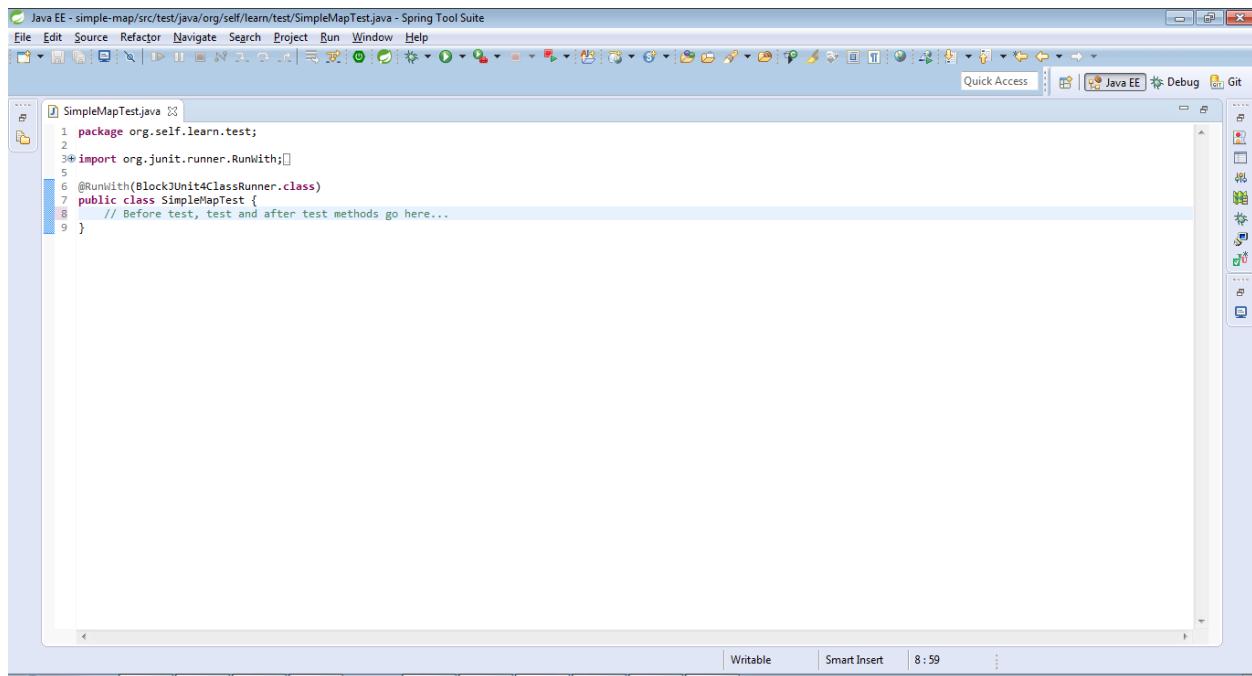


Simple Map without using Java Collections Framework (JCF)

1. Can have Null 'Key'
2. Can have null 'values'
3. Cannot contain duplicate 'keys'
4. May not retain order in which "keys/values" were saved →**No Test Required**
5. Performance is NOT a concern i.e. we will not implement Hash Table like features e.g. bucket, hash values etc. Also redundant iterations are allowed →**No Test Required**
6. Manages size dynamically
7. Throws exception if no such key found
8. Throws exception if duplicate key found
9. Should be able to perform the following operations:
 - a. Create Simple Parametrized Map (Generics)
 - b. Save 'key' and 'value' pair
 - c. Retrieve value given a 'key'
 - d. Remove a "key-value" pair given a 'key'
 - e. Retrieve size of the map i.e. number of "key-value" pairs
 - f. Check if a given 'key' is present
 - g. Check if a given 'value' is present

Fact that we are test driving development, test class is created even before your implementation class.
Here is what an empty implementation class looks like:

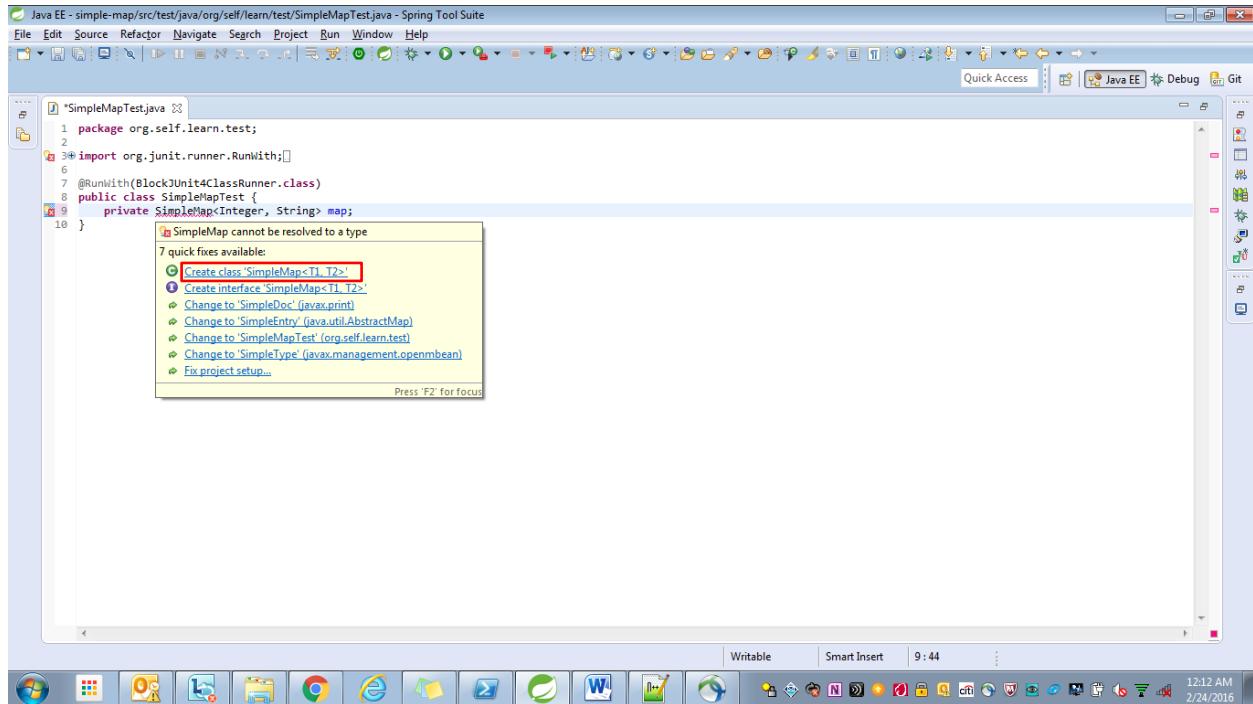


The screenshot shows the Spring Tool Suite (STS) interface. The title bar reads "Java EE - simple-map/src/test/java/org/self/learn/test/SimpleMapTest.java - Spring Tool Suite". The main window displays the code for "SimpleMapTest.java":

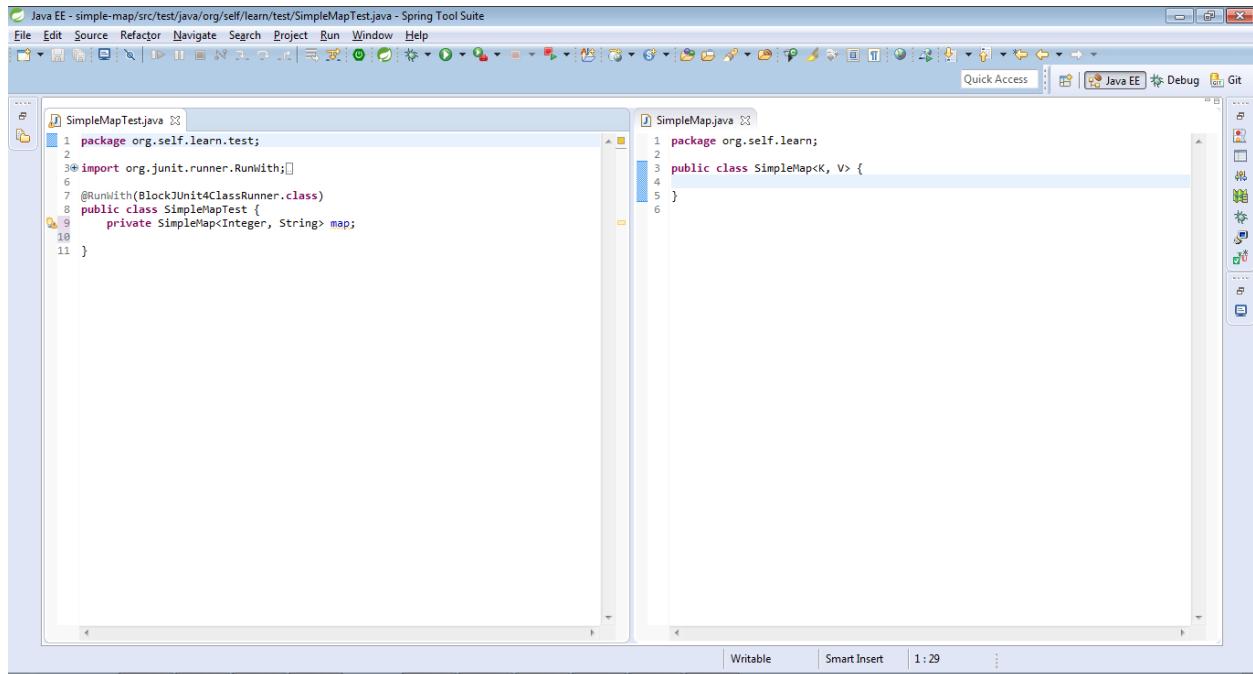
```
1 package org.self.learn.test;
2
3@import org.junit.runner.RunWith;
4
5 @RunWith(BlockJUnit4ClassRunner.class)
6 public class SimpleMapTest {
7     // Before test, test and after test methods go here...
8 }
9 }
```

The code editor has syntax highlighting for Java. The interface includes a toolbar with various icons, a navigation bar with tabs like "File", "Edit", "Source", "Refactor", "Navigate", "Search", "Project", "Run", "Window", "Help", and "Quick Access". On the right side, there is a palette with icons for file operations and Git integration.

Let's start by creating the test class and trying to instantiate the test object. I prefer to use the quick fix to create the class/methods under test. This, although might sound a bit exaggerated, ensures that our test drives our implementation.



Make note of the eclipse window... this is our preferred style for TDD i.e. having test and implementation class side by side.



Key idea behind TDD is that you start by test driving that smallest, simplest and tiniest bit and build on top of it. Start with a blank mind without having any preconceived notion of the implementation. Think about what (**and not how!**) would you like your class to do.

Let's write our first test - "itShouldBeAbleToInitializeAnEmptyMapWithSizeZero()"... Notice how your test tells what your implementation is.

The screenshot shows the Spring Tool Suite interface with two files open:

- SimpleMapTest.java**: Contains the following code:

```
1 package org.self.learn.test;
2
3 import static org.hamcrest.CoreMatchers.*;
4
5 @RunWith(BlockJUnit4ClassRunner.class)
6 public class SimpleMapTest {
7     private SimpleMap<Integer, String> map;
8
9     @Test
10    public void itShouldInitializeAnEmptyMap() {
11        map = new SimpleMap<>();
12        assertThat(map.size(), is(equalTo(0)));
13    }
14 }
```
- SimpleMap.java**: Contains the following code:

```
1 package org.self.learn;
2
3 public class SimpleMap<K, V> {
4
5 }
6
```

A code completion tooltip is displayed over the `size()` method call in the test code. It contains the following options:

- Create method 'size()' in type 'SimpleMap'
- Add cast to 'map'
- Rename in file (Ctrl+2, R)

The tooltip also includes the instruction: "Press 'Tab' from proposal table or click for focus".

I used quick fix on my test class to create my implementation again... With that, let's run the test which would obviously fail as we expect it to...

The screenshot shows the Spring Tool Suite interface with two code editors. The left editor contains `SimpleMapTest.java` and the right editor contains `SimpleMap.java`. The `SimpleMapTest.java` file has the following code:

```

1 package org.self.learn.test;
2
3 import static org.hamcrest.CoreMatchers.*;
4
5 @RunWith(BlockJUnit4ClassRunner.class)
6 public class SimpleMapTest {
7     private SimpleMap<Integer, String> map;
8
9     @Test
10    public void itShouldInitializeAnEmptyMap() {
11        map = new SimpleMap<>();
12        assertThat(map.size(), is(equalTo(0)));
13    }
14
15 }

```

The `SimpleMap.java` file has the following code:

```

1 package org.self.learn;
2
3 public class SimpleMap<K, V> {
4
5     public int size() {
6         // TODO Auto-generated method stub
7         return -1;
8     }
9
10 }

```

Below the editors is a toolbar with various icons. The bottom status bar shows "Finished after 0.03 seconds" and "Runs: 1/1 Errors: 0 Failures: 1". A "Failure Trace" panel on the right shows the error message: "java.lang.AssertionError: Expected: is <0> but: was <-1>".

