

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

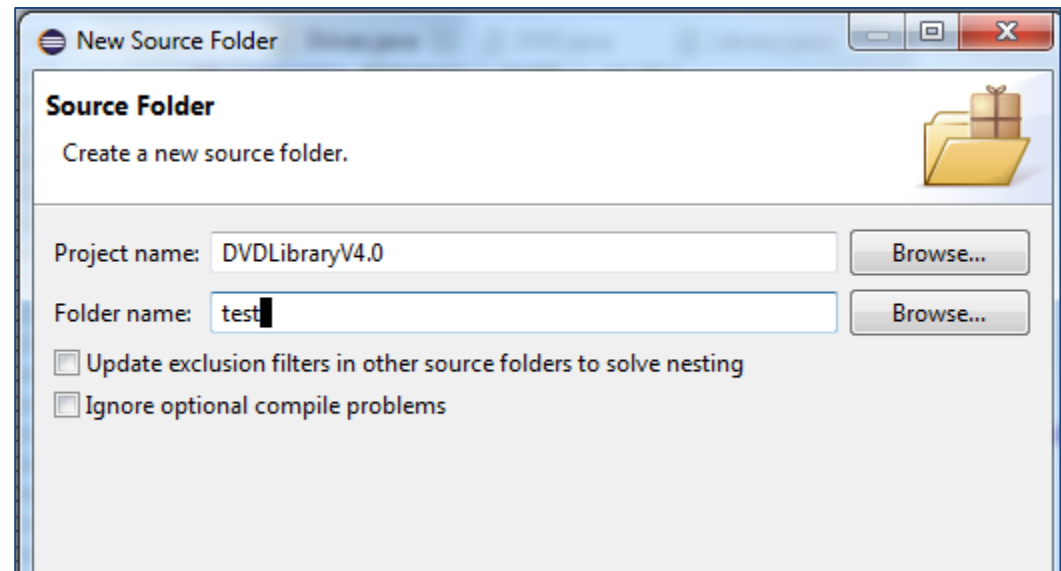
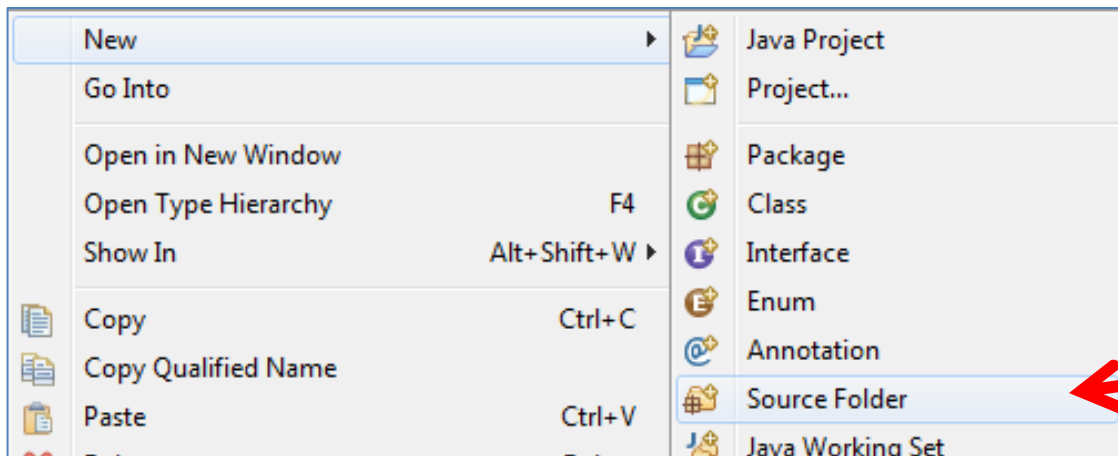
    public DVD(){
    }
```

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

