

```
1 package vorlesung;
2
3 import vorlesung.bankprojekt.verarbeitung.Kontoart;
4
5 /**
6  * testet die Möglichkeiten einer Enum aus.
7  * @author Dorothea Hubrich
8  *
9  */
10 public class EnumTest {
11
12     /**
13      * gibt alle Einträge der Enum Kontoart aus
14      * @param args wird nicht verwendet
15      */
16     public static void main(String[] args) {
17         //Man kann Variablen von der Enum deklarieren - sie ist
18         //ein Datentyp
19         Kontoart k;
20
21         //k = new Kontoart(); geht nicht!!!
22         //man greift auf die Konstante über den Namen der Enum zu, genauso
23         //wie auf andere statische Elemente in Klassen
24         k = Kontoart.FESTGELDKONTO;
25
26         //Für jede Enum gibt es die statische Methode valueOf, die die
27         //zum Namen gehörende Konstante liefert. Wenn man sich beim Namen
28         //verschreibt, wirft sie eine IllegalArgumentException.
29         k = Kontoart.valueOf("GIROKONTO");
30
31         //Jede Enum-Konstante hat automatisch die beiden Methoden
32         //name() und ordinal. Sie liefern den Namen der Konstanten (hier
33         //also "GIROKONTO" und die Nummer der Konstanten (hier 0).
34         //intern nummeriert Java die Konstanten also durch, obwohl es
35         //sich tatsächlich um Objekte handelt und nicht einfach um int-Zahlen
36         //wie in anderen Programmiersprachen
37         //name() und ordinal() sind in der Enum nicht überschreibbar.
38         System.out.println(k.name() + " " + k.ordinal());
39
40         //Jede Enum hat automatisch die Methode values(), die ein
41         //Array aller enthaltenen Konstanten liefert. Das ist praktisch, wenn
42         //man mit allen etwas machen will. Hier z.B. kann man eine
43         //Werbebrochure mit allen angebotenen Kontoarten schreiben.
44         Kontoart[] alle = Kontoart.values();
45         for(int i =0; i<alle.length; i++)
46         {
47             //Wie bei allen anderen Objekten auch kann man auf die
48             //Methoden der Konstanten zugreifen.
49             //toString ist natürlich überschreibbar, in der Standard-
50             //implementierung liefert sie nur den Konstantennamen
51             System.out.println(alle[i] + alle[i].getWerbebotschaft());
52         }
53
54     }
55
56 }
57
```