Returning zero takes care of that... (note that the test completes successfully). Idea is to only implement how much the tests force us to... It ensures that you have 100% code coverage.

The screenshot shows the Spring Tool Suite interface with the same two code editors. The `SimpleMapTest.java` file remains the same as in the previous screenshot. The `SimpleMap.java` file now has the following code:

```

1 package org.self.learn;
2
3 public class SimpleMap<K, V> {
4
5     public int size() {
6         return 0;
7     }
8
9 }

```

The bottom status bar shows "Finished after 0.024 seconds" and "Runs: 1/1 Errors: 0 Failures: 0". The "Failure Trace" panel is empty.

Next I separate the test set up (which in this case is just the test object initialization) into the `@Before` so the test runs it every time before it runs my tests. I have also refactored the name of my test to make it more explanatory.

```

Java EE - simple-map/src/test/java/org/self/learn/test/SimpleMapTest.java - Spring Tool Suite
File Edit Source Refactor Navigate Project Run Window Help
Quick Access Java EE Debug Git
SimpleMapTest.java
1 package org.self.learn.test;
2
3 import static org.hamcrest.CoreMatchers.*;
4
5 @RunWith(BLOCKJUnit4ClassRunner.class)
6 public class SimpleMapTest {
7     private SimpleMap<Integer, String> map;
8
9     @Before
10    public void init() {
11        map = new SimpleMap<>();
12    }
13
14    @Test
15    public void itShouldBeAbleToInitializeAnEmptyMapWithSizeZero() {
16        assertThat(map.size(), is(equalTo(0)));
17    }
18
19
20
21
22
23
24
25
26 }

SimpleMap.java
1 package org.self.learn;
2
3 public class SimpleMap<K, V> {
4
5     public int size() {
6         return 0;
7     }
8
9 }
10

```

Run: 1/1 Errors: 0 Failures: 0

Failure Trace

Let's get rolling with the subsequent tests based on a similar approach. The second test is "itShouldReturnSizeGreaterThanZeroUponAddingKeyValuePair()"...

Note again how the test demands implementation...

```

Java EE - simple-map/src/test/java/org/self/learn/test/SimpleMapTest.java - Spring Tool Suite
File Edit Source Refactor Navigate Project Run Window Help
Quick Access Java EE Debug Git
SimpleMapTest.java
1 package org.self.learn.test;
2
3 import static org.hamcrest.Matchers.*;
4 import static org.junit.Assert.*;
5
6 import org.junit.Before;
7 import org.junit.Test;
8 import org.junit.runner.RunWith;
9 import org.junit.runners.BlockJUnit4ClassRunner;
10 import org.self.learn.SimpleMap;
11
12 @RunWith(BLOCKJUnit4ClassRunner.class)
13 public class SimpleMapTest {
14     private SimpleMap<Integer, String> map;
15
16     @Before
17     public void init() {
18         map = new SimpleMap<>();
19     }
20
21     @Test
22     public void itShouldReturnSizeGreaterThanZeroUponAddingKeyValuePair() {
23         map.add(1, "Priyan");
24         assertThat(map.size(), is(greaterThan(0)));
25     }
26
27 }

SimpleMap.java
1 package org.self.learn;
2
3 public class SimpleMap<K, V> {
4
5     public int size() {
6         return 0;
7     }
8
9 }
10

The method add(int, String) is undefined for the type SimpleMap<Integer, String>

```

```
Java EE - simple-map/src/test/java/org/self/learn/test/SimpleMapTest.java - Spring Tool Suite
File Edit Source Refactor Navigate Project Run Window Help
SimpleMapTest.java simple-map/pom.xml
1 package org.self.learn.test;
2
3 import static org.hamcrest.Matchers.*;
4 import static org.junit.Assert.*;
5
6 import org.junit.Before;
7 import org.junit.Test;
8 import org.junit.runner.RunWith;
9 import org.junit.runners.BlockJUnit4ClassRunner;
10 import org.self.learn.SimpleMap;
11
12 @RunWith(BlockJUnit4ClassRunner.class)
13 public class SimpleMapTest {
14     private SimpleMap<Integer, String> map;
15
16     @Before
17     public void init() {
18         map = new SimpleMap<>();
19     }
20
21     @Test
22     public void itShouldBeAbleToInitializeAnEmptyMapWithSizeZero() {
23         assertThat(map.size(), is(equalTo(0)));
24     }
25
26     @Test
27     public void itShouldReturnSizeGreaterThanZeroUponAddingKeyValuePair() {
28         map.add(1, "Priyan");
29         assertThat(map.size(), is(greaterThan(0)));
30     }
31 }
```

```
Java EE - simple-map/src/main/java/org/self/learn/SimpleMap.java - Spring Tool Suite
File Edit Source Refactor Navigate Project Run Window Help
SimpleMap.java
1 package org.self.learn;
2
3 public class SimpleMap<K, V> {
4
5     public int size() {
6         return 0;
7     }
8
9     public void add(K i, V string) {
10        // TODO Auto-generated method stub
11    }
12
13 }
14
```

Test fails for obvious reason when ran...

```
Java EE - simple-map/src/test/java/org/self/learn/test/SimpleMapTest.java - Spring Tool Suite
File Edit Source Refactor Navigate Project Run Window Help
SimpleMapTest.java simple-map/pom.xml
9 import org.junit.runners.BlockJUnit4ClassRunner;
10 import org.self.learn.SimpleMap;
11
12 @RunWith(BlockJUnit4ClassRunner.class)
13 public class SimpleMapTest {
14     private SimpleMap<Integer, String> map;
15
16     @Before
17     public void init() {
18         map = new SimpleMap<>();
19     }
20
21     @Test
22     public void itShouldBeAbleToInitializeAnEmptyMapWithSizeZero() {
23         assertThat(map.size(), is(equalTo(0)));
24     }
25
26     @Test
27     public void itShouldReturnSizeGreaterThanZeroUponAddingKeyValuePair() {
28         map.add(1, "Priyan");
29         assertThat(map.size(), is(greaterThan(0)));
30     }
31 }
```

```
Java EE - simple-map/src/main/java/org/self/learn/SimpleMap.java - Spring Tool Suite
File Edit Source Refactor Navigate Project Run Window Help
SimpleMap.java
1 package org.self.learn;
2
3 public class SimpleMap<K, V> {
4
5     public int size() {
6         return 0;
7     }
8
9     public void add(K i, V string) {
10        // TODO Auto-generated method stub
11    }
12
13 }
14
```

Markers Properties Servers Data Source Explorer Debug Terminal JUnit

Finished after 0.041 seconds

Runs: 2/2 Errors: 0 Failures: 1

org.self.learn.test.SimpleMapTest [Runner: JUnit 4] (0.001 s)

- itShouldBeAbleToInitializeAnEmptyMapWithSizeZero (0.000 s)
- itShouldReturnSizeGreaterThanZeroUponAddingKeyValuePair (0.001 s)

Failure Trace

java.lang.AssertionError:
Expected: is a value greater than <0>
but: <0> was equal to <0>

at org.hamcrest.MatcherAssert.assertThat(MatcherAssert.java:20)

I again did the smallest change that would make my tests run successfully and they do...

The screenshot shows the Spring Tool Suite interface. On the left, the code editor displays `SimpleMapTest.java` and `SimpleMap.java`. `SimpleMapTest.java` contains JUnit tests for the `SimpleMap` class. `SimpleMap.java` defines a map implementation with a size counter. The bottom status bar indicates the tests finished after 0.028 seconds with 2/2 runs, 0 errors, and 0 failures. The JUnit view shows three test cases: `itShouldBeAbleToInitializeAnEmptyMapWithSizeZero` and `itShouldReturnSizeGreaterThanZeroUponAddingKeyValuePair` both took 0.000 s.

Quick refactor i.e. initialize “int size” inside the constructor. Tests still run fine.

Note: Refactors are anything that improve code quality while not change the behavior of my implementation

The screenshot shows the Spring Tool Suite interface after refactoring. The code editor now shows the updated `SimpleMap.java` where the `size` variable is initialized in the constructor. The tests remain successful, running in 0.026 seconds with 2/2 runs, 0 errors, and 0 failures. The JUnit view shows the same three test cases with execution times of 0.001 s each.

Another “`itShouldBeAbleToReturnSizeOfTheMap()`” test which the implementation has already taken care though...

The screenshot shows the Spring Tool Suite interface with two code editors open. The left editor contains `SimpleMapTest.java` and the right editor contains `SimpleMap.java`. The test class has three test methods: `itShouldBeAbleToInitializeAnEmptyMapWithSizeZero()`, `itShouldReturnSizeGreaterThanZeroUponAddingKeyValuePair()`, and `itShouldBeAbleToReturnSizeOfTheMap()`. All three tests pass, indicated by a green progress bar at the bottom.

```

SimpleMapTest.java
15  @Before
16  public void init() {
17      map = new SimpleMap<K, V>();
18  }
19
20
21@Test
22 public void itShouldBeAbleToInitializeAnEmptyMapWithSizeZero() {
23     assertThat(map.size(), is(equalTo(0)));
24 }
25
26@Test
27 public void itShouldReturnSizeGreaterThanZeroUponAddingKeyValuePair() {
28     map.add(1, "Priyan");
29     assertThat(map.size(), is(greaterThan(0)));
30 }
31
32@Test
33 public void itShouldBeAbleToReturnSizeOfTheMap() {
34     map.add(1, "Priyan");
35     assertThat(map.size(), is(equalTo(1)));
36 }
37 }

SimpleMap.java
1 package org.self.learn;
2
3 public class SimpleMap<K, V> {
4     private int size;
5
6     public SimpleMap() {
7         this.size = 0;
8     }
9
10    public int size() {
11        return this.size;
12    }
13
14    public void add(K key, V value) {
15        this.size = 1;
16    }
17
18 }
19

```

Well, the tests pass giving an illusion of completeness of the implementation. Let's find out if that's really the case though, by adding another test case...

The test fails this time...

The screenshot shows the Spring Tool Suite interface with the same two code editors. The test class now has four test methods: `itShouldBeAbleToInitializeAnEmptyMapWithSizeZero()`, `itShouldBeAbleToReturnSizeOfTheMap()`, `itShouldReturnSizeGreaterThanZeroUponAddingKeyValuePair()`, and a new method `itShouldBeAbleToReturnSizeOfTheMap()`. The last test fails with an `AssertionError`, indicated by a red progress bar at the bottom.

```

SimpleMapTest.java
16  @Before
17  public void init() {
18      map = new SimpleMap<K, V>();
19  }
20
21@Test
22 public void itShouldBeAbleToInitializeAnEmptyMapWithSizeZero() {
23     assertThat(map.size(), is(equalTo(0)));
24 }
25
26@Test
27 public void itShouldReturnSizeGreaterThanZeroUponAddingKeyValuePair() {
28     map.add(1, "Priyan");
29     assertThat(map.size(), is(greaterThan(0)));
30 }
31
32@Test
33 public void itShouldBeAbleToReturnSizeOfTheMap() {
34     map.add(1, "Priyan");
35     assertThat(map.size(), is(equalTo(1)));
36     map.add(5, "Priyadarshan");
37     assertThat(map.size(), is(equalTo(2)));
38 }

SimpleMap.java
1 package org.self.learn;
2
3 public class SimpleMap<K, V> {
4     private int size;
5
6     public SimpleMap() {
7         this.size = 0;
8     }
9
10    public int size() {
11        return this.size;
12    }
13
14    public void add(K key, V value) {
15        this.size = 1;
16    }
17
18 }
19

```

Let's take the next leap to address the test for size of the simple map. And fix does work, as the test confirms.

```

Java EE - simple-map/src/main/java/org/self/learn/SimpleMap.java - Spring Tool Suite
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access Java EE Debug Git
SimpleMapTest.java simple-map/pom.xml
16@ Before
17 public void init() {
18     map = new SimpleMap<K, V>();
19 }
20
21@ Test
22 public void itShouldBeAbleToInitializeAnEmptyMapWithSizeZero() {
23     assertThat(map.size(), isEqualTo(0));
24 }
25
26@ Test
27 public void itShouldReturnSizeGreaterThanZeroUponAddingKeyValuePair() {
28     map.add(1, "Priyan");
29     assertThat(map.size(), is(greaterThan(0)));
30 }
31
32@ Test
33 public void itShouldBeAbleToReturnSizeOfTheMap() {
34     map.add(1, "Priyan");
35     assertThat(map.size(), isEqualTo(1));
36     map.add(5, "Priyadarshan");
37     assertThat(map.size(), isEqualTo(2));
38 }
39
40
SimpleMap.java
1 package org.self.learn;
2
3 public class SimpleMap<K, V> {
4     private int size;
5
6     public SimpleMap() {
7         this.size = 0;
8     }
9
10    public int size() {
11        return this.size;
12    }
13
14    public void add(K key, V value) {
15        this.size++;
16    }
17
18 }
19