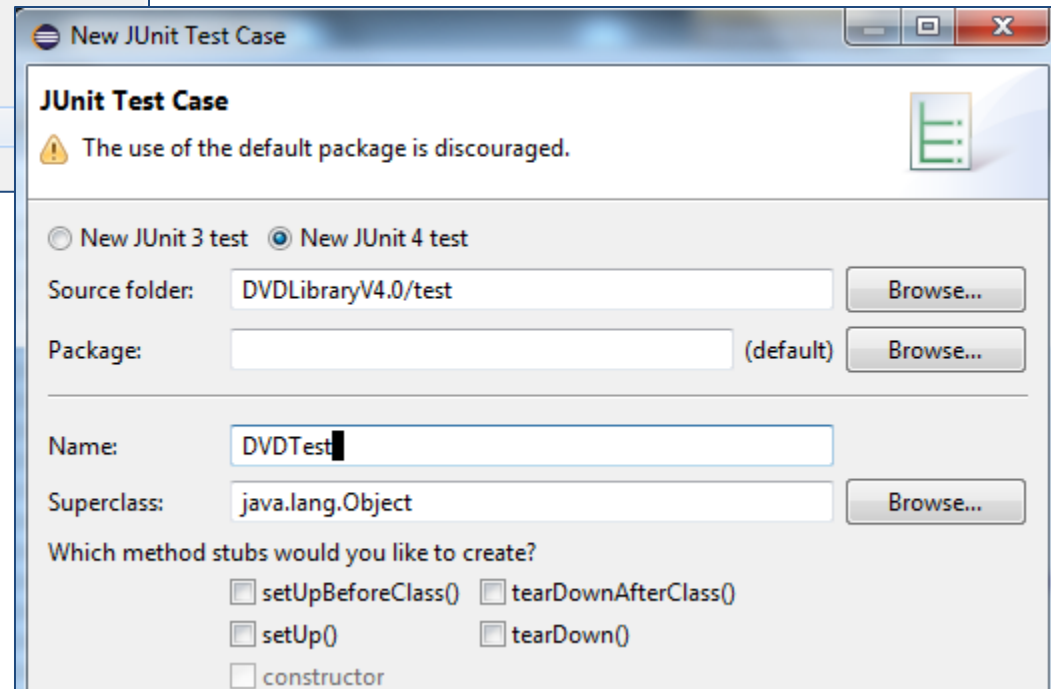
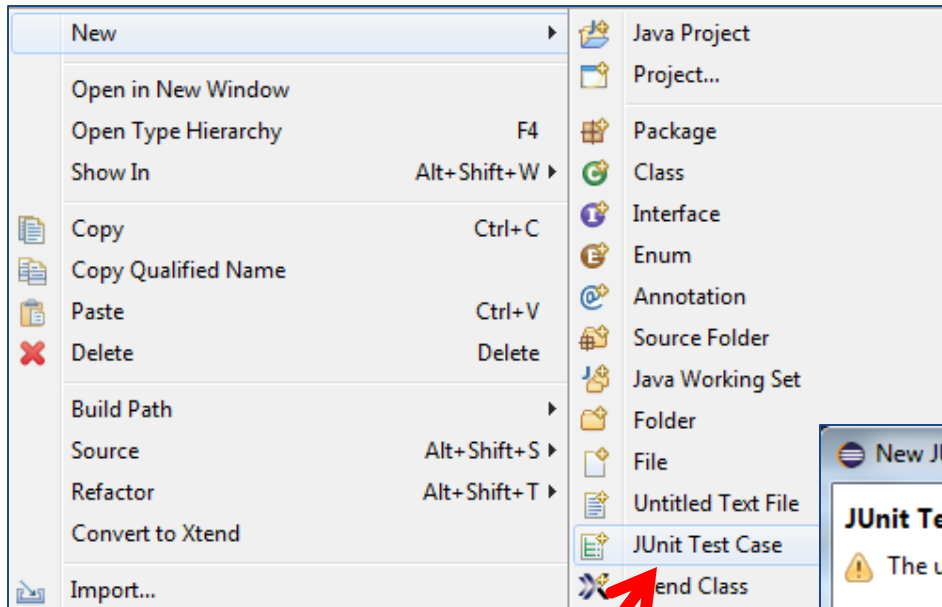
        public String getTitle(){
            return title;
        }

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

