

Telefonbuch

Frank Hedecke

Voraussetzungen: Testing, Exceptions, Collections

Zeithorizont: 45 Minuten

Lernziele:

1 Beschreibung

Schreibe ein kleines Telefonbuch, welches die gegebenen Testfälle erfüllt.

Das Telefonbuch enthält verschiedene Einträge, mit je einem Namen und einer Telefonnummer. Unter gleichen Namen können keine zwei Einträge vorhanden sein.

2 Aufgaben

1. Implementiere die Klassen *PhoneBook* und *Entry*.
2. Entnehme die benötigten Methoden den Testklassen.
3. Teste beide Klassen, bis alle Testfälle erfüllt werden.

3 Hinweise

- Nutze eine *LinkedList* als Grundlage für das Telefonbuch. (optional)
- Überschreibe die Methode *equals* in der Klasse *Entry*. (optional)
- Erstelle eine neue Exceptionklasse.
- <https://docs.oracle.com/javase/7/docs/api/>
- <http://junit.org/javadoc/latest/index.html>

4 Testklassen

```
1 import static org.junit.Assert.*;
2 import org.junit.*;
3
4 public class EntryTest {
5     private Entry e1;
6     private Entry e2;
7
8     @Before
9     public void init() {
10         this.e1 = new Entry("Dumbledore", "0123-3136487");
11         this.e2 = new Entry("Dumbledore", "0123-0123456");
12     }
13
14     @Test
15     public void testGetters() {
16         assertEquals(this.e1.getName(), "Dumbledore");
17         assertEquals(this.e1.getNumber(), "0123-3136487");
18     }
19
20     @Test
21     public void testEquality() {
22         assertEquals(e1, e2);
23     }
24 }

```



```
1 import static org.junit.Assert.*;
2 import org.junit.*;
3
4 public class PhoneBookTest {
5
6     private PhoneBook book;
7     private Entry e1;
8     private Entry e2;
9
10    @Before
11    public void init() {
12        this.e1 = new Entry("Dumbledore", "0123-3136487");
13        this.e2 = new Entry("Snape", "0123-1414327");
14
15        this.book = new PhoneBook();
16    }
17
18    @Test
19    public void testAddandGet() throws EntryAlreadyExistsException {
20        this.book.addEntry(this.e1);
21        this.book.addEntry(this.e2);
22        assertEquals(this.book.getEntry(0), this.e1);
23        assertEquals(this.book.getEntry(1), this.e2);
24    }
25 }
```

```
26     @Test(expected = NullPointerException.class)
27     public void testAddNull() throws EntryAlreadyExistsException {
28         this.book.addEntry(null);
29     }
30
31     @Test(expected = EntryAlreadyExistsException.class)
32     public void testAddTwice() throws EntryAlreadyExistsException {
33         this.book.addEntry(this.e1);
34         this.book.addEntry(this.e1);
35     }
36 }
```