```

Markers Properties Servers Data Source Explorer Debug Terminal JUnit

Finished after 0.029 seconds

Runs: 3/3 Errors: 0 Failures: 0

Failure Trace

- org.self.learn.test.SimpleMapTest [Runner:JUnit4] (0.000 s)
 - itShouldBeAbleToInitializeAnEmptyMapWithSizeZero (0.000 s)
 - itShouldBeAbleToReturnSizeOfTheMap (0.000 s)
 - itShouldReturnSizeGreaterThanZeroUponAddingKeyValuePair (0.000 s)

Let's go an extra mile to make sure that our implementation really works! And it does because tests are green.

```

Java EE - simple-map/src/test/java/org/self/learn/test/SimpleMapTest.java - Spring Tool Suite
File Edit Source Refactor Navigate Project Run Window Help
Quick Access Java EE Debug Git
SimpleMapTest.java simple-map/pom.xml
18     map = new SimpleMap<K, V>();
19
20
21@ Test
22 public void itShouldBeAbleToInitializeAnEmptyMapWithSizeZero() {
23     assertThat(map.size(), isEqualTo(0));
24 }
25
26@ Test
27 public void itShouldReturnSizeGreaterThanZeroUponAddingKeyValuePair() {
28     map.add(1, "Priyan");
29     assertThat(map.size(), is(greaterThan(0)));
30 }
31
32@ Test
33 public void itShouldBeAbleToReturnSizeOfTheMap() {
34     map.add(1, "Priyan");
35     assertThat(map.size(), isEqualTo(1));
36     map.add(5, "Priyadarshan");
37     assertThat(map.size(), isEqualTo(2));
38     map.add(200, "Priyan Parida");
39     assertThat(map.size(), isEqualTo(3));
40 }
41
SimpleMap.java
1 package org.self.learn;
2
3 public class SimpleMap<K, V> {
4     private int size;
5
6     public SimpleMap() {
7         this.size = 0;
8     }
9
10    public int size() {
11        return this.size;
12    }
13
14    public void add(K key, V value) {
15        this.size++;
16    }
17
18 }
19

```

Markers Properties Servers Data Source Explorer Debug Terminal JUnit

Finished after 0.028 seconds

Runs: 3/3 Errors: 0 Failures: 0

Failure Trace

- org.self.learn.test.SimpleMapTest [Runner:JUnit4] (0.000 s)
 - itShouldBeAbleToInitializeAnEmptyMapWithSizeZero (0.000 s)
 - itShouldBeAbleToReturnSizeOfTheMap (0.000 s)
 - itShouldSaveKeyValues (0.000 s)
 - itShouldReturnSizeGreaterThanZeroUponAddingKeyValuePair (0.000 s)

Let's add some meat into this by verifying whether our map is really saving keys/values...

The screenshot shows the Spring Tool Suite interface with two open files: `SimpleMapTest.java` and `SimpleMap.java`. In `SimpleMapTest.java`, there is a call to `map.get(1)`. A tooltip appears over the `get` method in `SimpleMap`, suggesting to 'Create method 'get(int)' in type 'SimpleMap''. The tooltip also includes options like 'Add cast to 'map'', 'Cast to 'Object'', and 'Rename in file (Ctrl+2, R)'. The status bar at the bottom indicates: 'The method get(int) is undefined for the type SimpleMap<Integer, String>'.

Here is what the completed test along with the auto generated method looks like. The test fails for obvious reasons.

The screenshot shows the Spring Tool Suite interface with the same two files. The `SimpleMapTest.java` file now contains several test cases. The `get` method in `SimpleMap` has been implemented as follows:

```

1 package org.self.learn;
2
3 public class SimpleMap<K, V> {
4     private int size;
5
6     public SimpleMap() {
7         this.size = 0;
8     }
9
10    public int size() {
11        return this.size;
12    }
13
14    public void add(K key, V value) {
15        this.size++;
16    }
17
18    public V get(K key) {
19        // TODO Auto-generated method stub
20        return null;
21    }
22
23}

```

The JUnit view shows the results of the run:

- Runs: 4/4
- Errors: 0
- Failures: 1

The failure details are:

- `org.self.learn.test.SimpleMapTest [Runner:JUnit 4] (0.000 s)`
- `itShouldBeAbleToInitializeAnEmptyMapWithSizeZero (0.000 s)`
- `itShouldBeAbleToReturnSizeOfTheMap (0.000 s)`
- `itShouldSaveKeyAndValuePair (0.000 s)`
- `itShouldReturnSizeGreaterThanZeroUponAddingKeyValuePair (0.000 s)`

The failure trace for the failing test is:

```

java.lang.AssertionError:
Expected: is "Priyan"
but: was null
at org.hamcrest.MatcherAssert.assertThat(MatcherAssert.java:20)

```

I have gone ahead and provided implementation to address this part. The tests show that the implementation is correct indeed!

Java EE - simple-map/src/main/java/org/self/learn/SimpleMap.java - Spring Tool Suite

SimpleMapTest.java

```

12 @RunWith(BlocksJUnit4ClassRunner.class)
13 public class SimpleMapTest {
14     private SimpleMap<Integer, String> map;
15
16     @Before
17     public void init() {
18         map = new SimpleMap<>();
19     }
20
21     @Test
22     public void itShouldBeAbleToInitializeAnEmptyMapWithSizeZero() {
23         assertEquals(map.size(), isEqualTo(0));
24     }
25
26     @Test
27     public void itShouldReturnSizeGreaterThanZeroUponAddingKeyValuePair() {
28         map.add(1, "Priyan");
29         assertEquals(map.size(), isGreaterThan(0));
30     }
31
32     @Test
33     public void itShouldBeAbleToReturnSizeOfTheMap() {
34         map.add(1, "Priyan");
35         assertEquals(map.size(), isEqualTo(1));
36         map.add(5, "Priyadarshan");
37         assertEquals(map.size(), isEqualTo(2));
38         map.add(200, "Priyan Parida");
39         assertEquals(map.size(), isEqualTo(3));
40     }
41
42     @Test
43     public void itShouldSaveKeyAndValuePair() throws Exception {
44         map.add(1, "Priyan");
45         assertEquals(map.get(1), isEqualTo("Priyan"));
46     }
47 }

```

SimpleMap.java

```

1 package org.self.learn;
2
3 public class SimpleMap<K, V> {
4     private int size;
5     private K key;
6     private V value;
7
8     public SimpleMap() {
9         this.size = 0;
10    }
11
12     public int size() {
13         return this.size;
14     }
15
16     public void add(K key, V value) {
17         this.key = key;
18         this.value = value;
19         this.size++;
20     }
21
22     public V get(K key) {
23         return this.value;
24     }
25
26 }
27

```

Java EE - simple-map/src/main/java/org/self/learn/SimpleMap.java - Spring Tool Suite

SimpleMapTest.java

```

25
26     @Test
27     public void itShouldReturnSizeGreaterThanZeroUponAddingKeyValuePair() {
28         map.add(1, "Priyan");
29         assertEquals(map.size(), isGreaterThan(0));
30     }
31
32     @Test
33     public void itShouldBeAbleToReturnSizeOfTheMap() {
34         map.add(1, "Priyan");
35         assertEquals(map.size(), isEqualTo(1));
36         map.add(5, "Priyadarshan");
37         assertEquals(map.size(), isEqualTo(2));
38         map.add(200, "Priyan Parida");
39         assertEquals(map.size(), isEqualTo(3));
40     }
41
42     @Test
43     public void itShouldSaveKeyAndValuePair() throws Exception {
44         map.add(1, "Priyan");
45         assertEquals(map.get(1), isEqualTo("Priyan"));
46     }
47 }

```

SimpleMap.java

```

5     private K key;
6     private V value;
7
8     public SimpleMap() {
9         this.size = 0;
10    }
11
12     public int size() {
13         return this.size;
14     }
15
16     public void add(K key, V value) {
17         this.key = key;
18         this.value = value;
19         this.size++;
20     }
21
22     public V get(K key) {
23         return this.value;
24     }
25
26 }
27

```

Markers Properties Servers Data Source Explorer Debug Terminal JUnit

Finished after 0.028 seconds

Runs: 4/4 Errors: 0 Failures: 0

Failure Trace

Let's make the test for values being saved a bit more elaborate. I have renamed my test to better tell what it is actually testing...

```

Java EE - simple-map/src/test/java/org/self/learn/test/SimpleMapTest.java - Spring Tool Suite
File Edit Source Refactor Navigate Search Project Run Window Help
SimpleMapTest.java simple-map/pom.xml
30 }
31
32 @Test
33 public void itShouldBeAbleToReturnSizeOfTheMap() {
34     map.add(1, "Priyan");
35     assertThat(map.size(), isEqualTo(1));
36     map.add(5, "Priyadarshan");
37     assertThat(map.size(), isEqualTo(2));
38     map.add(200, "Priyan Parida");
39     assertThat(map.size(), isEqualTo(3));
40 }
41
42 @Test
43 public void itShouldSaveAKeyAndValuePair() throws Exception {
44     map.add(1, "Priyan");
45     assertThat(map.get(1), is("Priyan"));
46 }
47
48 @Test
49 public void itShouldSaveMultipleKeyAndValuePairs() {
50 }
51 }

SimpleMap.java
5     private K key;
6     private V value;
7
8     public SimpleMap() {
9         this.size = 0;
10    }
11
12     public int size() {
13         return this.size;
14    }
15
16     public void add(K key, V value) {
17         this.key = key;
18         this.value = value;
19         this.size++;
20    }
21
22     public V get(K key) {
23         return this.value;
24    }
25
26 }

Markers Properties Servers Data Source Explorer Debug Terminal JUnit
Finished after 0.028 seconds
Runs: 4/4 Errors: 0 Failures: 0
org.self.learn.test.SimpleMapTest [Runner:JUnit4] (0.000 s)
Failure Trace

```

We want to test the map is able to save, not just a single key/value, but multiple keys/values. The test below gives an illusion that it does.

But does it really? Check the next test.

```

Java EE - simple-map/src/test/java/org/self/learn/test/SimpleMapTest.java - Spring Tool Suite
File Edit Source Refactor Navigate Project Run Window Help
SimpleMapTest.java simple-map/pom.xml
30 }
31
32 @Test
33 public void itShouldBeAbleToInitializeAnEmptyMapWithSizeZero() {
34     map.add(1, "Priyan");
35     assertThat(map.size(), isEqualTo(1));
36     map.add(5, "Priyadarshan");
37     assertThat(map.size(), isEqualTo(2));
38     map.add(200, "Priyan Parida");
39     assertThat(map.size(), isEqualTo(3));
40 }
41
42 @Test
43 public void itShouldSaveAKeyAndValuePair() throws Exception {
44     map.add(1, "Priyan");
45     assertThat(map.get(1), is("Priyan"));
46 }
47
48 @Test
49 public void itShouldSaveMultipleKeyAndValuePairs() {
50     map.add(1, "Priyan");
51     map.add(2, "Priyadarshan");
52     assertThat(map.get(2), is("Priyadarshan"));
53 }

SimpleMap.java
8     public SimpleMap() {
9         this.size = 0;
10    }
11
12     public int size() {
13         return this.size;
14    }
15
16     public void add(K key, V value) {
17         this.key = key;
18         this.value = value;
19         this.size++;
20    }
21
22     public V get(K key) {
23         return this.value;
24    }
25
26 }

Markers Properties Servers Data Source Explorer Debug Terminal JUnit
Finished after 0.029 seconds
Runs: 5/5 Errors: 0 Failures: 0
org.self.learn.test.SimpleMapTest [Runner:JUnit4] (0.000 s)
itShouldBeAbleToInitializeAnEmptyMapWithSizeZero (0.000 s)
itShouldBeAbleToReturnSizeOfTheMap (0.000 s)
itShouldSaveMultipleKeyAndValuePairs (0.000 s)
itShouldReturnSizeGreaterThanZeroUponAddingKeyValuePair (0.000 s)
itShouldSaveAKeyAndValuePair (0.000 s)

```

There we go... This test caught us. ☺ Expected “Priyan” but was “Priyadarshan”. Something is definitely wrong.

```

Java EE - simple-map/src/test/java/org/self/learn/test/SimpleMapTest.java - Spring Tool Suite
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access Java EE Debug Git
SimpleMapTest.java
36     map.size(), isEqualTo(1));
37     assertEquals(map.size(), isEqualTo(2));
38     map.add("Priyan Parida");
39     assertEquals(map.size(), isEqualTo(3));
40 }
41
42 @Test
43 public void itShouldSaveKeyAndValuePair() throws Exception {
44     map.add("Priyan");
45     assertEquals(map.get("Priyan"), "Priyan");
46 }
47
48 @Test
49 public void itShouldSaveMultipleKeyAndValuePairs() {
50     map.add("Priyan");
51     map.add("Priyadarshan");
52     assertEquals(map.get("Priyan"), "Priyan");
53     assertEquals(map.get("Priyadarshan"), "Priyadarshan");
54 }
55
56 }

SimpleMap.java
8     public SimpleMap() {
9         this.size = 0;
10    }
11
12    public int size() {
13        return this.size;
14    }
15
16    public void add(K key, V value) {
17        this.key = key;
18        this.value = value;
19        this.size++;
20    }
21
22    public V get(K key) {
23        return this.value;
24    }
25
26    public void remove(K key) {
27        this.size--;
28    }
29
30    public void clear() {
31        this.size = 0;
32    }
33
34    public int hashCode() {
35        return this.size;
36    }
37
38    public boolean equals(Object o) {
39        if (this == o) {
40            return true;
41        }
42        if (o instanceof SimpleMap) {
43            SimpleMap simpleMap = (SimpleMap) o;
44            return this.size == simpleMap.size();
45        }
46        return false;
47    }
48
49    @Override
50    public String toString() {
51        return "SimpleMap{" + "size=" + size + '}';
52    }
53
54 }

Markers Properties Servers Data Source Explorer Debug Terminal JUnit
Finished after 0.035 seconds
Runs: 5/5 Errors: 0 Failures: 1
Failure Trace
java.lang.AssertionError:
Expected: is "Priyan"
but was "Priyadarshan"
at org.hamcrest.MatcherAssert.assertThat(MatcherAssert.java:20)
at org.self.learn.test.SimpleMapTest.itShouldSaveMultipleKeyAndValuePairs(SimpleMapTest.java:53)

```

Let's go a few miles to actually store keys/values into the map.

```

New Java Class
Java Class
Create a new Java class.

Source folder: simple-map/src/main/java
Package: org.self.learn
Enclosing type: 

Name: KeyValuePair<K, V>
Modifiers: public package private protected
Abstract final static
Superclass: java.lang.Object
Interfaces: 

Which method stubs would you like to create?
public static void main(String[] args)
Constructors from superclass
Inherited abstract methods
Do you want to add comments? (Configure templates and default value here)
Generate comments

earns;
eMap() {
ze;
e;
ap() {
= 0;

e() {
s.size;

d(K key, V value) {
key;
= value;
+;


```

Here is what my Key Value Template looks like.