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

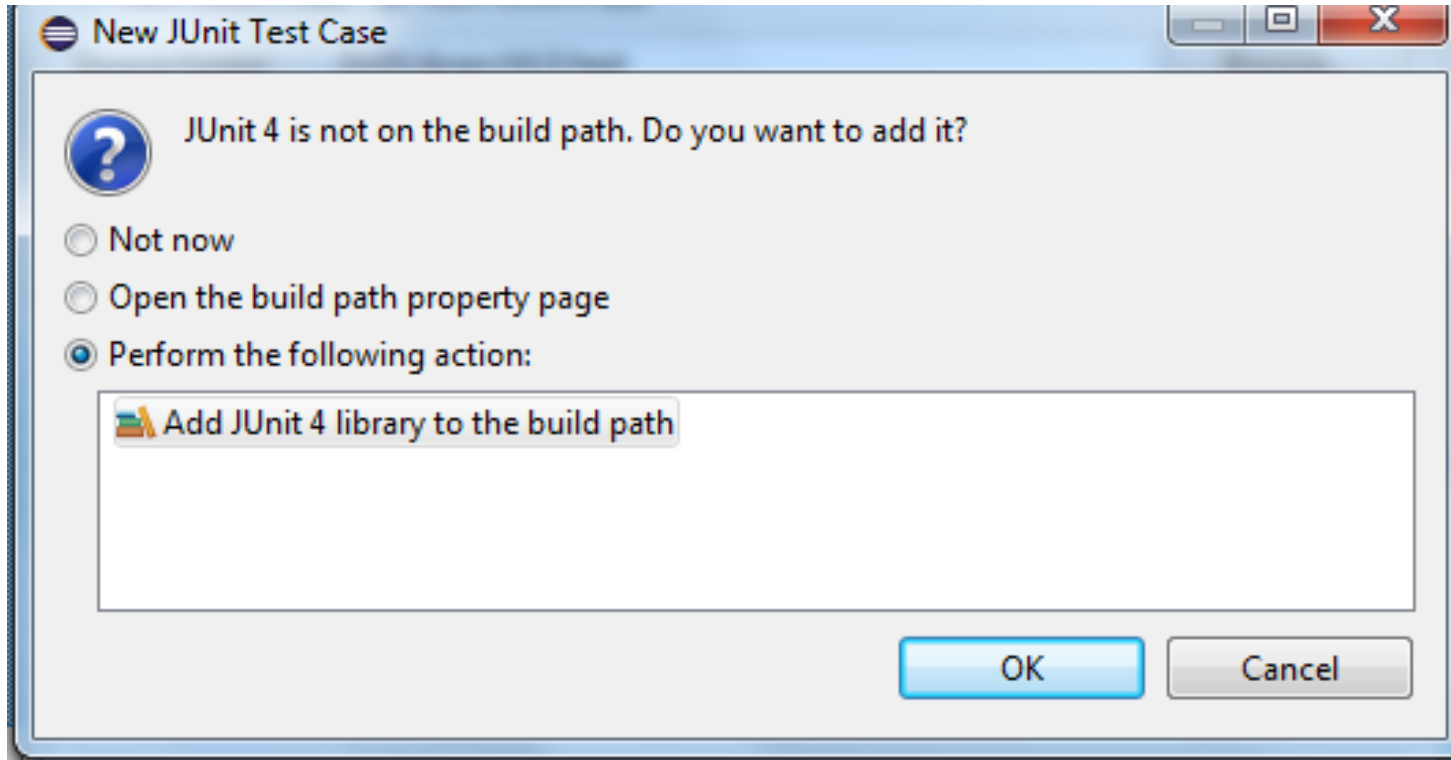


# Within this “test” folder, create a new JUnit Test Case



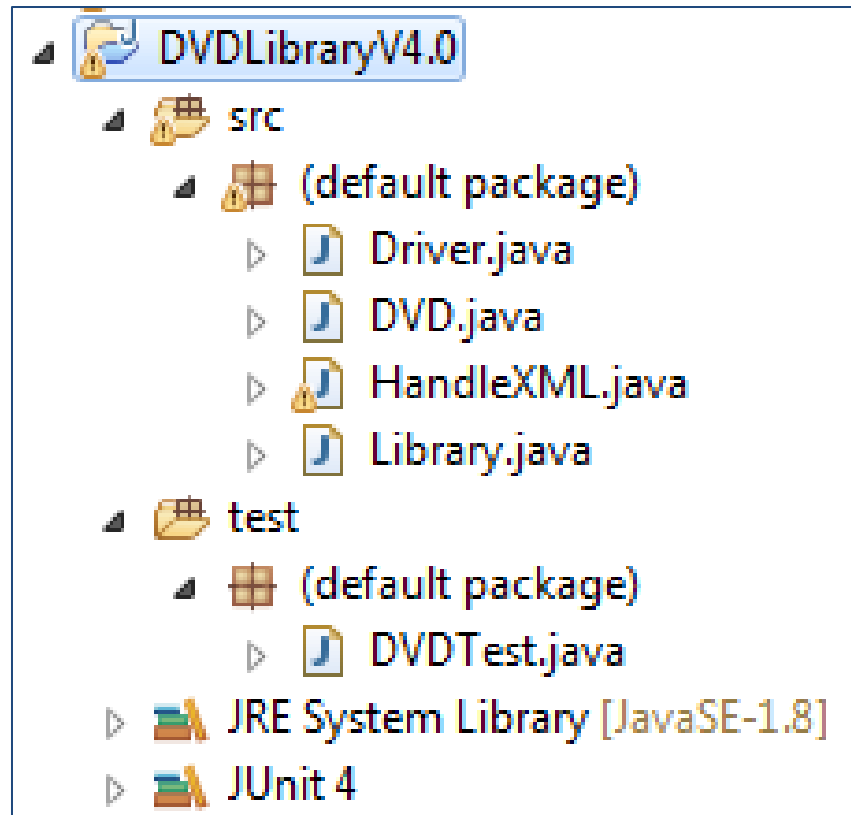
