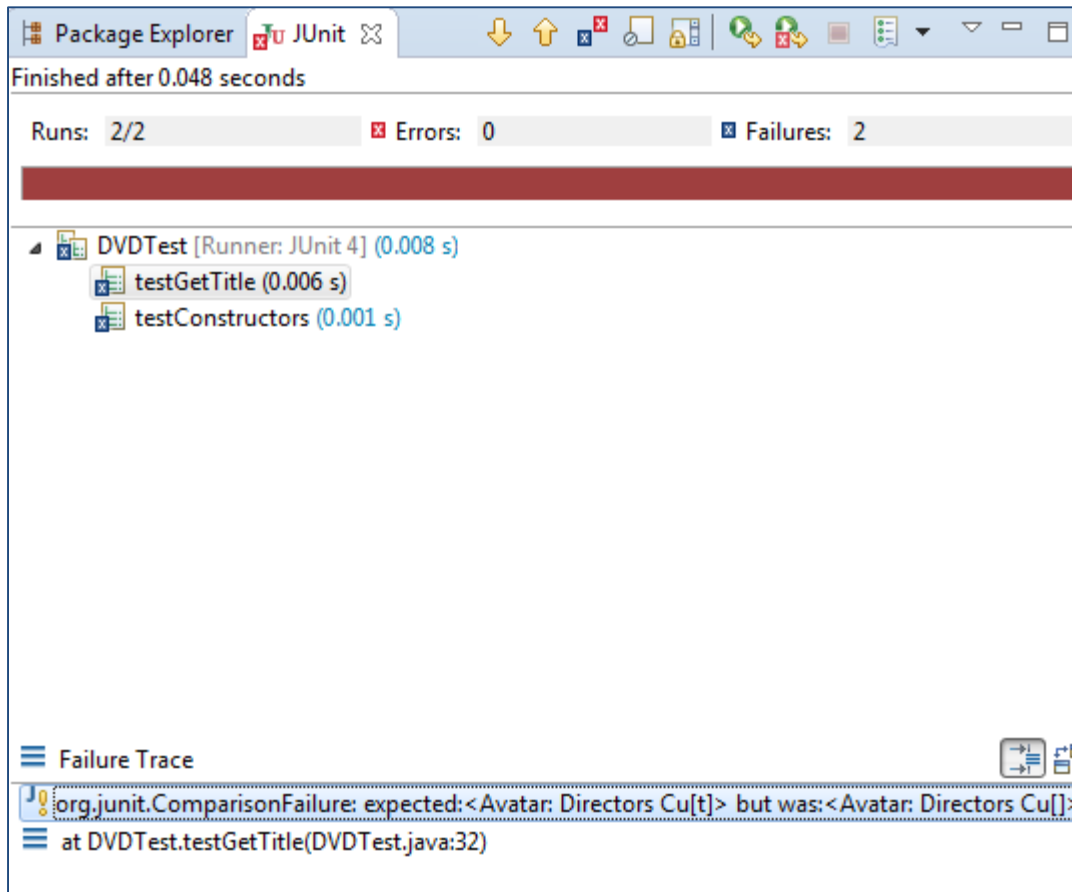
What do you think of this approach?

Better?

Will this test be successful?



# Testing Constructors (5)



Can you see  
what went  
wrong?

# Testing Constructors (6)

Package Explorer | JUnit

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)

Failure Trace

```
DVDTest.java | DVD.java
import static org.junit.Assert.*;

4
5 public class DVDTest {
6
7     private DVD dvd1, dvd2, dvd3, dvd4;
8
9     @Before
10    public void setUp()
11    {
12        dvd1 = new DVD("The Hobbit(Director)"); //title with 20 characters
13        dvd2 = new DVD("The Steve Jobs Film"); //title with 19 characters
14        dvd3 = new DVD("Avatar: Directors Cut"); //title with 21 characters
15        dvd4 = new DVD();
16    }
17
18    @Test
19    public void testConstructors()
20    {
21        assertEquals("The Hobbit(Director)", dvd1.getTitle());
22        assertEquals("The Steve Jobs Film", dvd2.getTitle());
23        assertEquals("Avatar: Directors Cu", dvd3.getTitle());
24        assertEquals(null, dvd4.getTitle());
25    }
26
27    @Test
28    public void testGetTitle()
29    {
30        assertEquals("The Hobbit(Director)", dvd1.getTitle());
31        assertEquals("The Steve Jobs Film", dvd2.getTitle());
32        assertEquals("Avatar: Directors Cu", dvd3.getTitle());
33    }
34
35 }
36
```

