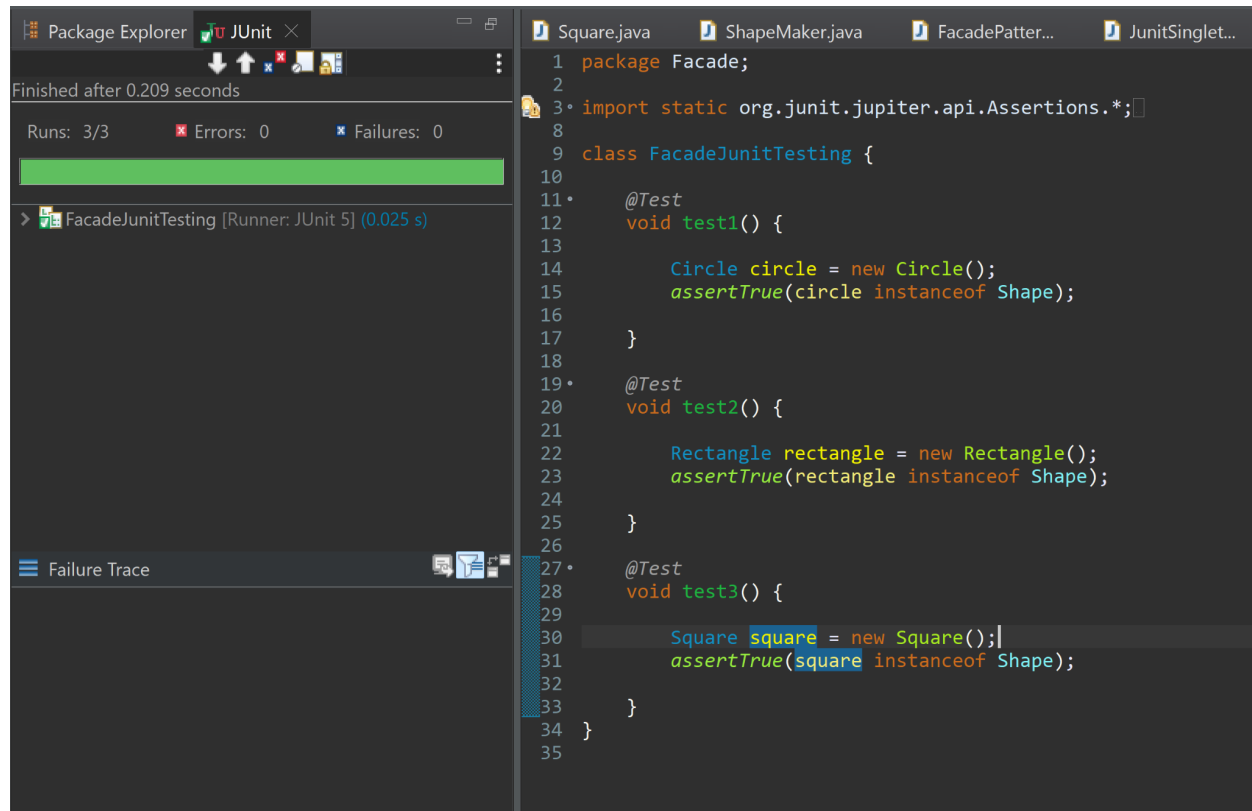


Screen Capture of Junit Tests

Part 1

Facade



The screenshot shows an IDE with the JUnit test runner interface on the left and the source code on the right. The test runner indicates that the tests passed successfully.

JUnit Test Results:

- Package Explorer: JUnit
- Finished after 0.209 seconds
- Runs: 3/3
- Errors: 0
- Failures: 0
- FacadeJUnitTesting [Runner: JUnit 5] (0.025 s)

Source Code (Square.java):

```
1 package Facade;
2
3 import static org.junit.jupiter.api.Assertions.*;
4
5 class FacadeJUnitTesting {
6
7     @Test
8     void test1() {
9
10         Circle circle = new Circle();
11         assertTrue(circle instanceof Shape);
12     }
13
14     @Test
15     void test2() {
16
17         Rectangle rectangle = new Rectangle();
18         assertTrue(rectangle instanceof Shape);
19     }
20
21     @Test
22     void test3() {
23
24         Square square = new Square();
25         assertTrue(square instanceof Shape);
26     }
27 }
28
29
30
31
32
33
34
35
```

Factory

```
class FactoryJUnitTesting {  
  
    @Test  
    void test1() {  
  
        ShapeFactory shapeFactory = new ShapeFactory();  
  
        Shape shape = shapeFactory.getShape("CIRCLE");  
  
        assertTrue(shape instanceof Circle);  
    }  
  
    @Test  
    void test2() {  
  
        ShapeFactory shapeFactory = new ShapeFactory();  
  
        Shape shape = shapeFactory.getShape("RECTANGLE");  
  
        assertTrue(shape instanceof Rectangle);  
    }  
  
    @Test  
    void test3() {  
  
        ShapeFactory shapeFactory = new ShapeFactory();  
  
        Shape shape = shapeFactory.getShape("SQUARE");  
  
        assertTrue(shape instanceof Square);  
    }  
}
```


Finished after 0.204 seconds


Runs: 3/3

Errors: 0

Failures: 0



>  FactoryJUnitTesting [Runner: JUnit 5] (0.033 s)