You may be asked to add JUnit 4 to your build path → Click OK.

---



# Revised Folder Structure

---



# The generated DVDTest.java

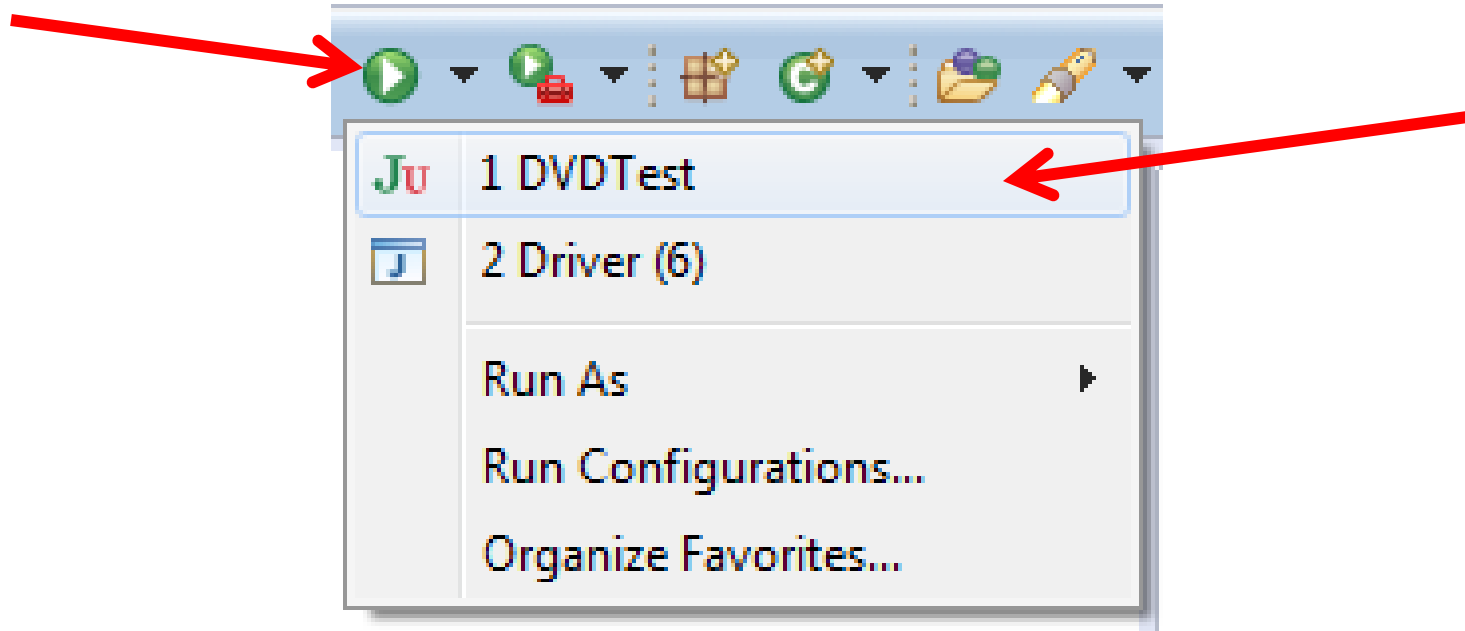
---