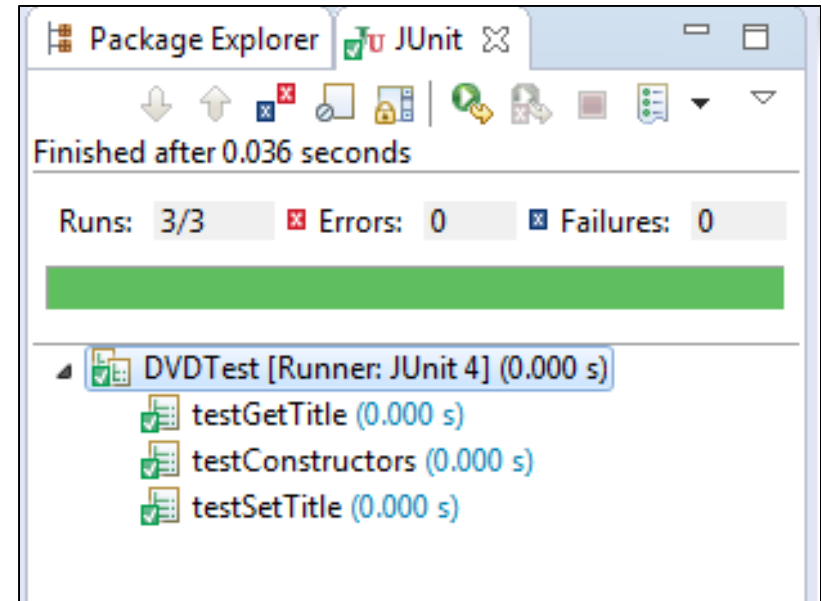
Fixed!

Fixed!

# Testing setTitle method (1)

```
DVDTest.java LibraryTest.java DVD.java
1 *import static org.junit.Assert.*;
2
3
4
5
6 public class DVDTest {
7
8     private DVD dvd1, dvd2, dvd3;
9
10    @Before
11    public void setUp() {
12        dvd1 = new DVD("The Hobbit(Director)"); //title with 20 characters
13        dvd2 = new DVD("The Steve Jobs Film"); //title with 19 characters
14        dvd3 = new DVD("Avatar: Directors Cut"); //title with 21 characters
15    }
16
17
18    public void tearDown() {}
19
20
21
22    public void testConstructors() {}
23
24
25
26
27
28
29    public void testGetTitle() {}
30
31
32
33
34
35    @Test
36    public void testSetTitle() {
37        dvd1.setTitle("The Hobbit");
38        assertEquals("The Hobbit", dvd1.getTitle());
39        dvd1.setTitle("The Hobbit (Director)"); //attempting to set title to 21 characters
40        assertEquals("The Hobbit (Director)", dvd1.getTitle());
41        dvd1.setTitle("The Hobbit(Director)"); //attempting to set title to 20 characters
42        assertEquals("The Hobbit(Director)", dvd1.getTitle());
43        dvd1.setTitle("The Hobbit:Director"); //attempting to set title to 20 characters
44        assertEquals("The Hobbit:Director", dvd1.getTitle());
45    }
46
47
48    public void testToString() {}
49
50
51
52
53
54 }
```

# Testing setTitle method (2)



```
35 @Test
36 public void testSetTitle() {
37     dvd1.setTitle("The Hobbit");
38     assertEquals("The Hobbit", dvd1.getTitle());
39     dvd1.setTitle("The Hobbit (Director)"); //attempting to set title to 21 characters
40     assertEquals("The Hobbit (Director", dvd1.getTitle());
41     dvd1.setTitle("The Hobbit(Director)"); //attempting to set title to 20 characters
42     assertEquals("The Hobbit(Director)", dvd1.getTitle());
43     dvd1.setTitle("The Hobbit:Director"); //attempting to set title to 20 characters
44     assertEquals("The Hobbit:Director", dvd1.getTitle());
45 }
46
```

