Die Klassen des Projektes

Die Klasse `Karte`

```
1  public class Karte {
2
3    int pin;
4    int kartenNr;
5    int kontoNr;
6    int blz;
7
8    Karte(int krtnr, int ktonr, int b, int p) {
9      pin = p;
10     kartenNr = krtnr;
11     kontoNr = ktonr;
12     blz = b;
13   }
14
15   int gibPin() {
16     return pin;
17   }
18
19   int gibkartenNr() {
20     return kartenNr;
21   }
22
23   int gibKontoNr() {
24     return kontoNr;
25   }
26
27   int gibBLZ() {
28     return blz;
29   }
30
31 }
```

Die Klasse `Konto`

```
1  public class Konto {
2    int kontoNr;
3    int blz;
4    double kontoStand;
5
6    Konto(int ktonr, int b, double d) {
7      kontoNr = ktonr;
8      blz = b;
9      kontoStand = d;
10   }
11
12   int gibKontoNr() {
13     return kontoNr;
14   }
15
16   int gibBLZ() {
17     return blz;
18   }
19
20   double gibKontoStand() {
21     return kontoStand;
22   }
23
24   boolean hebeAb(double betrag) {
25     if (kontoStand > betrag) {
26       kontoStand = kontoStand - betrag;
27       return true;
28     } else {
29       return false;
30     }
31   }
32 }
```

Die Klasse `KontoVerwaltung`

```
1  public class KontoVerwaltung {
2
3    Konto konto1;
4    Konto konto2;
5    Konto konto3;
6
7    KontoVerwaltung() {
8      konto1 = new Konto(747, 50059050 , 200.0);
9      konto2 = new Konto(256, 50059050, 15000.0);
10     konto3= new Konto(815, 30039030, -275.0);
11   }
12
13   Konto findeKonto(int ktoNr) {
14     if (konto1.gibKontoNr() == ktoNr) {
15       return konto1;
16     } else if (konto2.gibKontoNr() == ktoNr) {
17       return konto2;
18     } else if (konto3.gibKontoNr() == ktoNr) {
19       return konto3;
20     } else {
21       return null;
22     }
23   }
24 }
```

Die Klasse `BankAutomat`

```
1  public class BankAutomat {
2
3    KontoVerwaltung ktoVerwaltung;
4    Karte aktuelleKarte;
5    Konto aktuellesKonto;
6    double bargeldBestand;
7
8
9    BankAutomat(double bar) {
10     ktoVerwaltung = new KontoVerwaltung();
11     bargeldBestand = bar;
12   }
13
14   boolean pruefePin(Karte krt, int pin) {
15     boolean ergebnis;
16     if (krt.gibPin() == pin) {
17       aktuelleKarte = krt;
18       aktuellesKonto = ktoVerwaltung.findeKonto(aktuelleKarte.gibKontoNr());
19       if (aktuellesKonto != null) {
20         ergebnis =  true;
21       } else {
22         ergebnis = false;
23       }
24     } else {
25       beendeTransaktion();
26       ergebnis = false;
27     }
28     return ergebnis;
29   }
30
31
32   boolean hebeAb(double betrag) {
33     boolean ergebnis = false;
34     if (aktuellesKonto != null && betrag < bargeldBestand) {
35       ergebnis = aktuellesKonto.hebeAb(betrag);
36     }
37     return ergebnis;
38   }
39
40   double gibKontoStand() {
41     if (aktuellesKonto != null) {
42       return aktuellesKonto.gibKontoStand();
43     } else {
44       return -1.0;
45     }
46   }
47
```

```
48    void beendeTransaktion() {
49      aktuellesKonto = null;
50      aktuelleKarte = null;
51    }
52  }
```

Die Testklassen des Projekts

Die Klasse `KarteTest`

```
1   package bankautomat;
2
3   import junit.framework.TestCase;
4
5   public class KarteTest extends TestCase {
6
7     public void testKonstruktor() {
8       Karte k = new Karte(7711, 555, 30044050, 2345);
9       assertEquals(7711, k.gibkartenNr());
10      assertEquals(555, k.gibKontoNr());
11      assertEquals(30044050, k.gibBLZ());
12      assertEquals(2345, k.gibPin());
13    }
14  }
```