```
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
```



# Running DVDTest.java

---



# JUnit Results Pane (1)

The screenshot displays the JUnit Results Pane in an IDE. The top section shows the test execution summary: "Finished after 0.067 seconds", "Runs: 1/1", "Errors: 0", and "Failures: 1". A red progress bar indicates the failure. Below this, the test results are listed: "DVDTest [Runner: JUnit 4] (0.003 s)" and "test (0.003 s)".

The code editor on the right shows the source code for `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
```

The bottom section of the IDE shows the Failure Trace:

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

The Console pane at the bottom right shows the command prompt output: `<terminated> DVDTest [JUnit] C:\Program Files\Java\jre1.8.0_25\bin\jav`

# JUnit Results Pane (2)

Package Explorer JUnit

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.

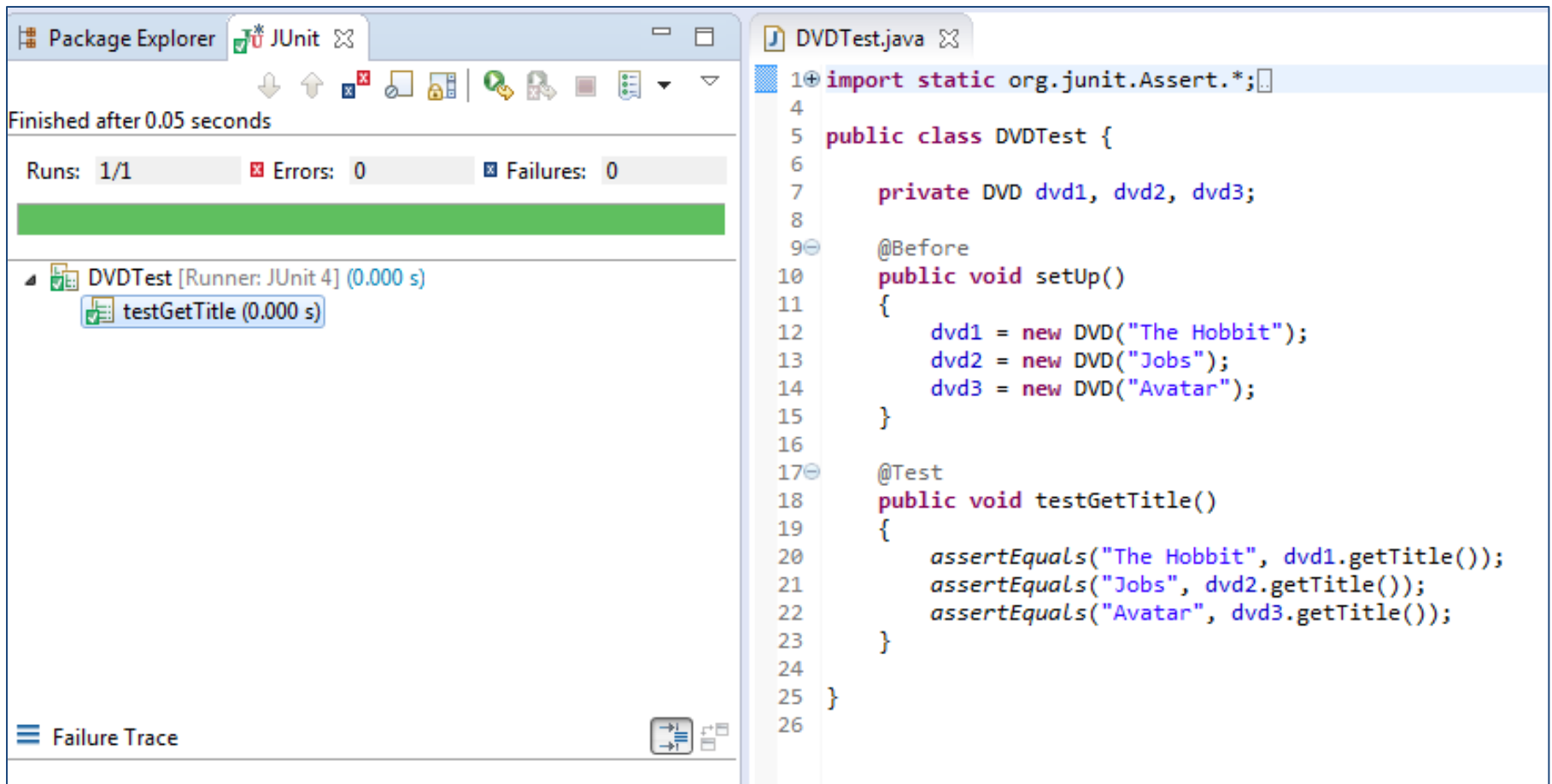
Test Bar indicating failure.

