

Polymorphismus Vögel

Lösung

```

Application.java  Vogel.java  Huhn.java
1  package ch.bbw.pr.ueberflieger;
2
3  import ch.bbw.pr.ueberflieger.model.Ente;
4
5
6
7  /**
8   * Application
9   * @author Peter Rutschmann
10  * @version 28.03.2019
11  */
12  public class Application {
13
14      public static void main(String[] args) {
15          System.out.println("Die Überflieger");
16          System.out.println();
17
18          Vogel birdy = new Vogel("Birdy", "Schwarz");
19          Huhn helena = new Huhn("Helena", "Weiss");
20          Ente quak = new Ente("Quak", "Gelb");
21          birdy.fliegen();
22          helena.fliegen();
23          quak.schwimmen();
24          Vogel einVogel = birdy;
25          einVogel.fliegen();
26          Vogel nochEinVogel = helena;
27          nochEinVogel.fliegen();
28
29          //nochEinVogel.schwimmen();
30          //man muss eine Ente sein, damit man schwimmen kann
31
32          //Ente ede = new Vogel("Ede", "Braun");
33          //Eine Ente muss als Ente instanziiert werden
34
35          Ente emma = quak;
36          emma.schwimmen();
37
38          //Ente elsa = (Ente) birdy;
39          //birdy ist keine Ente
40          //elsa.schwimmen();
41
42          Vogel beate = new Ente("Beate", "Gefleckt");
43          beate.fliegen();
44
45          //beate.schwimmen();
46          //beate ist nur Typ Vogel, kann deshalb nicht schwimmen
47
48          //((Ente)birdy).schwimmen();
49          //birdy ist nur vom Typ Vogel, kann deshalb nicht schwimmen
50
51          ((Ente)beate).schwimmen();
52
53
54          //Die Versuche mit super
55          //==> Siehe Klasse Huhn!!
56
57      }
58  }

```



```

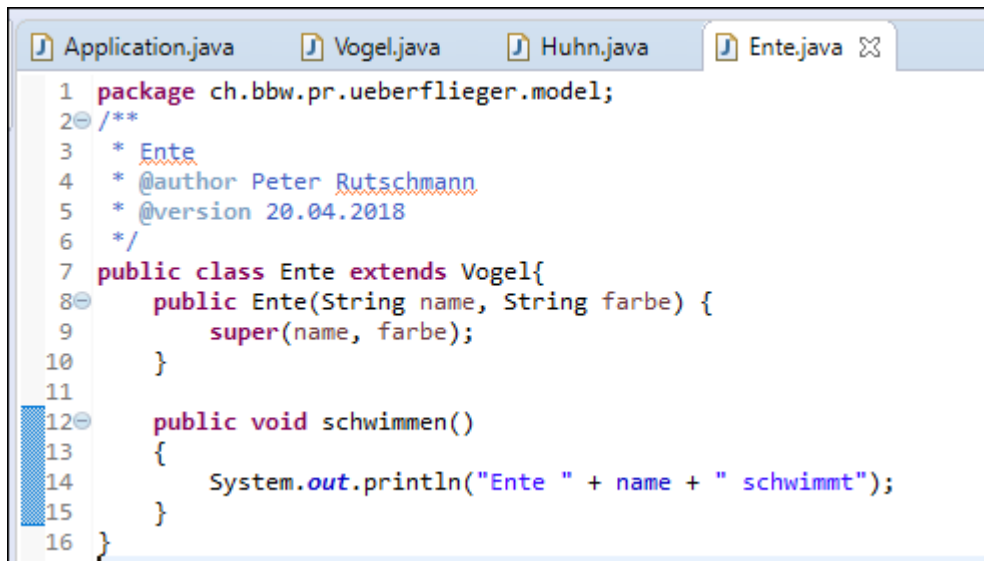
Huhn.java  Vogel.java
1 package ch.bbw.pr.ueberflieger.model;
2 /**
3  * Vogel
4  * @author Peter Rutschmann
5  * @version 20.04.2018
6  */
7 public class Vogel {
8     protected String name;
9     protected String farbe;
10
11     public Vogel() {
12         name = "unbekannt";
13         farbe = "unbekannt";
14     }
15
16     public Vogel(String name, String farbe) {
17         super();
18         this.name = name;
19         this.farbe = farbe;
20     }
21
22     public void fliegen()
23     {
24         System.out.println("Vogel " + name + " fliegt.");
25     }
26
27     public void flattern()
28     {
29         System.out.println("Vogel " + name + " flattert.");
30     }
31 }

```

```

Huhn.java  Vogel.java
1 package ch.bbw.pr.ueberflieger.model;
2
3 /**
4  * Huhn
5  *
6  * @author Peter Rutschmann
7  * @version 28.03.2019
8  */
9 public class Huhn extends Vogel {
10
11     public Huhn() {
12         // super() ruft den Konstruktor der Basisklasse auf.
13         super();
14     }
15
16     public Huhn(String name, String farbe) {
17         /*
18          * super(name, vorname) ruft den Konstruktor der Basisklasse auf und übergibt
19          * dabei den Namen und den Vornamen. Der Konstruktor hat also zwei Parameter.
20          */
21         super(name, farbe);
22     }
23
24     @Override
25     public void fliegen() {
26         System.out.println("Huhn " + name + " fliegt.");
27     }
28
29     public void gehtAuchFlattern() {
30         /* Aufruf der Methode der Basisklasse geht mit flattern
31          * oder mit super.flattern.
32          */
33         flattern();
34         super.flattern();
35
36         // Hier wird explizit das fliegen der Basisklasse Vogel aufgerufen
37         super.fliegen();
38     }
39 }

```



```
1 package ch.bbw.pr.ueberflieger.model;
2 /**
3  * Ente
4  * @author Peter Rutschmann
5  * @version 20.04.2018
6  */
7 public class Ente extends Vogel{
8     public Ente(String name, String farbe) {
9         super(name, farbe);
10    }
11
12    public void schwimmen()
13    {
14        System.out.println("Ente " + name + " schwimmt");
15    }
16 }
```