 Failure Trace



Filter

```
@Test
void test1() {

    List<Person> persons = new ArrayList<Person>();
    persons.add(new Person("Jane", "Female", "No"));
    persons.add(new Person("Bob", "Male", "No"));

    Criteria female = new CriteriaFemale();

    List<Person> actual = female.meetCriteria(persons);
    List<Person> expected = new ArrayList<Person>();

    expected.add(new Person("Jane", "Female", "No"));

    assertEquals(expected.get(0).getName(), actual.get(0).getName());
}

@Test
void test2() {

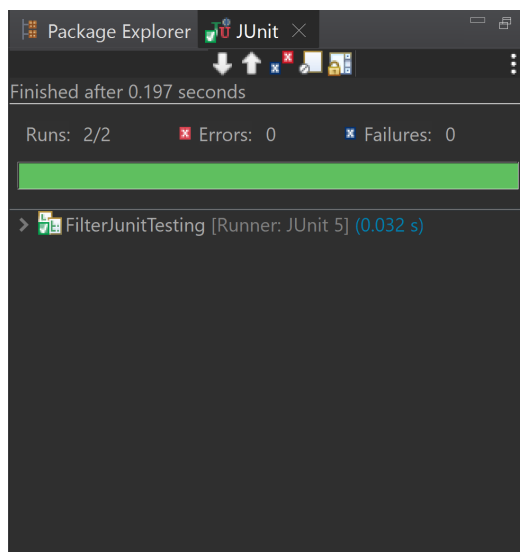
    List<Person> persons = new ArrayList<Person>();
    persons.add(new Person("Jane", "Female", "No"));
    persons.add(new Person("Bob", "Male", "No"));

    Criteria male = new CriteriaMale();

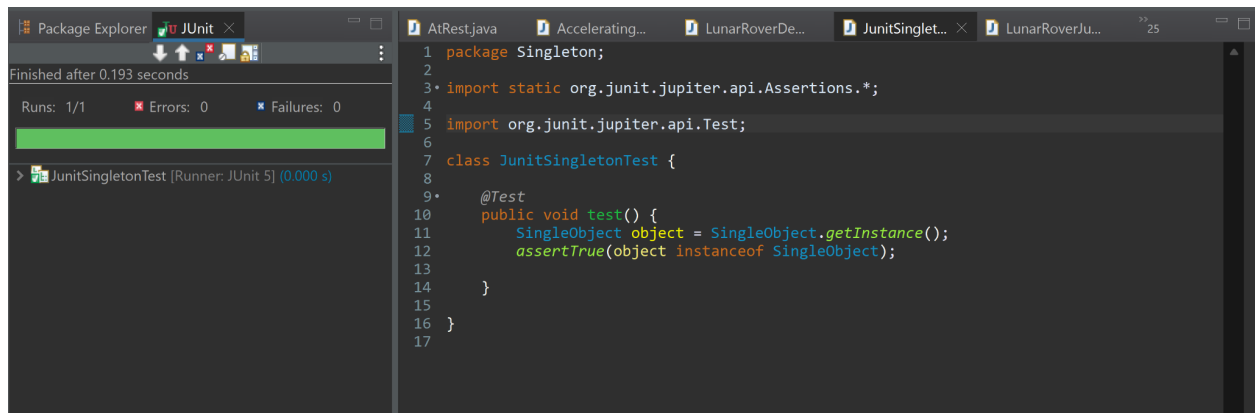
    List<Person> actual = male.meetCriteria(persons);
    List<Person> expected = new ArrayList<Person>();

    expected.add(new Person("Bob", "Male", "No"));

    assertEquals(expected.get(0).getName(), actual.get(0).getName());
}
```



Singleton



Part 2

```
1 package Part2;
2
3 import static org.junit.jupiter.api.Assertions.*;
4
5
6
7 class LunarRoverJUnitTesting {
8
9
10     // When the left pedal was pressed once it accelerates the buggy forward.
11     @Test
12     void test1() {
13         LunarRoverContext lunarRover = new LunarRoverContext();
14         lunarRover.l1();
15         String output = lunarRover.getState().toString();
16         assertEquals(output, "Accelerating Forwards State");
17     }
18
19     //If accelerating forward and you press right pedal twice it deaccelerates.
20     @Test
21     void test2() {
22         LunarRoverContext lunarRover = new LunarRoverContext();
23         lunarRover.l1();
24         lunarRover.r2();
25         String output = lunarRover.getState().toString();
26         assertEquals(output, "Deaccelerating State");
27     }
28 }
```

```

29
30 //To achieve constant forward speed press the right pedal for more than 5 seconds.
31 • @Test
32 void test3() {
33     LunarRoverContext lunarRover = new LunarRoverContext();
34     lunarRover.l1();
35     lunarRover.rm5();
36     String output = lunarRover.getState().toString();
37     assertEquals(output, "Constant Forward Speed State");
38 }
39
40 //If the buggy is at rest and the left pedal is pressed for more than
41 //accelerate backwards.
42 • @Test
43 void test4() {
44     LunarRoverContext lunarRover = new LunarRoverContext();
45     lunarRover.lm5();
46     String output = lunarRover.getState().toString();
47     assertEquals(output, "Accelerating Backwards State");
48 }
49

```

Package Explorer JUnit

Finished after 0.206 seconds

Runs: 4/4 Errors: 0 Failures: 0

> LunarRoverJUnitTesting [Runner: JUnit 5] (0.000 s)

