

```

1 package Tests;
2
3 import static org.junit.Assert.*;
13
14 public class IntegrationTest {
15
16     private IHotelFassade hotelFassade;
17     private int zNr;
18     private ZusatzleistungTyp wlan;
19
20     @Before
21     public void setUp() {
22         try {
23             this.hotelFassade = new HotelFassade();
24             hotelFassade.erzeugeGast(1, "Steffen", email("st", "hw", "de"));
25             hotelFassade.erzeugeGast(2, "Swaneet", email("sw", "hw", "de"));
26             hotelFassade.erzeugeGast(3, "Flasche", email("ab", "cd", "de"));
27             zNr = 0;
28             wlan = hotelFassade.erzeugeZusatzleistung("WLAN");
29         } catch (Exception e) {
30             e.printStackTrace();
31         }
32     }
33
34     @Test
35     public void testIntegration() {
36         try {

```

```

37     GastTyp steffen = hotelFassade.sucheGastNachName("Steffen");
38     GastTyp swaneet = hotelFassade.sucheGastNachName("Swaneet");
39     GastTyp flasche = hotelFassade.sucheGastNachName("Flasche");
40
41     assertNull(hotelFassade.sucheGastNachName("cant Touch Me"));
42     assertNotNull(steffen);
43     assertNotNull(swaneet);
44     assertNotNull(flasche);
45     assertFalse(steffen.istStammkunde());
46     assertFalse(swaneet.istStammkunde());
47     assertFalse(flasche.istStammkunde());
48
49     // steffen macht nicht gen.gend reservierungen mit zusatzleistungen
50     for (int i = 1; i <= 2; i++) {
51         ReservierungTyp res = hotelFassade.reserviereZimmer(
52             steffen.nr(), zimmerNr());
53         hotelFassade.bucheZusatzleistung(res.nr(), wlan.nr());
54     }
55     // swaneet macht 3 reservierungen mit zusatzleistungen
56     for (int i = 1; i <= 3; i++) {
57         ReservierungTyp res = hotelFassade.reserviereZimmer(
58             swaneet.nr(), zimmerNr());
59         hotelFassade.bucheZusatzleistung(res.nr(), wlan.nr());
60     }
61     // flasche macht eine reservierung- zu wenig um stammkunde zu sein
62     for (int i = 1; i <= 5; i++) {
63         hotelFassade.reserviereZimmer(flasche.nr(), zimmerNr());

```

```

64     }
65
66     steffen = hotelFassade.sucheGastNachName("Steffen");
67     swaneet = hotelFassade.sucheGastNachName("Swaneet");
68     flasche = hotelFassade.sucheGastNachName("Flasche");
69
70     assertFalse(steffen.istStammkunde());
71     assertTrue(swaneet.istStammkunde());
72     assertFalse(flasche.istStammkunde());
73
74     // jetzt hat flasch gen.gend reservierungen
75     hotelFassade.reserviereZimmer(flasche.nr(), zimmerNr());
76     flasche = hotelFassade.sucheGastNachName("Flasche");
77     assertTrue(flasche.istStammkunde());
78     } catch (Exception e) {
79         e.printStackTrace();
80     }
81 }
82
83 // gibt immer eine neue zimmerNr zur.ck
84 private int zimmerNr() {
85     zNr += 1;
86     return zNr;
87 }
88 }
89

```