Die Klasse `KontoTest`

```
1   package bankautomat;
2
3   import junit.framework.TestCase;
4
5   public class KontoTest extends TestCase {
6
7     public void testGibKontoNr() {
8       Konto k1 = new Konto(4711, 50055050, 5000.0);
9       assert(k1.gibKontoStand() == 5000.0);
10      assert(k1.gibBLZ() == 50055050);
11      assert(k1.gibKontoNr() == 4711);
12    }
13
14  }
```

Die Klasse `KontoVerwaltungTest`

```
1   package bankautomat;
2
3   import junit.framework.TestCase;
4
5   public class KontoVerwaltungTest extends TestCase {
6
7     public void testFindeKonto() {
8       KontoVerwaltung kv = new KontoVerwaltung();
9       Konto k = kv.findeKonto(747);
10      assertNotNull(k);
11      assertEquals(50059050, k.gibBLZ());
12      assertEquals(200.0, k.gibKontoStand());
13      k = kv.findeKonto(256);
14      assertNotNull(k);
15      assertEquals(50059050, k.gibBLZ());
16      assertEquals(15000.0, k.gibKontoStand());
17      k = kv.findeKonto(815);
18      assertNotNull(k);
19      assertEquals(30039030, k.gibBLZ());
20      assertEquals(-275.0, k.gibKontoStand());
21      k = kv.findeKonto(676);
22      assertNull(k);
23    }
24  }
```

Die Klasse `BankAutomatTest`

```
1  package bankautomat;
2
3
4  import junit.framework.TestCase;
5
6  public class BankAutomatTest extends TestCase {
7
8    public void testPruefePin() {
9      BankAutomat ba = new BankAutomat(10000.0);
10     Karte k = new Karte(7711, 747, 50059050, 1111);
11     assertEquals(true, ba.pruefePin(k, 1111));
12     assertNotNull(ba.aktuelleKarte);
13     assertEquals(false, ba.pruefePin(k, 1212));
14     assertNull(ba.aktuelleKarte);
15   }
16
17   public void testBeendeTransaktion() {
18     BankAutomat ba = new BankAutomat(10000.0);
19     Karte k = new Karte(7711, 747, 50059050, 1111);
20     ba.pruefePin(k, 1111);
21     ba.beendeTransaktion();
22     assertNull(ba.aktuelleKarte);
23     assertNull(ba.aktuellesKonto);
24   }
25
26   public void testHebeAb() {
27     BankAutomat baVielGeld = new BankAutomat(10000.0);
28     Karte k = new Karte(7711, 747, 50059050, 1111);
29
30     // Bei diesem Automat ist genug Geld vorhanden
31     baVielGeld.pruefePin(k, 1111);
32     assertEquals(true, baVielGeld.hebeAb(50.0));
33     assertEquals(150.0, baVielGeld.gibKontoStand());
34     assertEquals(false, baVielGeld.hebeAb(200.0));
35     assertEquals(150.0, baVielGeld.gibKontoStand());
36
37     // Dieser Automat hat zu wenig Geld fuer die Auszahlung
38     BankAutomat baWenigGeld = new BankAutomat(20.0);
39     baWenigGeld.pruefePin(k, 1111);
40     assertEquals(false, baWenigGeld.hebeAb(50.0));
41     assertEquals(200.0, baWenigGeld.gibKontoStand());
42   }
43
44   public void testGibKontoStand() {
45     BankAutomat ba = new BankAutomat(10000.0);
46     Karte k = new Karte(7711, 747, 50059050, 1111);
47     ba.pruefePin(k, 1111);
48     assertEquals(200.0, ba.gibKontoStand());
49     ba.beendeTransaktion();
50     assertEquals(-1.0, ba.gibKontoStand());
51   }
52
53 }
```