The screenshot shows the Java EE - simple-map project in the Spring Tool Suite. The Project Explorer view on the left lists various Java files and resources. The central workspace displays two open files: SimpleMapTest.java and SimpleMap.java.

SimpleMapTest.java

```
1 package org.self.learn;
2
3 import static org.junit.Assert.assertEquals;
4 import static org.junit.Assert.assertTrue;
5
6 import org.junit.Test;
7
8 public class SimpleMapTest {
9
10     @Test
11     public void itShouldBeAbleToInitializeAnEmptyMapWithSizeZero() {
12         assertEquals(0, map.size());
13     }
14
15     @Test
16     public void itShouldReturnSizeGreaterThanOrEqualToZeroUponAddingKeyValuePair() {
17         map.add(1, "Priyan");
18         assertTrue(map.size() >= 0);
19     }
20
21     @Test
22     public void itShouldBeAbleToReturnSizeOfTheMap() {
23         map.add(1, "Priyan");
24         assertEquals(1, map.size());
25         map.add(5, "Priyadarshan");
26         assertEquals(2, map.size());
27         map.add(200, "Priyan Parida");
28         assertEquals(3, map.size());
29     }
30
31     @Test
32     public void itShouldSaveAKeyValuePair() throws Exception {
33         map.add(1, "Priyan");
34         assertEquals("Priyan", map.get(1));
35     }
36
37     @Test
38     public void itShouldSaveMultipleKeyAndValuePairs() {
39         map.add(1, "Priyan");
40         map.add(2, "Priyadarshan");
41         assertEquals("Priyadarshan", map.get(2));
42         assertEquals("Priyan", map.get(1));
43     }
44 }
```

SimpleMap.java

```
1 package org.self.learn;
2
3 public class SimpleMap {
4
5     private Map<Integer, String> map = new HashMap<Integer, String>();
6
7     public void add(Integer key, String value) {
8         map.put(key, value);
9     }
10
11     public String get(Integer key) {
12         return map.get(key);
13     }
14 }
```

Added array of key value template as a member variable, saved keys/values into the array. The "get()" method now returns the value corresponding to the key.

The screenshot shows the Java EE - simple-map project in Spring Tool Suite. The left pane displays `SimpleMapTest.java`, which contains several test cases for the `SimpleMap` class. The right pane displays the source code for `SimpleMap.java`. The code implements a map with `K` and `V` types, adding key-value pairs and returning their sizes.

```
1 package org.self.learn;
2
3 public class SimpleMap<K, V> {
4     private int size;
5     private KeyValuePair<K, V>[] keyValuePair;
6
7     @SuppressWarnings("unchecked")
8     public SimpleMap() {
9         this.size = 0;
10        this.keyValuePair = (KeyValuePair<K, V>[] ) new KeyValuePair[5];
11    }
12
13    public int size() {
14        return this.size;
15    }
16
17    public void add(K key, V value) {
18        this.keyValuePair[this.size++] = new KeyValuePair<K, V>(key, value);
19    }
20
21    public V get(K key) {
22        KeyValuePair<K, V> keyValuePair;
23
24        for(int i=0;i<size;i++) {
25            keyValuePair = this.keyValuePair[i];
26            if(key.equals(keyValuePair.getKey())) {
27                return keyValuePair.getValue();
28            }
29        }
30
31        return null;
32    }
33
34    }
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56 }
```

I run the tests and they complete successfully this time.

The screenshot shows the Java EE - simple-map project in Spring Tool Suite. The left pane displays `SimpleMapTest.java` with several test methods for the `SimpleMap` class. The right pane shows the source code for `SimpleMap.java`, specifically the implementation of the `get` method. The code uses a `KeyValuePairs` array to store key-value pairs and iterates through it to find a matching key. The interface for `SimpleMap` is also visible at the top.

```
Java EE - simple-map/src/main/java/org/self/learn/SimpleMap.java - Spring Tool Suite
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access Java EE Debug Git
SimpleMapTest.java
15
16
17 public void add(K key, V value) {
18     this.keyValuePair[this.size++] = new KeyValuePair<K, V>(key, value);
19 }
20
21 public V get(K key) {
22     KeyValuePair<K, V> keyValuePair;
23
24     for(int i=0;i<size;i++) {
25         keyValuePair = this.keyValuePair[i];
26         if(key.equals(keyValuePair.getKey())) {
27             return keyValuePair.getValue();
28         }
29     }
30
31     return null;
32 }
33
34 }
35
36
37 @Test
38 public void itShouldInitializeAnEmptyMapWithSizeZero() {
39     map.add(200, "Priyan Parida");
40     assertEquals(map.size(), 0);
41 }
42
43 @Test
44 public void itShouldSaveKeyAndValuePair() throws Exception {
45     map.add(1, "Priyan");
46     assertEquals(map.get(1), "Priyan");
47 }
48
49 @Test
50 public void itShouldSaveMultipleKeyAndValuePairs() {
51     map.add(1, "Priyan");
52     map.add(2, "Priyadarshan");
53     assertEquals(map.get(2), "Priyadarshan");
54     assertEquals(map.get(1), "Priyan");
55 }
56 }
```

Next, let's write test cases for "containsKey"

The screenshot shows the Java EE - simple-map/src/test/java/org/self/learn/test/SimpleMapTestJava - Spring Tool Suite interface. On the left, the `SimpleMapTest.java` file is open, showing several test methods for the `SimpleMap` class. In the middle, the `SimpleMap.java` file is open, showing the implementation of the `SimpleMap` class. A code completion tooltip is displayed over the `containsKey(int)` method stub in the `SimpleMap` class. The tooltip contains the following options:

- Create method 'containsKey(int)' in type 'SimpleMap'
- Add cast to 'map'
- Rename in file (Ctrl+2, R)

The status bar at the bottom displays the message: "The method containsKey(int) is undefined for the type SimpleMap<Integer, String>".

Successful completion of the test gives us an illusion that the implementation is complete. But, it is not. Let's add another test case to prove this.

The screenshot shows the Spring Tool Suite interface. On the left, the code editor displays `SimpleMapTest.java` and `Matchers.class`. In the center, the code editor displays `SimpleMap.java`. The right panel shows the JUnit results for the run after 0.032 seconds. The results table has three columns: Runs: 6/6, Errors: 0, and Failures: 0. The failures section is empty. Below the table, the Failure Trace shows several test methods from `org.self.learn.test.SimpleMapTest` that passed, including `itShouldBeAbleToInitializeAnEmptyMapWithSizeZero`, `itShouldIndicateIfGivenKeyIsPartOfTheMap`, `itShouldBeAbleToReturnSizeOfTheMap`, `itShouldSaveMultipleKeyAndValuePairs`, `itShouldReturnSizeGreaterThanZeroUponAddingKeyValuePair`, and `itShouldSaveAKeyAndValuePair`.

The test fails when ran now. We know returning “true” always isn’t the right implementation. So let’s go fix it.

The screenshot shows the Spring Tool Suite interface after modifying the `containsKey` method. The code editor on the right now contains the following implementation:

```

    public boolean containsKey(K key) {
        KeyValPair<K, V> keyValPair;
        for(int i=0;i<size;i++) {
            keyValPair = this.keyValuePair[i];
            if(key.equals(keyValPair.getKey())) {
                return keyValPair.getValue();
            }
        }
        return null;
    }

```

The JUnit results table shows a failure: Failures: 1. The Failure Trace indicates a `java.lang.AssertionError` at `org.self.learn.test.SimpleMapTest.itShouldIndicateIfGivenKeyIsPartOfTheMap(SimpleMapTest.java:60)`. This means the test `itShouldIndicateIfGivenKeyIsPartOfTheMap` failed because the implementation returned `true` for a key that was not present in the map.

The test result indicates that the implementation is complete now.

```

Java EE - simple-map/src/main/java/org/self/learn/SimpleMap.java - Spring Tool Suite
File Edit Source Refactor Navigate Project Run Window Help
Quick Access Java EE Debug Git
SimpleMapTest.java
    public void itShouldInitializeAnEmptyMapWithSizeZero() throws Exception {
        map.add(1, "Priyan");
        assertEquals(map.get(1), "Priyan");
    }
    @Test
    public void itShouldSaveMultipleKeyAndValuePairs() {
        map.add(1, "Priyan");
        map.add(2, "Priyadarshan");
        assertEquals(map.get(2), "Priyadarshan");
        assertEquals(map.get(1), "Priyan");
    }
    @Test
    public void itShouldIndicateIfGivenKeyIsPartOfTheMap() throws Exception {
        map.add(1, "Priyan");
        assertTrue(map.containsKey(1));
        assertFalse(map.containsKey(500));
        assertFalse(map.containsKey(51), !map.containsKey(51));
    }
}

```

```

SimpleMap.java
    for(int i=0;i<size;i++) {
        keyValuePair = this.keyValuePair[i];
        if(key.equals(keyValuePair.getKey())) {
            return keyValuePair.getValue();
        }
    }
    return null;
}
public boolean containsKey(K key) {
    KeyValuePair<K, V> keyValuePair;
    for(int i=0;i<size;i++) {
        keyValuePair = this.keyValuePair[i];
        if(key.equals(keyValuePair.getKey())) {
            return true;
        }
    }
    return false;
}

```

Markers Properties Servers Data Source Explorer Debug Terminal JUnit Finished after 0.029 seconds Runs: 6/6 Errors: 0 Failures: 0 Failure Trace

Next, let's test the edge cases w.r.t. getKey(). We want it to throw an exception when a key isn't found. Note how the tests drive the implementation.

We verify exception using a Junit Rule. It's one of the several ways of testing this.

```

Java EE - simple-map/src/test/java/org/self/learn/test/SimpleMapTest.java - Spring Tool Suite
File Edit Source Refactor Navigate Project Run Window Help
Quick Access Java EE Debug Git
SimpleMapTest.java
    package org.self.learn.test;
    import static org.hamcrest.Matchers.*;
    @RunWith(BlockJUnit4ClassRunner.class)
    public class SimpleMapTest {
        private SimpleMap<Integer, String> map;
        @Rule
        public ExpectedException exception = ExpectedException.none();
        @Before
        public void init() {
            map = new SimpleMap<>();
        }
        @Test
        public void itShouldBeAbleToInitializeAnEmptyMapWithSizeZero() {
            assertEquals(map.size(), 0);
        }
        @Test
        public void itShouldReturnSizeGreaterThanZeroUponAddingKeyValuePair() {
            map.add(1, "Priyan");
            assertEquals(map.size(), 1);
            map.add(5, "Priyadarshan");
            assertEquals(map.size(), 2);
            map.add(200, "Priyan Parida");
            assertEquals(map.size(), 3);
        }
    }

```

```

SimpleMap.java
    package org.self.learn;
    public class SimpleMap<K, V> {
        private int size;
        private KeyValuePair<K, V>[] keyValuePair;
        @SuppressWarnings("unchecked")
        public SimpleMap() {
            this.size = 0;
            this.keyValuePair = (KeyValuePair<K, V>) new KeyValuePair[5];
        }
        public int size() {
            return this.size;
        }
        public void add(K key, V value) {
            this.keyValuePair[this.size++] = new KeyValuePair<K, V>(key, value);
        }
        public V get(K key) {
            KeyValuePair<K, V> keyValuePair;
            for(int i=0;i<size;i++) {
                keyValuePair = this.keyValuePair[i];
                if(key.equals(keyValuePair.getKey())) {
                    return keyValuePair.getValue();
                }
            }
            return null;
        }
        public boolean containsKey(K key) {
            KeyValuePair<K, V> keyValuePair;
            for(int i=0;i<size;i++) {

```

Next we write test and create the custom exception object..

