

Software Engineering

: Junit Test Case Generation

소프트웨어학과
201333186 문지환

1# Use the Junit Test Case Generation

1# Use the Junit Test Case Generation

The screenshot displays the IntelliJ IDEA IDE interface. On the left, the source code editor shows the implementation of the `testCube_1()` method. The code includes annotations for exception throwing and generation, followed by the method body which initializes a `testABC` fixture, sets variables `a` and `b` to 1, calls the `cube(a, b)` method, and asserts the result against 1.0.

```
6 *  
7 * @throws Exception  
8 *  
9 * @generatedBy CodePro at 18. 5. 28. 오후 9:54  
10 */  
11 @Test  
12 public void testCube_1()  
13     throws Exception {  
14     testABC fixture = new testABC();  
15     int a = 1;  
16     int b = 1;  
17  
18     double result = fixture.cube(a, b);  
19  
20     // add additional test code here  
21     assertEquals(1.0, result, 0.1);  
22 }  
23
```

On the right, the Run tool window shows the execution results of the test. A single test case, `testCube_1`, passed successfully. The output table details the arguments used and the resulting value.

	Test Method	Arguments	Ass...ns		
		fixture	a	b	result
1	testCube_1	new testABC()	1	1	1.0

Just testing Generated Codes

1# Use the Junit Test Case Generation

* testABC.cube(int, int)					
Test Method	Arguments			Ass...ns	
	fixture	a	b	result	
1 testCube_1	new testABC()	1	1	1.0	
2 testCube_2	new testABC()	2	3	4	
3 testCube_3	new testABC()	3	3	27	
4 testCube_4	new testABC()	5	3	125	
5 testCube_5	new testABC()	7	3	49	
6 testCube_6	new testABC()	10	3	1000	



Generated codes
Including wrong
Expected value

```
21 @Test
22 public void testCube_1()
23     throws Exception {
24     testABC fixture = new testABC();
25     int a = 1;
26     int b = 1;
27
28     double result = fixture.cube(a, b);
29
30     // add additional test code here
31     assertEquals(1.0, result, 0.1);
32 }
33
34 @Test
35 public void testCube_2()
36     throws Exception {
37     testABC testABC = new testABC();
38     int a = 2;
39     int b = 3;
40     double result = testABC.cube(a, b);
41     assertEquals(4, result, 0.1);
42 }
```

```
44 @Test
45 public void testCube_3()
46     throws Exception {
47     testABC testABC = new testABC();
48     int a = 3;
49     int b = 3;
50     double result = testABC.cube(a, b);
51     assertEquals(27, result, 0.1);
52 }
53
54 @Test
55 public void testCube_4()
56     throws Exception {
57     testABC testABC = new testABC();
58     int a = 5;
59     int b = 3;
60     double result = testABC.cube(a, b);
61     assertEquals(125, result, 0.1);
62 }
```

```
64 @Test
65 public void testCube_5()
66     throws Exception {
67     testABC testABC = new testABC();
68     int a = 7;
69     int b = 3;
70     double result = testABC.cube(a, b);
71     assertEquals(49, result, 0.1);
72 }
73
74 @Test
75 public void testCube_6()
76     throws Exception {
77     testABC testABC = new testABC();
78     int a = 10;
79     int b = 3;
80     double result = testABC.cube(a, b);
81     assertEquals(1000, result, 0.1);
82 }
```

1# Use the Junit Test Case Generation

* testABC.cube(int, int)					
Test Method	Arguments			Ass...ns	
	fixture	a	b	result	
1 testCube_1	new testABC()	1	1	1.0	
2 testCube_2	new testABC()	2	3	8	
3 testCube_3	new testABC()	3	3	27	
4 testCube_4	new testABC()	5	3	125	
5 testCube_5	new testABC()	7	3	343	
6 testCube_6	new testABC()	10	3	1000	



Generated codes
Including correct
Expected value

```
22 public void testCube_1()
23     throws Exception {
24     testABC fixture = new testABC();
25     int a = 1;
26     int b = 1;
27
28     double result = fixture.cube(a, b);
29
30     // add additional test code here
31     assertEquals(1.0, result, 0.1);
32 }
33
34 @Test
35 public void testCube_2()
36     throws Exception {
37     testABC testABC = new testABC();
38     int a = 2;
39     int b = 3;
40     double result = testABC.cube(a, b);
41     assertEquals(8, result, 0.1);
42 }
```

```
44 @Test
45 public void testCube_3()
46     throws Exception {
47     testABC testABC = new testABC();
48     int a = 3;
49     int b = 3;
50     double result = testABC.cube(a, b);
51     assertEquals(27, result, 0.1);
52 }
53
54 @Test
55 public void testCube_4()
56     throws Exception {
57     testABC testABC = new testABC();
58     int a = 5;
59     int b = 3;
60     double result = testABC.cube(a, b);
61     assertEquals(125, result, 0.1);
62 }
```

```
65 public void testCube_5()
66     throws Exception {
67     testABC testABC = new testABC();
68     int a = 7;
69     int b = 3;
70     double result = testABC.cube(a, b);
71     assertEquals(343, result, 0.1);
72 }
73
74 @Test
75 public void testCube_6()
76     throws Exception {
77     testABC testABC = new testABC();
78     int a = 10;
79     int b = 3;
80     double result = testABC.cube(a, b);
81     assertEquals(1000, result, 0.1);
82 }
```