# Testing toString method (1)

---

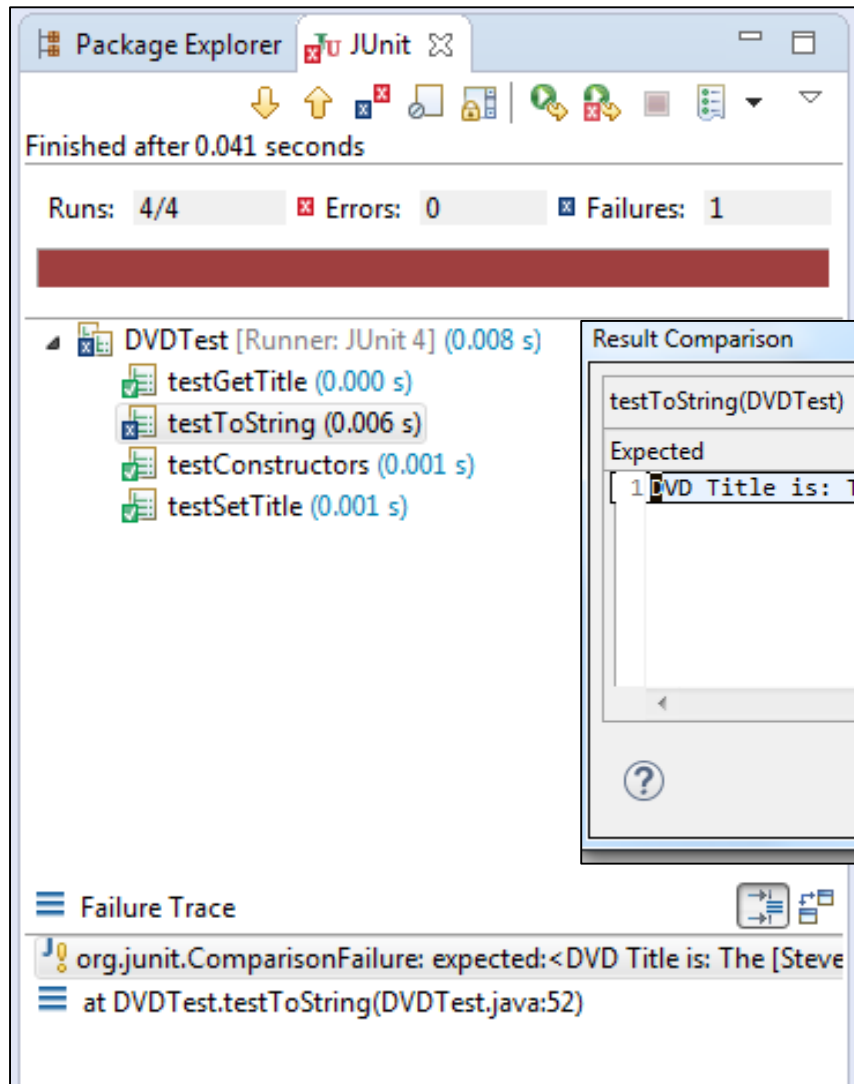
```
/**
 * toString() - This method returns the title of the DVD.
 *
 * @return      the title of the DVD entered by the user
 */
public String toString()
{
    return "DVD Title is: " + title;
}
}
```

DVDTest.java X DVD.java

```
1 import static org.junit.Assert.*;
4
5 public class DVDTest {
6
7     private DVD dvd1, dvd2, dvd3, dvd4;
8
9     @Before
10    public void setUp()
11    {
12        dvd1 = new DVD("The Hobbit(Director)"); //title with 20 characters
13        dvd2 = new DVD("The Steve Jobs Film"); //title with 19 characters
14        dvd3 = new DVD("Avatar: Directors Cut"); //title with 21 characters
15        dvd4 = new DVD();
16    }
17
19    public void testConstructors()
26
28    public void testGetTitle()
34
36    public void testSetTitle()
47
48    @Test
49    public void testToString()
50    {
51        assertEquals("DVD Title is: The Hobbit(Director)", dvd1.toString());
52        assertEquals("DVD Title is: The Steve Jobs Film", dvd1.toString());
53        assertEquals("DVD Title is: Avatar: Directors Cu", dvd1.toString());
54    }
55
56 }
57
```

Does this test look right?  
Will it run successfully?

# Testing toString method (3)



Package Explorer JUnit

Finished after 0.041 seconds

Runs: 4/4 Errors: 0 Failures: 1

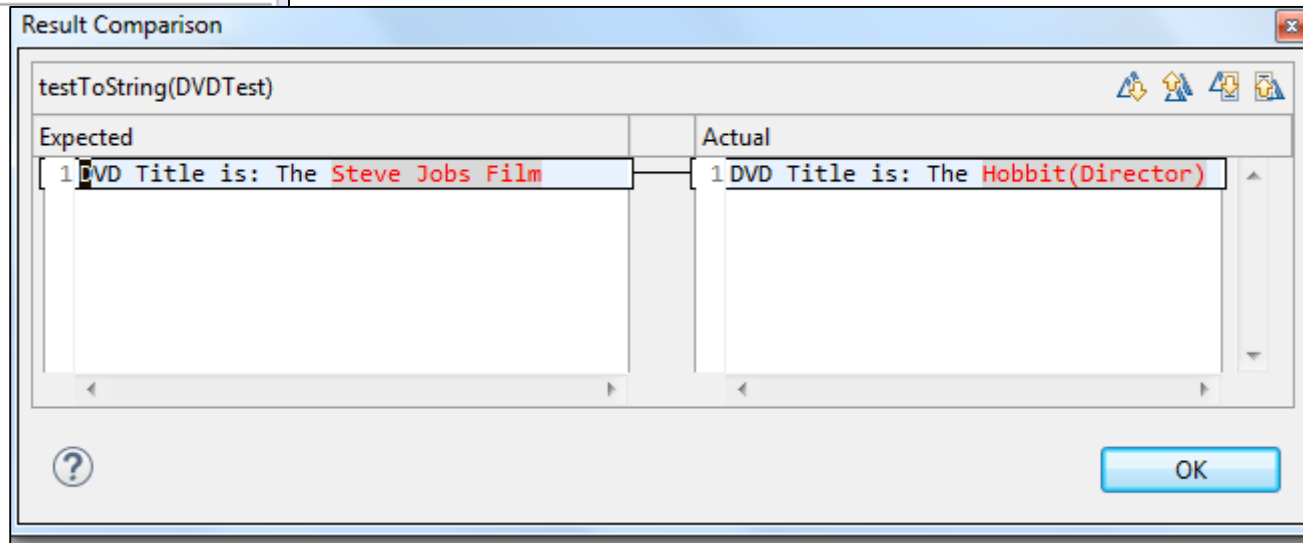
DVDTest [Runner: JUnit 4] (0.008 s)