Java EE - simple-map/src/test/java/org/self/learn/test/SimpleMapTest.java - Spring Tool Suite

```

40     assertThat(map.size(), isEqualTo(1));
41     map.add(5, "Priyadarshan");
42     assertThat(map.size(), isEqualTo(2));
43     map.add(200, "Priyan Parida");
44     assertThat(map.size(), isEqualTo(3));
45 }
46
47 @Test
48 public void itShouldSaveKeyValuePair() throws Exception {
49     map.add(1, "Priyan");
50     assertThat(map.get(1), isEqualTo("Priyan"));
51 }
52
53 @Test
54 public void itShouldSaveMultipleKeyValuePair() {
55     map.add(1, "Priya");
56     map.add(2, "Priya");
57     assertThat(map.get(1), Create class 'NoSuchKeyFoundException');
58     assertThat(map.get(2), Create interface 'NoSuchKeyFoundException');
59 }
60
61 @Test
62 public void itShouldDistinguishBetweenAddAndGet() {
63     map.add(1, "Priya");
64     assertTrue(map.containsKey(1));
65     assertFalse(map.containsValue("Priya"));
66     assertThat(map.get(1), Create class 'NoSuchKeyFoundException');
67 }
68
69 @Test
70 public void itShouldThrowExceptionWhenGettingNonExistentKey() {
71     map.add(1, "Priya");
72     exception.expect(NoSuchKeyFoundException.class);
73     map.get(45);
74 }
75 }
    
```

NoSuchKeyFoundException cannot be resolved to a type

Compilation errors are gone...

Java EE - simple-map/src/test/java/org/self/learn/test/SimpleMapTest.java - Spring Tool Suite

```

41     assertThat(map.size(), isEqualTo(1));
42     map.add(5, "Priyadarshan");
43     assertThat(map.size(), isEqualTo(2));
44     map.add(200, "Priyan Parida");
45     assertThat(map.size(), isEqualTo(3));
46 }
47
48 @Test
49 public void itShouldSaveKeyValuePair() throws Exception {
50     map.add(1, "Priyan");
51     assertThat(map.get(1), isEqualTo("Priyan"));
52 }
53
54 @Test
55 public void itShouldSaveMultipleKeyValuePair() {
56     map.add(1, "Priyan");
57     map.add(2, "Priyadarshan");
58     assertThat(map.get(2), isEqualTo("Priyadarshan"));
59     assertThat(map.get(1), isEqualTo("Priyan"));
60 }
61
62 @Test
63 public void itShouldIndicateIfGivenKeyIsPartOfTheMap() {
64     map.add(1, "Priyan");
65     assertTrue(map.containsKey(1));
66     assertFalse(map.containsKey(500));
67     assertThat(map.containsValue(51), is(not(equalTo(true))));
68 }
69
70 @Test
71 public void itShouldThrowNoSuchKeyFoundException() throws Exception {
72     map.add(1, "Priyan");
73     exception.expect(NoSuchKeyFoundException.class);
74     map.get(45);
75 }
    
```

With the compilation issues resolved, we are ready to run the tests now. The test fails as we expected it to.

The screenshot shows the Spring Tool Suite interface with two code editors open. The left editor contains `SimpleMapTest.java` and the right editor contains `SimpleMap.java`. Below the editors is a toolbar with various icons and a status bar indicating "Finished after 0.047 seconds". At the bottom, a results table shows "Runs: 7/7" and "Failures: 0". A "Failure Trace" section is visible on the right.

```

SimpleMapTest.java
54     map.add(1, "Priyan");
55     map.add(2, "Priyadarshan");
56     assertEquals(map.get(2), isEqualTo("Priyadarshan"));
57     assertEquals(map.get(1), isEqualTo("Priyan"));
58 }
59
60 @Test
61 public void itShouldIndicateIfGivenKeyIsPartOfTheMap() {
62     map.add(1, "Priyan");
63     assertTrue(map.containsKey(1));
64     assertFalse(map.containsKey(500));
65     assertFalse(map.containsKey(51), is(not_equalTo(true)));
66 }
67
68 @Test
69 public void itShouldThrowNoSuchKeyFoundException() throws Exception {
70     map.add(1, "Priyan");
71     exception.expect(NoSuchKeyFoundException.class);
72     map.get(45);
73 }
74
75 }

SimpleMap.java
1 package org.self.learn;
2
3 public class SimpleMap<K, V> {
4     private int size;
5     private KeyValuePair<K, V>[] keyValuePairs;
6
7     @SuppressWarnings("unchecked")
8     public SimpleMap() {
9         this.size = 0;
10    this.keyValuePairs = (KeyValuePair<K, V>[] ) new KeyValuePair[5];
11 }
12
13 public int size() {
14     return this.size;
15 }
16
17 public void add(K key, V value) {
18     this.keyValuePairs[this.size++] = new KeyValuePair<K, V>(key, value);
19 }
20
21 public V get(K key) throws NoSuchKeyFoundException {
22     KeyValuePair<K, V> keyValuePair;
23
24     if(!containsKey(key)) {
25         throw new NoSuchKeyFoundException();
26     }
27
28     for(int i=0;i<size;i++) {
29         keyValuePair = this.keyValuePairs[i];
30         if(key.equals(keyValuePair.getKey())) {
31             return keyValuePair.getValue();
32         }
33     }
34
35     return null;
36 }
37
38 }

Markers Properties Servers Data Source Explorer Debug Terminal JUnit

```

Let's quickly get this working... The test completes successfully.

The screenshot shows the Spring Tool Suite interface with two code editors open. The left editor contains `SimpleMapTest.java` and the right editor contains `SimpleMap.java`. Below the editors is a toolbar with various icons and a status bar indicating "Finished after 0.037 seconds". At the bottom, a results table shows "Runs: 7/7" and "Failures: 0". A "Failure Trace" section is visible on the right.

```

SimpleMapTest.java
54     map.add(1, "Priyan");
55     map.add(2, "Priyadarshan");
56     assertEquals(map.get(2), isEqualTo("Priyadarshan"));
57     assertEquals(map.get(1), isEqualTo("Priyan"));
58 }
59
60 @Test
61 public void itShouldIndicateIfGivenKeyIsPartOfTheMap() {
62     map.add(1, "Priyan");
63     assertTrue(map.containsKey(1));
64     assertFalse(map.containsKey(500));
65     assertFalse(map.containsKey(51), is(not_equalTo(true)));
66 }
67
68 @Test
69 public void itShouldThrowNoSuchKeyFoundException() throws Exception {
70     map.add(1, "Priyan");
71     exception.expect(NoSuchKeyFoundException.class);
72     map.get(45);
73 }
74
75 }

SimpleMap.java
1 package org.self.learn;
2
3 public class SimpleMap<K, V> {
4     private int size;
5     private KeyValuePair<K, V>[] keyValuePairs;
6
7     @SuppressWarnings("unchecked")
8     public SimpleMap() {
9         this.size = 0;
10    this.keyValuePairs = (KeyValuePair<K, V>[] ) new KeyValuePair[5];
11 }
12
13 public int size() {
14     return this.size;
15 }
16
17 public void add(K key, V value) {
18     this.keyValuePairs[this.size++] = new KeyValuePair<K, V>(key, value);
19 }
20
21 public V get(K key) throws NoSuchKeyFoundException {
22     KeyValuePair<K, V> keyValuePair;
23
24     if(!containsKey(key)) {
25         throw new NoSuchKeyFoundException();
26     }
27
28     for(int i=0;i<size;i++) {
29         keyValuePair = this.keyValuePairs[i];
30         if(key.equals(keyValuePair.getKey())) {
31             return keyValuePair.getValue();
32         }
33     }
34
35     return null;
36 }
37
38 }

Markers Properties Servers Data Source Explorer Debug Terminal JUnit

```

Next up is test/implementation for "containsValue()"

```

Java EE - simple-map/src/test/java/org/self/learn/test/SimpleMapTest.java - Spring Tool Suite
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access Java EE Debug Git
SimpleMapTest.java
68 }
69 }
70 @Test
71 public void itShouldThrowNoSuchKeyFoundException() throws Exception {
72     map.add(1, "Priyan");
73     exception.expect(NoSuchKeyFoundException.class);
74     map.get(45);
75 }
76
77 @Test
78 public void itShouldIndicateIfGivenValueIsPartOfTheMap() {
79     map.add(1, "Priyan");
80     map.add(2, "Priyan");
81     map.add(3, "Priyadarshan");
82     assertTrue(map.containsValue("Priyan"));
83     assertTrue(map.containsValue("Priyadarshan"));
84     assertFalse(map.containsValue("Priyan Parida"));
85 }
86 }

SimpleMap.java
18     this.keyValuePair[this.size+] = new KeyValuePair<K, V>(key, value);
19 }
20
21 public V get(K key) throws NoSuchKeyFoundException {
22     KeyValuePair<K, V> keyValuePair;
23
24     if(!containsKey(key)) {
25         throw new NoSuchKeyFoundException();
26     }
27
28     for(int i=0;i<size;i++) {
29         keyValuePair = this.keyValuePair[i];
30         if(key.equals(keyValuePair.getKey())) {
31             return keyValuePair.getValue();
32         }
33     }
34 }

Press 'Tab' from proposal table or click for focus

```

Markers Properties Servers Data Source Explorer Debug Terminal JUnit

Finished after 0.037 seconds

Runs: 7/7

org.self.learn.test.SimpleMapTest

- itShouldBeAbleToInitializeA
- itShouldIndicateIfGivenKeyIsPartOfTheMap
- itShouldThrowNoSuchKeyFoundException
- itShouldBeAbleToReturnSizeOfTheMap
- itShouldIndicateIfGivenValueIsPartOfTheMap
- itShouldSaveMultipleKeyAndValuePairs (0.000 s)
- itShouldReturnSizeGreaterThanZeroUponAddingKeyValuePair (0.002 s)
- itShouldSaveAKeyAndValuePair (0.001 s)

The method containsValue(String) is undefined for the type SimpleMap<Integer, String>

I have skipped a couple of steps here i.e. auto generate method, implement the smallest bit, run test and start over with the next bit of implementation... Here is what the final test and implementation looks like. That took care of it, the test results indicate that.

```

Java EE - simple-map/src/test/java/org/self/learn/test/SimpleMapTest.java - Spring Tool Suite
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access Java EE Debug Git
SimpleMapTest.java
68 }
69 }
70 @Test
71 public void itShouldThrowNoSuchKeyFoundException() throws Exception {
72     map.add(1, "Priyan");
73     exception.expect(NoSuchKeyFoundException.class);
74     map.get(45);
75 }
76
77 @Test
78 public void itShouldIndicateIfGivenValueIsPartOfTheMap() {
79     map.add(1, "Priyan");
80     map.add(2, "Priyan");
81     map.add(3, "Priyadarshan");
82     assertTrue(map.containsValue("Priyan"));
83     assertTrue(map.containsValue("Priyadarshan"));
84     assertFalse(map.containsValue("Priyan Parida"));
85 }
86 }

SimpleMap.java
18     this.keyValuePair[this.size+] = new KeyValuePair<K, V>(key, value);
19 }
20
21 public V get(K key) throws NoSuchKeyFoundException {
22     KeyValuePair<K, V> keyValuePair;
23
24     if(!containsKey(key)) {
25         throw new NoSuchKeyFoundException();
26     }
27
28     for(int i=0;i<size;i++) {
29         keyValuePair = this.keyValuePair[i];
30         if(key.equals(keyValuePair.getKey())) {
31             return keyValuePair.getValue();
32         }
33     }
34 }

50 public boolean containsValue(V value) {
51     KeyValuePair<K, V> keyValuePair;
52     for(int i=0;i<size;i++) {
53         keyValuePair = this.keyValuePair[i];
54         if(value.equals(keyValuePair.getValue())) {
55             return true;
56         }
57     }
58 }

Press 'Tab' from proposal table or click for focus

```

Markers Properties Servers Data Source Explorer Debug Terminal JUnit

Finished after 0.039 seconds

Runs: 8/8 Errors: 0 Failures: 0

Failure Trace

org.self.learn.test.SimpleMapTest [Runner: JUnit 4] (0.014 s)

- itShouldBeAbleToInitializeAnEmptyMapWithSizeZero (0.002 s)
- itShouldIndicateIfGivenKeyIsPartOfTheMap (0.001 s)
- itShouldThrowNoSuchKeyFoundException (0.008 s)
- itShouldBeAbleToReturnSizeOfTheMap (0.000 s)
- itShouldIndicateIfGivenValueIsPartOfTheMap (0.000 s)
- itShouldSaveMultipleKeyAndValuePairs (0.000 s)
- itShouldReturnSizeGreaterThanZeroUponAddingKeyValuePair (0.002 s)
- itShouldSaveAKeyAndValuePair (0.001 s)

