

```

1 package Reservierungskomponente;
2
3 import static Utilities.Precondition.assertArgument;
4
5
6 public class ReservierungsAnwendungsfall implements IReservierungServices {
7
8     private Reservierungsverwalter resVerw = null;
9     private IGastServicesFuerReservierung gServFuerRes;
10
11     public ReservierungsAnwendungsfall(Reservierungsverwalter resVerw,
12         IGastServicesFuerReservierung gServFuerRes) {
13         this.resVerw = resVerw;
14         this.gServFuerRes = gServFuerRes;
15     }
16
17     @Override
18     public ZusatzleistungTyp erzeugeZusatzleistung(String name)
19         throws TechnicalException {
20         assertArgument(name != null && name.length() > 0);
21         return this.resVerw.erzeugeZusatzleistung(name);
22     }
23
24     @Override
25     public ReservierungTyp reserviereZimmer(Integer gastNr, Integer zimmerNr)
26         throws TechnicalException {
27         assertArgument(gastNr != null && gastNr > 0);
28         assertArgument(zimmerNr != null);
29     }

```

```

30     ReservierungTyp result = resVerw.reserviereZimmer(gastNr, zimmerNr);
31     gServFuerRes.markiereGastAlsStammkunden(gastNr);
32     return result;
33 }
34
35 @Override
36 public void bucheZusatzleistung(Integer reservierungNr,
37     Integer zusatzleistungNr) throws TechnicalException {
38     assertArgument(zusatzleistungNr != null && zusatzleistungNr > 0);
39     assertArgument(reservierungNr != null && reservierungNr > 0);
40     Integer gastNr = resVerw.sucheGastNrNachReservierungNr(reservierungNr);
41     resVerw.bucheZusatzleistung(reservierungNr, zusatzleistungNr);
42     gServFuerRes.markiereGastAlsStammkunden(gastNr);
43 }
44
45 @Override
46 public Integer sucheGastNrNachReservierungNr(Integer reservierungNr)
47     throws TechnicalException {
48     return this.resVerw.sucheGastNrNachReservierungNr(reservierungNr);
49 }
50
51 }
52

```