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;  
}
```



The screenshot displays an IDE window with two panes. The left pane shows the JUnit test runner interface, indicating a successful test run for DVDTest. The right pane shows the source code for DVDTest.java, which includes a setUp() method for initializing DVD objects and a testGetTitle() method that uses assertEquals() to verify the titles.

Package Explorer JUnit

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

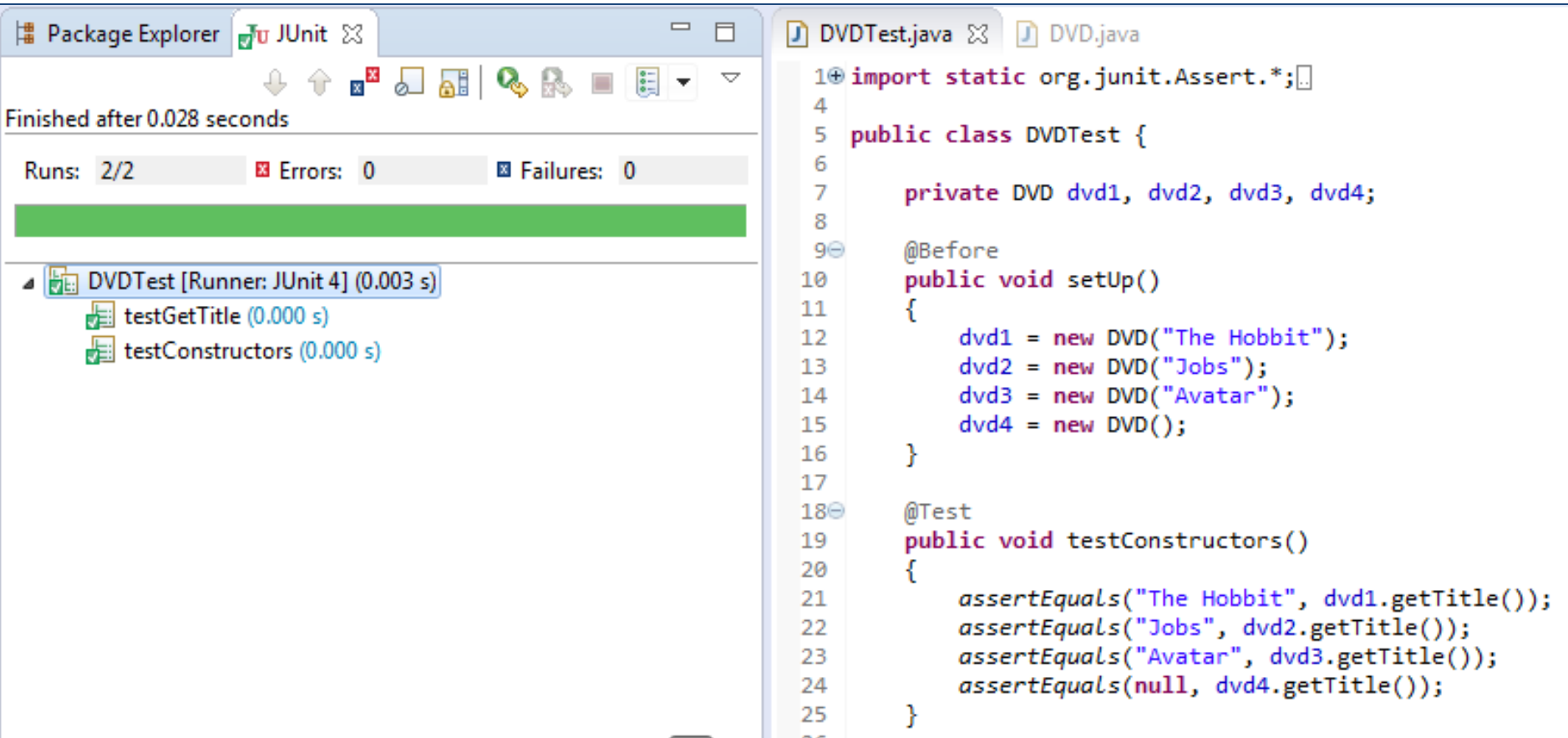
```
DVDTest.java  
1 import static org.junit.Assert.*;  
4  
5 public class DVDTest {  
6  
7     private DVD dvd1, dvd2, dvd3;  
8  
9     @Before  
10    public void setUp()  
11    {  
12        dvd1 = new DVD("The Hobbit");  
13        dvd2 = new DVD("Jobs");  
14        dvd3 = new DVD("Avatar");  
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)