We do not want to allow our map to contain duplicate key and throw an exception when that is the case. Let's write a test for that.

```

Java EE - simple-map/src/test/java/org/self/learn/test/SimpleMapTest.java - Spring Tool Suite
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access Java EE Debug Git
SimpleMapTest.java
75 }
76
77 @Test
78 public void itShouldIndicateIfGivenValueIsPartOfTheMap() {
79     map.add(1, "Priyan");
80     map.add(2, "Priyan");
81     map.add(3, "Priyadarshan");
82     assertTrue(map.containsValue("Priyan"));
83     assertTrue(map.containsValue("Priyadarshan"));
84     assertFalse(map.containsValue("Priyan Parida"));
85 }
86
87 @Test
88 public void testDuplicateKeyFoundException() throws NoSuchKeyNotFoundException {
89     map.add(1, "Priyan");
90     exception.expect(DuplicateKeyFoundException.class);
91     map.add(1, "Priyadarshan");
92 }
93 }

SimpleMap.java
10         this.keyValuePair = (KeyValuePair<K, V>[]) new KeyValuePair[5];
11     }
12
13     public int size() {
14         return this.size;
15     }
16
17     public void add(K key, V value) {
18         this.keyValuePair[this.size++] = new KeyValuePair(key, value);
19     }
20
21     public V get(K key) throws NoSuchKeyNotFoundException {
22         KeyValuePair<K, V> keyValuePair;
23
24         if(!containsKey(key)) {
25             throw new NoSuchKeyNotFoundException();
26         }
27     }

```

Markers Properties Servers Finished after 0.038 seconds Runs: 9/9 org.self.learn.test.SimpleMapTest

DuplicateKeyFoundException cannot be resolved to a type Writable Smart Insert 90:52

2:09 AM 2/24/2016

Create the custom exception class to resolve compilation error...

```

Java EE - simple-map/src/main/java/org/self/learn/DuplicateKeyFoundException.java - Spring Tool Suite
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access Java EE Debug Git
SimpleMapTest.java DuplicateKeyFoundException.java
1 package org.self.learn;
2
3 public class DuplicateKeyFoundException extends Exception {
4     private static final long serialVersionUID = 3949470449695281086L;
5 }
6

SimpleMap.java
76 }
77
78 @Test
79 public void itShouldIndicateIfGivenValueIsPartOfTheMap() {
80     map.add(1, "Priyan");
81     map.add(2, "Priyan");
82     map.add(3, "Priyadarshan");
83     assertTrue(map.containsValue("Priyan"));
84     assertTrue(map.containsValue("Priyadarshan"));
85     assertFalse(map.containsValue("Priyan Parida"));
86 }
87
88 @Test
89 public void testDuplicateKeyFoundException() throws NoSuchKeyNotFoundException {
90     map.add(1, "Priyan");
91     exception.expect(DuplicateKeyFoundException.class);
92     map.add(1, "Priyadarshan");
93 }

SimpleMap.java
1 package org.self.learn;
2
3 public class DuplicateKeyFoundException extends Exception {
4     private static final long serialVersionUID = 3949470449695281086L;
5 }
6


```

Markers Properties Servers Data Source Explorer Debug Terminal JUnit Failure Trace

Finished after 0.038 seconds Runs: 9/9 Errors: 0 Failures: 0 0.015 s org.self.learn.test.SimpleMapTest [Runner: JUnit 4] 3:40

The test fails because we are not surfacing an exception yet.

The screenshot shows the Spring Tool Suite interface. In the top-left pane, the test class `SimpleMapTest.java` is displayed with several test methods. In the top-right pane, the source code for `SimpleMap.java` is shown. The bottom section displays the JUnit results, indicating 9/9 runs, 0 errors, and 1 failure. The failure details show a `java.lang.AssertionError` with the message "Expected test to throw an instance of org.self.learn.DuplicateKeyNotFoundException".

```

Java EE - simple-map/src/test/java/org/self/learn/test/SimpleMapTest.java - Spring Tool Suite
File Edit Source Refactor Navigate Project Run Window Help
Quick Access Java EE Debug Git
SimpleMapTest.java
76 }
77
78@ Test
79 public void itShouldIndicateIfGivenValueIsPartOfTheMap() {
80     map.add(1, "Priyan");
81     map.add(2, "Priyan");
82     map.add(3, "Priyadarshan");
83     assertTrue(map.containsValue("Priyan"));
84     assertTrue(map.containsValue("Priyadarshan"));
85     assertFalse(map.containsValue("Priyan Parida"));
86 }
87
88@ Test
89 public void testDuplicateKeyFoundException() throws NoSuchKeyNotFoundException {
90     map.add(1, "Priyan");
91     exception.expect(DuplicateKeyNotFoundException.class);
92     map.add(1, "Priyadarshan");
93 }
94 }

SimpleMap.java
10     this.keyValuePair = (KeyValuePair<K, V>[]) new KeyValuePair[5];
11 }
12
13@ Public int size() {
14     return this.size;
15 }
16
17@ Public void add(K key, V value) {
18     this.keyValuePair[this.size++] = new KeyValuePair<K, V>(key, value);
19 }
20
21@ Public V get(K key) throws NoSuchKeyNotFoundException {
22     KeyValuePair<K, V> keyValuePair;
23
24     if(!containsKey(key)) {
25         throw new NoSuchKeyNotFoundException();
26     }
27 }

Markers Properties Servers Data Source Explorer Debug Terminal JUnit
Finished after 0.057 seconds
Runs: 9/9 Errors: 0 Failures: 1
Failure Trace
java.lang.AssertionError: Expected test to throw an instance of org.self.learn.DuplicateKeyNotFoundException

```

That will fix it, the tests are all green now.

The screenshot shows the Spring Tool Suite interface. The code has been modified in `SimpleMapTest.java` to handle the exception correctly. The JUnit results now show 9/9 runs, 0 errors, and 0 failures, indicating all tests are green.

```

Java EE - simple-map/src/test/java/org/self/learn/test/SimpleMapTest.java - Spring Tool Suite
File Edit Source Refactor Navigate Project Run Window Help
Quick Access Java EE Debug Git
SimpleMapTest.java
83     public void itShouldIndicateIfGivenValueIsPartOfTheMap() throws DuplicateKeyException {
84         map.add(1, "Priyan");
85         map.add(2, "Priyan");
86         map.add(3, "Priyadarshan");
87         assertTrue(map.containsValue("Priyan"));
88         assertTrue(map.containsValue("Priyadarshan"));
89         assertFalse(map.containsValue("Priyan Parida"));
90     }
91
92@ Test
93 public void testDuplicateKeyFoundException()
94     throws NoSuchKeyNotFoundException, DuplicateKeyNotFoundException {
95     map.add(1, "Priyan");
96     exception.expect(DuplicateKeyNotFoundException.class);
97     map.add(1, "Priyadarshan");
98 }
99

SimpleMap.java
8     public SimpleMap() {
9         this.size = 0;
10        this.keyValuePair = (KeyValuePair<K, V>[]) new KeyValuePair[5];
11    }
12
13@ Public int size() {
14     return this.size;
15 }
16
17@ Public void add(K key, V value) throws DuplicateKeyNotFoundException {
18     if(containsKey(key)) {
19         throw new DuplicateKeyNotFoundException();
20     }
21
22     this.keyValuePair[this.size++] = new KeyValuePair<K, V>(key, value);
23 }

Markers Properties Servers Data Source Explorer Debug Terminal JUnit
Finished after 0.039 seconds
Runs: 9/9 Errors: 0 Failures: 0
Failure Trace

```

Btw, here is another way of asserting exceptions

```

Java EE - simple-map/src/test/java/org/self/learn/test/SimpleMapTest.java - Spring Tool Suite
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access Java EE Debug Git
SimpleMapTest.java
83     public void itShouldIndicateIfGivenValueIsPartOfTheMap() throws DuplicateKeyF
84         map.add(1, "Priyan");
85         map.add(2, "Priyan");
86         map.add(3, "Priyadarshan");
87         assertTrue(map.containsValue("Priyan"));
88         assertTrue(map.containsValue("Priyadarshan"));
89         assertFalse(map.containsValue("Priyan Parida"));
90     }
91
92     @Test(expected = DuplicateKeyFoundException.class)
93     public void itShouldThrowAnExceptionWhenDuplicateKeyIsEntered()
94         throws NoSuchKeyFoundException, DuplicateKeyFoundException {
95         map.add(1, "Priyan");
96         map.add(1, "Priyadarshan");
97     }
98 }

SimpleMap.java
8     public SimpleMap() {
9         this.size = 0;
10        this.keyValuePair = (KeyValuePair<K, V>[][]) new KeyValuePair[5];
11    }
12
13    public int size() {
14        return this.size;
15    }
16
17    public void add(K key, V value) throws DuplicateKeyFoundException {
18        if(containsKey(key)) {
19            throw new DuplicateKeyFoundException();
20        }
21        this.keyValuePair[this.size++] = new KeyValuePair<K, V>(key, value);
22    }
23
24

```

Markers Properties Servers Data Explorer Debug Terminal JUnit

Finished after 0.041 seconds

Runs: 9/9 Errors: 0 Failures: 0

Failure Trace

- org.self.learn.test.SimpleMapTest [Runner: JUnit 4] (0.005 s)
 - itShouldBeAbleToInitializeAnEmptyMapWithSizeZero (0.000 s)
 - itShouldThrowAnExceptionWhenDuplicateKeyIsEntered (0.000 s)
 - itShouldIndicateIfGivenKeyIsPartOfTheMap (0.000 s)
 - itShouldThrowNoSuchKeyFoundException (0.001 s)
 - itShouldBeAbleToReturnSizeOfTheMap (0.001 s)
 - itShouldIndicateIfGivenValueIsPartOfTheMap (0.000 s)
 - itShouldSaveMultipleKeyAndValuePairs (0.000 s)
 - itShouldReturnSizeGreaterThanZeroUponAddingKeyValuePair (0.002 s)
 - itShouldSaveAKeyAndValuePair (0.001 s)

Along with asserting exception, I also want to ensure that the value inserted with duplicate key is discarded.

Make note of the assertion style we used in this case. Junit rule could not have been used in this case because anything after exception is thrown, won't be executed i.e. the following would not work:

```

map.save(1, "Priyan");
thrown.expect(DuplicateKeyFoundException.class);
map.save(1, "Priyadarshan");
assertEquals("Priyan", map.get(1)); →This assertion will never be executed.
Same is the case with @Test(expected = DuplicateKeyFoundException.class)

```

I have added another way of dealing with exception based assertions to write that test. The test did complete successfully.

Java EE - simple-map/src/test/java/org/self/learn/test/SimpleMapTest.java - Spring Tool Suite

File Edit Source Refactor Navigate Search Project Run Window Help

SimpleMapTest.java

```

98
99     @Test
100    public void itShouldDiscardEntryWithDuplicateKeyInAdditionToThrowingException() {
101        try {
102            map.add(1, "Priyan");
103            map.add(1, "Priyadarshan");
104        } catch (Exception e) {
105            assertEquals(e, instanceof(DuplicateKeyFoundException));
106        }
107
108        assertEquals(map.get(1), is(not(equalTo("Priyadarshan"))));
109    }
110 }

```

SimpleMap.java

```

7
8     public SimpleMap() {
9         this.size = 0;
10        this.keyValuePair = (KeyValuePair<K, V>[][]) new KeyValuePair[5];
11    }
12
13    public int size() {
14        return this.size;
15    }
16
17    public void add(K key, V value) throws DuplicateKeyFoundException {
18        if(containsKey(key)) {
19            throw new DuplicateKeyFoundException();
20        }
21
22        this.keyValuePair[this.size] = new KeyValuePair<K, V>(key, v);
23    }

```

Markers Properties Servers Data Source Explorer Debug Terminal JUnit

Finished after 0.041 seconds

Runs: 10/10 Errors: 0 Failures: 0

Failure Trace

- org.self.learn.test.SimpleMapTest [Runner: JUnit 4] (0.010 s)
 - itShouldBeAbleToInitializeAnEmptyMapWithSizeZero (0.001 s)
 - itShouldThrowAnExceptionWhenDuplicateKeysEntered (0.000 s)
 - itShouldIndicateGivenKeyIsPartOfTheMap (0.000 s)
 - itShouldThrowNoSuchKeyFoundException (0.004 s)
 - itShouldDiscardEntryWithDuplicateKeyInAdditionToThrowingException (0.000 s)
 - itShouldBeAbleToReturnSizeOfTheMap (0.001 s)
 - itShouldIndicateGivenValueIsPartOfTheMap (0.000 s)
 - itShouldSaveMultipleKeyAndValuePairs (0.000 s)
 - itShouldReturnSizeGreaterThanZeroUponAddingKeyValuePair (0.003 s)
 - itShouldSaveAKeyAndValuePair (0.001 s)

Let's test if the map is able to handle null key (and not key~~s~~) and values. I ran a premature test to find that the map does not handle that yet! Make note of the NullPointerException which leads to test failure.