- testGetTitle (0.000 s)
- testToString (0.006 s)
- testConstructors (0.001 s)
- testSetTitle (0.001 s)

Failure Trace

org.junit.ComparisonFailure: expected:<DVD Title is: The [Steve

at DVDTest.testToString(DVDTest.java:52)



Result Comparison

testToString(DVDTest)

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

OK

DVDTest.java X DVD.java

```
1 import static org.junit.Assert.*;
4
5 public class DVDTest {
6
7     private DVD dvd1, dvd2, dvd3, dvd4;
8
9     @Before
10    public void setUp()
11    {
12        dvd1 = new DVD("The Hobbit(Director)"); //title with 20 characters
13        dvd2 = new DVD("The Steve Jobs Film"); //title with 19 characters
14        dvd3 = new DVD("Avatar: Directors Cut"); //title with 21 characters
15        dvd4 = new DVD();
16    }
17
19    public void testConstructors()
26
28    public void testGetTitle()
34
36    public void testSetTitle()
47
48    @Test
49    public void testToString()
50    {
51        assertEquals("DVD Title is: The Hobbit(Director)", dvd1.toString());
52        assertEquals("DVD Title is: The Steve Jobs Film", dvd1.toString());
53        assertEquals("DVD Title is: Avatar: Directors Cu", dvd1.toString());
54    }
55
56 }
57
```

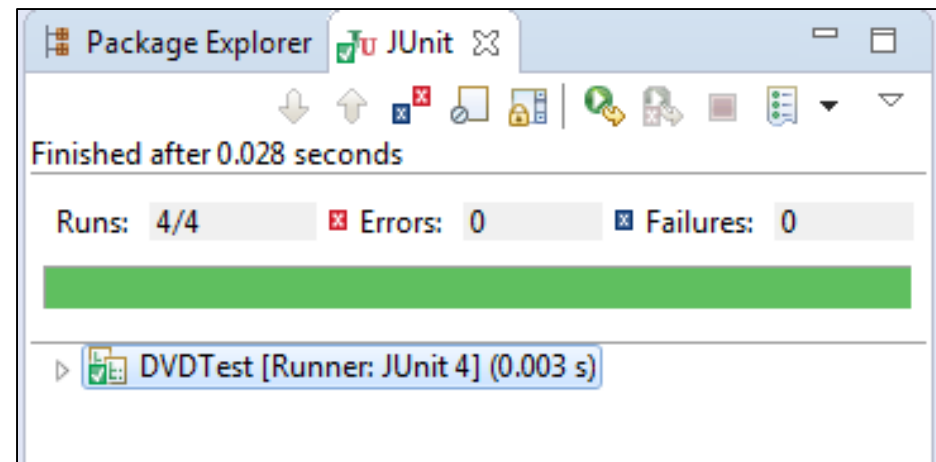
We are referencing the wrong dvd objects in the last two method calls.



# Testing toString method (5)

```
47  
48 @Test  
49 public void testToString()  
50 {  
51     assertEquals("DVD Title is: The Hobbit(Director)", dvd1.toString());  
52     assertEquals("DVD Title is: The Steve Jobs Film", dvd2.toString());  
53     assertEquals("DVD Title is: Avatar: Directors Cu", dvd3.toString());  
54 }  
55  
56 }  
57
```

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



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

---

```
public class DVD
{
    private String title;

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

    public DVD(){
    }

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

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

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

---

**Any  
Questions?**





Except where otherwise noted, this content is licensed under a Creative Commons Attribution-NonCommercial 3.0 License.

For more information, please see <http://creativecommons.org/licenses/by-nc/3.0/>



Waterford Institute of Technology  
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

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