---

```
J *DVDTest.java  J DVD.java ✕
12  /**
13   * Constructors for objects of class DVD
14   */
15  public DVD(String title)
16  {
17      if (title.length() <= 20){
18          this.title = title;
19      }
20      else{
21          this.title = title.substring(0,20);
22      }
23  }
24
25  public DVD(){
26  }
27
```

# Testing Constructors (2)



The screenshot displays an IDE interface with two main panels. The left panel, titled 'Package Explorer', shows the results of a JUnit test run. It indicates that the test finished after 0.028 seconds, with 2/2 runs, 0 errors, and 0 failures. A green progress bar is visible. Below this, the test results for 'DVDTest [Runner: JUnit 4] (0.003 s)' are listed, showing two successful tests: 'testGetTitle (0.000 s)' and 'testConstructors (0.000 s)'. The right panel shows the source code for 'DVDTest.java'. The code includes an import for JUnit's Assert class, a class definition for 'DVDTest', a private array of 'DVD' objects, a '@Before' annotated 'setUp()' method that initializes the DVD objects with titles 'The Hobbit', 'Jobs', and 'Avatar', and a '@Test' annotated 'testConstructors()' method that uses 'assertEquals' to verify the titles of the first three DVD objects and that the fourth is null.

```
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());
}
```

```
test.java    DVD.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)

```
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)

Package Explorer JUnit

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)

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!