Java EE - simple-map/src/main/java/org/self/learn/SimpleMap.java - Spring Tool Suite

File Edit Source Refactor Navigate Search Project Run Window Help

SimpleMapTest.java

```

104     map.add(1, "Priyadarshan");
105 } catch (Exception e) {
106     assertEquals(e, instanceof(DuplicateKeyFoundException.class));
107 }
108
109 assertEquals(map.get(1), is(not(equalTo("Priyadarshan"))));
110 }
111
112 @Test
113 public void itShouldBeAbleToAcceptANullKeyAndNullValues() {
114     map.add(1, null);
115     map.add(null, "Priyan");
116 }
117

```

SimpleMap.java

```

41
42    public boolean containsKey(K key) {
43        KeyValuePair<K, V> keyValuePair;
44        for(int i=0;i<size;i++) {
45            keyValuePair = this.keyValuePair[i];
46            if(key.equals(keyValuePair.getKey())) {
47                return true;
48            }
49        }
50
51        return false;
52    }
53
54    public boolean containsValue(V value) {
55        KeyValuePair<K, V> keyValuePair;
56
57        for(int i=0;i<size;i++) {
58            keyValuePair = this.keyValuePair[i];
59            if(value.equals(keyValuePair.getValue())) {
60                return true;
61            }
62        }
63
64        return false;
65    }

```

Markers Properties Servers Data Source Explorer Debug Terminal JUnit

Finished after 0.049 seconds

Runs: 11/11 Errors: 1 Failures: 0

Failure Trace

- org.self.learn.test.SimpleMapTest [Runner: JUnit 4] (0.009 s)
 - itShouldBeAbleToInitializeAnEmptyMapWithSizeZero (0.000 s)
 - itShouldBeAbleToAcceptANullKeyAndNullValues (0.000 s)
 - itShouldThrowAnExceptionWhenDuplicateKeysEntered (0.000 s)
 - itShouldIndicateGivenKeyIsPartOfTheMap (0.000 s)
 - itShouldThrowNoSuchKeyFoundException (0.006 s)
 - itShouldDiscardEntryWithDuplicateKeyInAdditionToThrowingException (0.000 s)
 - itShouldBeAbleToReturnSizeOfTheMap (0.001 s)
 - itShouldIndicateGivenValueIsPartOfTheMap (0.000 s)
 - itShouldSaveMultipleKeyAndValuePairs (0.000 s)
 - itShouldReturnSizeGreaterThanZeroUponAddingKeyValuePair (0.002 s)
 - itShouldSaveAKeyAndValuePair (0.000 s)

Refer to the fix highlighted in RED. Test results confirms that the fix took care of the issue.

Java EE - simple-map/src/main/java/org/self/learn/SimpleMap.java - Spring Tool Suite

File Edit Source Refactor Navigate Search Project Run Window Help

SimpleMapTest.java

```

104     map.add(1, "Priyadarshan");
105 } catch (Exception e) {
106     assertEquals(e, instanceof(DuplicateKeyFoundException.class));
107 }
108
109     assertEquals(map.get(1), is(not(equalTo("Priyadarshan"))));
110 }
111
112 @Test
113 public void itShouldBeAbleToAcceptANullKeyAndNullValues()
114 throws DuplicateKeyFoundException {
115     map.add(1, null);
116     map.add(null, null);
117 }
118 }
```

SimpleMap.java

```

40
41     public boolean containsKey(K key) {
42         KeyValuePair<K, V> keyValuePair;
43         for(int i=0;i<size;i++) {
44             keyValuePair = this.keyValuePair[i];
45             if((key==null && keyValuePair.getKey()==null) ||
46                 (key!=null && key.equals(keyValuePair.getKey()))) {
47                 return true;
48             }
49         }
50         return false;
51     }
52
53     public boolean containsValue(V value) {
54 }
```

Markers Properties Servers Data Source Explorer Debug Terminal JUnit

Finished after 0.04 seconds

Runs: 11/11 Errors: 0 Failures: 0

Failure Trace

- org.self.learn.test.SimpleMapTest [Runner: JUnit 4] (0.007 s)
 - itShouldBeAbleToInitializeAnEmptyMapWithSizeZero (0.000 s)
 - itShouldBeAbleToAcceptANullKeyAndNullValues (0.000 s)
 - itShouldThrowAnExceptionWhenDuplicateKeysEntered (0.000 s)
 - itShouldIndicateIfGivenKeysPartOfTheMap (0.000 s)
 - itShouldThrowNoSuchKeyFoundException (0.002 s)
 - itShouldDiscardEntryWithDuplicateKeyInAdditionToThrowingException (0.000 s)
 - itShouldBeAbleToReturnSizeOfTheMap (0.001 s)
 - itShouldIndicateIfGivenValuesPartOfTheMap (0.000 s)
 - itShouldSaveMultipleKeyAndValuePairs (0.000 s)
 - itShouldReturnSizeGreaterThanZeroUponAddingKeyValuePair (0.003 s)
 - itShouldSaveAKeyAndValuePair (0.001 s)

Here is an attempt to make the test more comprehensive and test result indicates that we haven't cared for all scenarios yet.

Java EE - simple-map/src/main/java/org/self/learn/SimpleMap.java - Spring Tool Suite

File Edit Source Refactor Navigate Search Project Run Window Help

SimpleMapTest.java

```

111
112 @Test
113 public void itShouldBeAbleToAcceptANullKeyAndNullValues()
114 throws DuplicateKeyFoundException, NoSuchKeyFoundException {
115     map.add(null, null);
116     map.add(1, null);
117     assertTrue(map.containsKey(null));
118     try {
119         map.add(null, "Priyan");
120     } catch (Exception e) {
121     }
122     assertEquals(e, instanceof(DuplicateKeyFoundException.class));
123     assertNull(map.get(null));
124 }
125 }
```

SimpleMap.java

```

25
26     public V get(K key) throws NoSuchKeyFoundException {
27         KeyValuePair<K, V> keyValuePair;
28
29         if(!containsKey(key)) {
30             throw new NoSuchKeyFoundException();
31         }
32
33         for(int i=0;i<size;i++) {
34             keyValuePair = this.keyValuePair[i];
35             if(key.equals(keyValuePair.getKey())) {
36                 return keyValuePair.getValue();
37             }
38         }
39
40         return null;
41 }
```

Markers Properties Servers Data Source Explorer Debug Terminal JUnit

Finished after 0.043 seconds

Runs: 11/11 Errors: 1 Failures: 0

Failure Trace

- org.self.learn.test.SimpleMapTest [Runner: JUnit 4] (0.010 s)
 - itShouldBeAbleToInitializeAnEmptyMapWithSizeZero (0.000 s)
 - itShouldBeAbleToAcceptANullKeyAndNullValues (0.000 s)
 - itShouldThrowAnExceptionWhenDuplicateKeysEntered (0.000 s)
 - itShouldIndicateIfGivenKeysPartOfTheMap (0.004 s)
 - itShouldThrowNoSuchKeyFoundException (0.003 s)
 - itShouldDiscardEntryWithDuplicateKeyInAdditionToThrowingException (0.001 s)
 - itShouldBeAbleToReturnSizeOfTheMap (0.000 s)
 - itShouldIndicateIfGivenValuesPartOfTheMap (0.000 s)
 - itShouldSaveMultipleKeyAndValuePairs (0.000 s)
 - itShouldReturnSizeGreaterThanZeroUponAddingKeyValuePair (0.001 s)
 - itShouldSaveAKeyAndValuePair (0.001 s)

Refer to the highlighted code fix, tests look good with that.

```

Java EE - simple-map/src/main/java/org/self/learn/SimpleMap.java - Spring Tool Suite
File Edit Source Refactor Navigate Search Project Run Window Help
SimpleMapTest.java
108     assertEquals(map.get(1), is(not(equalTo("Priyadarshan"))));
109 }
110
111 @Test
112 public void itShouldBeAbleToAcceptANullKeyAndNullValues()
113     throws DuplicateKeyFoundException, NoSuchKeyNotFoundException {
114     map.add(null, null);
115     map.add(1, null);
116     assertTrue(map.containsKey(null));
117     try {
118         map.add(null, "Priyan");
119     } catch (Exception e) {
120         assertEquals(e, isInstanceOf(DuplicateKeyFoundException.class));
121     }
122     assertNull(map.get(null));
123 }
124
125 }

SimpleMap.java
25 public V get(K key) throws NoSuchKeyNotFoundException {
26     KeyValuePair<K, V> keyValuePair;
27
28     if(!containsKey(key)) {
29         throw new NoSuchKeyNotFoundException();
30     }
31
32     for(int i=0;i<size();i++) {
33         keyValuePair = this.keyValuePair[i];
34         if((key==null && keyValuePair.getKey()==null) ||
35             (key!=null && key.equals(keyValuePair.getKey()))) {
36             return keyValuePair.getValue();
37         }
38     }
39
40     return null;
41 }
42
43 }

Markers Properties Servers Data Source Explorer Debug Terminal JUnit
Finished after 0.04 seconds
Runs: 11/11 Errors: 0 Failures: 0
Failure Trace
org.self.learn.test.SimpleMapTest [Runner:JUnit4] (0.006 s)
itShouldBeAbleToInitializeAnEmptyMapWithSizeZero (0.000 s)
itShouldBeAbleToAcceptANullKeyAndNullValues (0.000 s)
itShouldThrowAnExceptionWhenDuplicateKeysEntered (0.000 s)
itShouldIndicateIfGivenKeysPartOfTheMap (0.000 s)
itShouldThrowNoSuchKeyNotFoundException (0.002 s)
itShouldDiscardEntryWithDuplicateKeyInAdditionToThrowingException (0.001 s)
itShouldBeAbleToReturnSizeOfTheMap (0.000 s)
itShouldIndicateIfGivenValuesPartOfTheMap (0.000 s)
itShouldSaveMultipleKeyAndValuePairs (0.000 s)

```

Following test-build iteration is to ensure that the same handled for containsValue() method as well.

```

Java EE - simple-map/src/test/java/org/self/learn/test/SimpleMapTest.java - Spring Tool Suite
File Edit Source Refactor Navigate Search Project Run Window Help
SimpleMapTest.java
111
112 @Test
113 public void itShouldBeAbleToAcceptANullKeyAndNullValues()
114     throws DuplicateKeyFoundException, NoSuchKeyNotFoundException {
115     map.add(null, null);
116     map.add(1, null);
117     assertTrue(map.containsKey(null));
118     try {
119         map.add(null, "Priyan");
120     } catch (Exception e) {
121         assertEquals(e, isInstanceOf(DuplicateKeyFoundException.class));
122     }
123     assertNull(map.get(null));
124     assertEquals(map.containsKey(null), is(equalTo(true)));
125     assertEquals(map.containsValue(null), is(equalTo(true)));
126 }
127
128 }

SimpleMap.java
50
51
52
53
54
55
56 public boolean containsValue(V value) {
57     KeyValuePair<K, V> keyValuePair;
58     for(int i=0;i<size();i++) {
59         keyValuePair = this.keyValuePair[i];
60         if(value.equals(keyValuePair.getValue())) {
61             return true;
62         }
63     }
64
65     return false;
66 }
67

Markers Properties Servers Data Source Explorer Debug Terminal JUnit
Finished after 0.04 seconds
Runs: 11/11 Errors: 1 Failures: 0
Failure Trace
java.lang.NullPointerException
at org.self.learn.SimpleMap.containsValue(SimpleMap.java:56)
at org.self.learn.test.SimpleMapTest.itShouldBeAbleToAcceptANullKeyAndNullValues(SimpleMapTest.java:125)

```

That will take care of it!

Java EE - simple-map/src/main/java/org/self/learn/SimpleMap.java - Spring Tool Suite

File Edit Source Refactor Navigate Search Project Run Window Help

SimpleMapTest.java

```

111     @Test
112     public void itShouldBeAbleToAcceptANullKeyAndNullValues()
113         throws DuplicateKeyFoundException, NoSuchKeyFoundException {
114         map.add(null, null);
115         map.add(1, null);
116         assertTrue(map.containsKey(null));
117         try {
118             map.add(null, "Priyan");
119         } catch (Exception e) {
120             assertEquals(e, instanceof(DuplicateKeyFoundException.class));
121         }
122         assertNull(map.get(null));
123         assertEquals(map.containsKey(null), isEqualTo(true));
124         assertEquals(map.containsValue(null), isEqualTo(true));
125     }
126 }
127
128 }
```

SimpleMap.java

```

53         return false;
54     }
55
56     public boolean containsValue(V value) {
57         KeyValuePair<K, V> keyValPair;
58         for(int i=0;i<size;i++) {
59             keyValPair = this.keyValuePair[i];
60             if((value==null && keyValPair.getValue()==null)|||
61                 (value!=null && value.equals(keyValPair.getValue())))
62                 return true;
63         }
64     }
65
66     return false;
67 }
68
69 }
```

Marker Properties Servers Data Source Explorer Debug Terminal JUnit

Finished after 0.04 seconds

Runs: 11/11 Errors: 0 Failures: 0

Failure Trace

- org.self.learn.test.SimpleMapTest [Runner: JUnit 4] [0.006 s]
 - itShouldBeAbleToInitializeAnEmptyMapWithSizeZero (0.000 s)
 - itShouldBeAbleToAcceptANullKeyAndNullValues (0.000 s)
 - itShouldThrowAnExceptionWhenDuplicateKeysEntered (0.000 s)
 - itShouldIndicateIfGivenKeyIsPartOfTheMap (0.000 s)
 - itShouldThrowNoSuchKeyFoundException (0.002 s)
 - itShouldDiscardEntryWithDuplicateKeyInAdditionToThrowingException (0.001 s)
 - itShouldBeAbleToReturnSizeOfTheMap (0.000 s)
 - itShouldIndicateIfGivenValueIsPartOfTheMap (0.000 s)
 - itShouldSaveMultipleKeyAndValuePairs (0.000 s)

Next up is remove... Following few screenshots depict how we put its implementation together using TDD approach.

Java EE - simple-map/src/test/java/org/self/learn/test/SimpleMapTest.java - Spring Tool Suite

File Edit Source Refactor Navigate Search Project Run Window Help

SimpleMapTest.java

```

103     map.add(1, "Priyan");
104     map.add(1, "Priyadarshan");
105 } catch (Exception e) {
106     assertEquals(e, instanceof(DuplicateKeyFoundException.class));
107 }
108
109     assertEquals(map.get(1), is(not(equalTo("Priyadarshan"))));
110 }
111
112 @Test
113     public void itShouldBeAbleToAcceptANullKeyAndNullValues()
114         throws DuplicateKeyFoundException, NoSuchKeyFoundException {
115         map.add(null, null);
116         map.add(1, null);
117         assertTrue(map.containsKey(null));
118         try {
119             map.add(null, "Priyan");
120         } catch (Exception e) {
121             assertEquals(e, instanceof(DuplicateKeyFoundException.class));
122         }
123         assertNull(map.get(null));
124         assertEquals(map.containsKey(null), isEqualTo(true));
125         assertEquals(map.containsValue(null), isEqualTo(true));
126     }
127
128 @Test
129     public void itShouldAllowToRemoveKeyValuePairGivenCorrespondingKey() throws Exception {
130         map.add(1, "Priyan");
131         map.add(2, "Priyadarshan");
132         map.remove(2);
133         assertEquals(map.containsKey(2), isEqualTo(false));
134         assertEquals(map.containsValue("Priyadarshan"), isEqualTo(false));
135         assertEquals(map.size(), isEqualTo(1));
136     }
137
138 }
```

The method remove(int) is undefined

SimpleMap.java

```

35         (key!=null && key.equals(keyValuePair.getKey())) {
36             return keyValuePair.getValue();
37         }
38
39         return null;
40     }
41
42     public boolean containsKey(K key) {
43         KeyValuePair<K, V> keyValPair;
44         for(int i=0;i<size;i++) {
45             keyValPair = this.keyValuePair[i];
46             if((key==null && keyValPair.getKey()==null)|||
47                 (key!=null && key.equals(keyValPair.getKey())))
48                 return true;
49         }
50     }
51
52     return false;
53 }
54
55     public boolean containsValue(V value) {
56         KeyValuePair<K, V> keyValPair;
57         for(int i=0;i<size;i++) {
58             keyValPair = this.keyValuePair[i];
59             if((value==null && keyValPair.getValue()==null)|||
60                 (value!=null && value.equals(keyValPair.getValue())))
61                 return true;
62         }
63     }
64
65     return false;
66 }
67
68 }
```

Writable Smart Insert 132:19

2:52 AM 2/24/2016

```

Java EE - simple-map/src/main/java/org/self/learn/SimpleMap.java - Spring Tool Suite
File Edit Source Refactor Navigate Project Run Window Help
Quick Access Java EE Debug Git
SimpleMapTest.java
121     assertThat(e, isInstanceOf(DuplicateKeyFoundException.class));
122 }
123 assertNull(map.get(null));
124 assertThat(map.containsKey(null), is(equalTo(true)));
125 assertThat(map.containsValue(null), is(equalTo(true)));
126 }

127 @Test
128 public void itShouldAllowToRemoveKeyValuePairGivenCorrespondingKey() throws Exception {
129     map.add(1, "Priyan");
130     map.add(2, "Priyadarshan");
131     map.remove(2);
132     assertThat(map.containsKey(2), is(equalTo(false)));
133     assertThat(map.containsValue("Priyadarshan"), is(equalTo(false)));
134     assertThat(map.size(), is(equalTo(1)));
135 }
136 }
137
138 }

SimpleMap.java
57     KeyValuePair<K, V> keyValuePair;
58     for(int i=0;i<size;i++) {
59         keyValuePair = this.keyValuePair[i];
60         if((value==null && keyValuePair.getValue()==null) ||
61             (value!=null && value.equals(keyValuePair.getValue())))
62             return true;
63     }
64
65     return false;
66 }
67
68 public void remove(K i) {
69     // TODO Auto-generated method stub
70 }
71 }
72 }
73
74

```

Markers Properties Servers Data Explorer Debug Terminal JUnit

Finished after 0.081 seconds

Runs: 12/12 Errors: 0 Failures: 1

Failure Trace

- java.lang.AssertionError:
Expected: is <false>
but was <true>
- at org.hamcrest.MatcherAssert.assertThat(MatcherAssert.java:20)
- at org.self.learn.test.SimpleMapTest.itShouldAllowToRemoveKeyValuePairGivenCorrespondingKey(SimpleMapTest.java:133)

Let's implement the remove method

```

Java EE - simple-map/src/main/java/org/self/learn/SimpleMap.java - Spring Tool Suite
File Edit Source Refactor Navigate Project Run Window Help
Quick Access Java EE Debug Git
SimpleMapTest.java
120     } catch (Exception e) {
121         assertThat(e, isInstanceOf(DuplicateKeyFoundException.class));
122     }
123     assertNull(map.get(null));
124     assertThat(map.containsKey(null), is(equalTo(true)));
125     assertThat(map.containsValue(null), is(equalTo(true)));
126 }

127 @Test
128 public void itShouldAllowToRemoveKeyValuePairGivenCorrespondingKey() throws Exception {
129     map.add(1, "Priyan");
130     map.add(2, "Priyadarshan");
131     map.remove(2);
132     assertThat(map.containsKey(2), is(equalTo(false)));
133     assertThat(map.containsValue("Priyadarshan"), is(equalTo(false)));
134     assertThat(map.size(), is(equalTo(1)));
135 }
136 }
137
138 }

SimpleMap.java
69 public void remove(K key) {
70     KeyValuePair<K, V> keyValuePair;
71
72     for(int i=0;i<size;i++) {
73         keyValuePair = this.keyValuePair[i];
74         if((key==null && keyValuePair.getKey()==null) ||
75             (key!=null && key.equals(keyValuePair.getKey()))) {
76             if(this.size==1){
77                 this.keyValuePair[i] = null;
78             } else {
79                 this.keyValuePair[i] = this.keyValuePair[this.size-1];
80                 this.keyValuePair[this.size-1] = null;
81             }
82         }
83     }
84
85     this.size--;
86 }
87

```

Markers Properties Servers Data Explorer Debug Terminal JUnit

Finished after 0.041 seconds

Runs: 12/12 Errors: 0 Failures: 0

Failure Trace

Added couple of more tests for surfacing exception when key is not found. Tests are failing at this point.

The screenshot shows the Spring Tool Suite interface with two code editors. The left editor displays `SimpleMapTest.java` and the right editor displays `SimpleMap.java`. The `SimpleMapTest.java` file contains several test cases for the `remove` method. The `Failure Trace` panel at the bottom shows a single failure:

```
java.lang.AssertionError: Expected exception: org.self.learn.NoSuchKeyException
```

The status bar at the bottom indicates "Runs: 14/14 Errors: 0 Failures: 2".

Here we go with the fix... and that should be it for remove.

The screenshot shows the Spring Tool Suite interface with the same two code editors. The `SimpleMapTest.java` file now includes a check for the key before removing it. The `Failure Trace` panel is empty, indicating no failures. The status bar at the bottom indicates "Runs: 14/14 Errors: 0 Failures: 0".

The final one is support for dynamic size. I have added six key/value pairs to the map to end up with `ArrayIndexOutOfBoundsException` which lead to test failure.

```

Java EE - simple-map/src/main/java/org/self/learn/SimpleMap.java - Spring Tool Suite
File Edit Source Refactor Navigate Project Run Window Help
Quick Access Java EE Debug Git
SimpleMapTest.java
149     map.remove(100);
150 }
151 @Test
152 public void itShouldGrowInSizeAtRuntimeBasedOnNeed() throws DuplicateKeyFoundException, NoSuchKeyNotFoundException {
153     map.add(1, "Priyan-1");
154     map.add(2, "Priyan-2");
155     map.add(3, "Priyan-3");
156     map.add(4, "Priyan-4");
157     map.add(5, "Priyan-5");
158     map.add(6, "Priyan-6");
159     assertEquals(6, map.size());
160 }
161 assertEquals("Priyan-6", map.get(6));
162 assertEquals("Priyan-5", map.get(5));
163 assertEquals("Priyan-4", map.get(4));
164 assertEquals("Priyan-3", map.get(3));
165 assertEquals("Priyan-2", map.get(2));
166 assertEquals("Priyan-1", map.get(1));
167 assertEquals(6, map.size());
168 }

SimpleMap.java
13④ public int size() {
14     return this.size;
15 }
16
17④ public void add(K key, V value) throws DuplicateKeyFoundException {
18     if(containsKey(key)) {
19         throw new DuplicateKeyFoundException();
20     }
21
22     this.keyValuePair[this.size++] = new KeyValuePair<K, V>(key, value);
23 }
24
25④ public V get(K key) throws NoSuchKeyNotFoundException {
26     KeyValuePair<K, V> keyValuePair;
27
28     if(!containsKey(key)) {
29         throw new NoSuchKeyNotFoundException();
30     }
31
32     for(int i=0;i<size;i++) {
33

```

Runs: 15/15 Errors: 0 Failures: 0

Failure Trace

- java.lang.ArrayIndexOutOfBoundsException: 5
 - at org.self.learn.SimpleMap.add(SimpleMap.java:22)
 - at org.self.learn.test.SimpleMapTest.itShouldGrowInSizeAtRuntimeBasedOnNeed(SimpleMapTest.java:160)

Following is the completed test and corresponding implementation. Note that all the tests complete successfully at this point. And we have built all desired functionality into our map as well. That concludes this exercise.

```

Java EE - simple-map/src/main/java/org/self/learn/SimpleMap.java - Spring Tool Suite
File Edit Source Refactor Navigate Project Run Window Help
Quick Access Java EE Debug Git
SimpleMapTest.java
149     map.remove(100);
150 }
151 @Test
152 public void itShouldGrowInSizeAtRuntimeBasedOnNeed() throws DuplicateKeyFoundException, NoSuchKeyNotFoundException {
153     map.add(1, "Priyan-1");
154     map.add(2, "Priyan-2");
155     map.add(3, "Priyan-3");
156     map.add(4, "Priyan-4");
157     map.add(5, "Priyan-5");
158     map.add(6, "Priyan-6");
159     assertEquals(6, map.size());
160 }
161 assertEquals("Priyan-6", map.get(6));
162 assertEquals("Priyan-5", map.get(5));
163 assertEquals("Priyan-4", map.get(4));
164 assertEquals("Priyan-3", map.get(3));
165 assertEquals("Priyan-2", map.get(2));
166 assertEquals("Priyan-1", map.get(1));
167 assertEquals(6, map.size());
168 }

SimpleMap.java
13④ public int size() {
14     return this.size;
15 }
16
17④ public void add(K key, V value) throws DuplicateKeyFoundException {
18     if(containsKey(key)) {
19         throw new DuplicateKeyFoundException();
20     }
21
22     if(this.size>this.keyValuePair.length) {
23         this.keyValuePair[this.size++] = new KeyValuePair<K, V>(key, value);
24     } else {
25         @SuppressWarnings("unchecked")
26         KeyValuePair<K, V>[] newKeyValuePair = (KeyValuePair<K, V>[]) new KeyValuePair[0];
27         System.arraycopy(this.keyValuePair, 0, newKeyValuePair, 0, this.size);
28         this.keyValuePair = newKeyValuePair;
29         this.keyValuePair[this.size++] = new KeyValuePair<K, V>(key, value);
30     }
31 }
32
33

```

Runs: 15/15 Errors: 0 Failures: 0

Failure Trace

- java.lang.ArrayIndexOutOfBoundsException: 5
 - at org.self.learn.SimpleMap.add(SimpleMap.java:22)
 - at org.self.learn.test.SimpleMapTest.itShouldGrowInSizeAtRuntimeBasedOnNeed(SimpleMapTest.java:160)