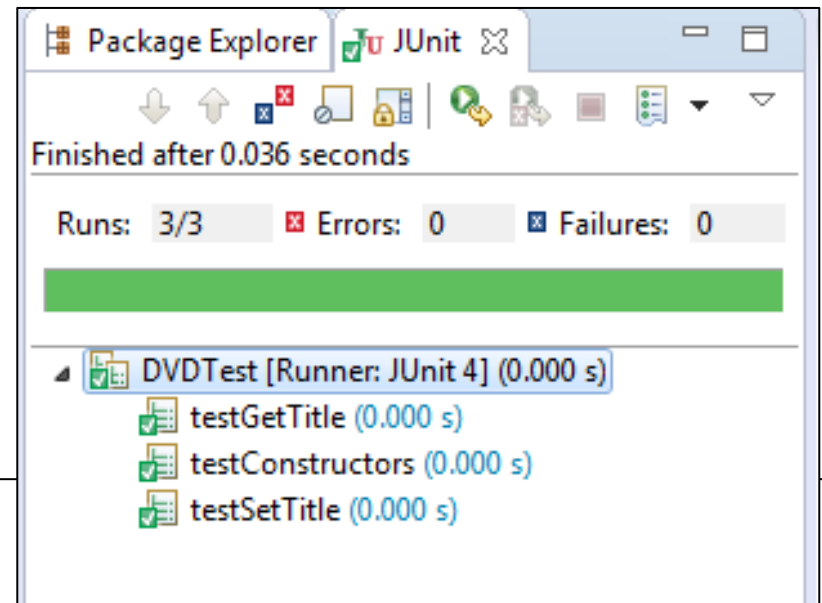
Fixed!

# Testing setTitle method (1)

```
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
35 @Test
36 public void testSetTitle()
37 {
38     dvd1.setTitle("The Hobbit");
39     assertEquals("The Hobbit", dvd1.getTitle());
40     dvd1.setTitle("The Hobbit (Director)"); //attempting to set title to 21 characters
41     assertEquals("The Hobbit", dvd1.getTitle());
42     dvd1.setTitle("The Hobbit(Director)"); //attempting to set title to 20 characters
43     assertEquals("The Hobbit(Director)", dvd1.getTitle());
44     dvd1.setTitle("The Hobbit:Director"); //attempting to set title to 20 characters
45     assertEquals("The Hobbit:Director", dvd1.getTitle());
46 }
47 }
48
```

# Testing setTitle method (2)

```
@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;
}
}
```

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
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

DVDTTest.java DVD.java

```
1 import static org.junit.Assert.*;
4
5 public class DVDTTest {
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
```

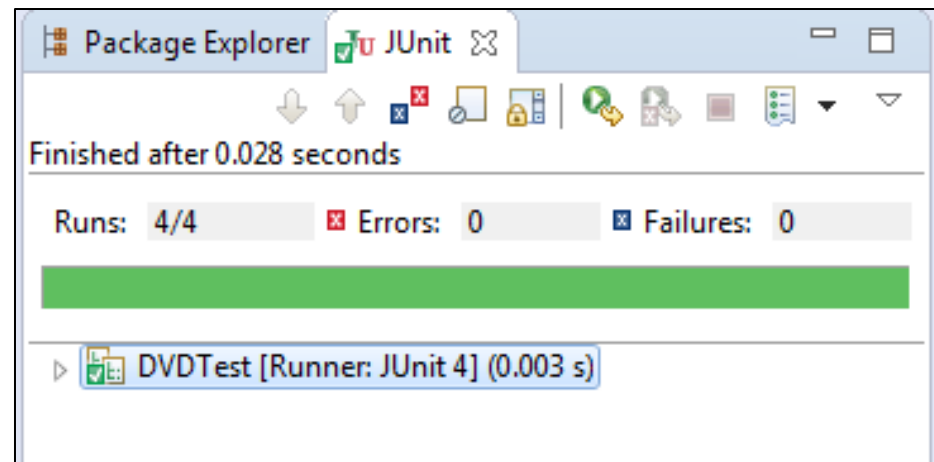
We are referencing the wrong dvd objects in the last two method calls.



# 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.



The screenshot shows the IDE's JUnit runner interface. At the top, it says "Finished after 0.028 seconds". Below that, the test results are displayed: "Runs: 4/4", "Errors: 0", and "Failures: 0". A green progress bar is visible, indicating a successful run. At the bottom, the test name "DVDTest [Runner: JUnit 4] (0.003 s)" is shown.

# 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?**





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/>