

JAVA Programmierung

ECLIPSE & JAVA 8



Themenübersicht

01

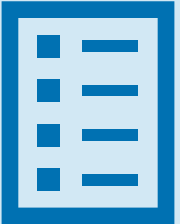
Enum

1. Generelles
2. Beispiele

02

Aufgabe

1. Aufgabe
2. Lösung





Java

01 Enum

1. *Generelles*
2. *Beispiele*



1.1

Enum

Generelles



1.1 Enum

- Generelles

Enums deklarieren einen festen Satz an **Konstanten***. Zur Deklaration wird das Schlüssel Wort „enum“ verwendet.

```
enum Farbe // Enum-Typ
{
    ROT, GRUEN, BLAU; // Enumerationskonstanten
}
```

Enum Konstanten werden in Großbuchstaben dargestellt.

Der Name leitet sich von **Enumerations** ab und bezieht sich auf die Eigenschaft einer Aufzählung. Wir können jetzt über den Namen des Enums auf unsere Konstanten zugreifen.

```
System.out.println(Farbe.BLAU);
```

***Konstanten** => sind Variablen, die nach einmaligem Initialisieren nicht mehrverändert werden können.

1.1 Enum

- Generelles

Es ist möglich, den Konstanten zusätzliche Attribute zu übergeben. Hierfür werden die Enum Konstanten mit dem Attribut in runden Klammern erweitert. Über den Konstruktor können wir dem jeweiligen Element das Attribut zuweisen.

```
enum Farbe // Enum-Typ
{
    // innerhalb der Klammer können eigenschaften der jeweiligen Konstanten übergeben werden
    // an den Konstruktor => hierzu später mehr
    ROT("lieblingsfarbe"), GRUEN("die Natur"), BLAU("der Himmel"); // Enumerationskonstanten

    String meineEigenschaft; // Member der jeweiligen Konstanten

    Farbe(String eigenschaft){ // Konstruktor des Enum-Typ
        meineEigenschaft = eigenschaft; //
    }
}
```

⇒ <https://docs.oracle.com/javase/7/docs/api/java/lang/Enum.html>

1.1 Enum

- **Generelles**

Über die Punkt-Notation kann auf die jeweiligen Konstanten des Enums zugegriffen werden. Zudem bietet es mit der Methode „values()“ die Möglichkeit, mittels Schleife über das Enum Typ zu iterieren.

```
System.out.println(Farbe.);  
  
for (Farbe f : Farbe.values())  
    System.out.println(f);  
}
```

- BLAU : FirstClass.Farbe - FirstClass.Farbe
- class : Class<com.FirstClass.Farbe>
- GRUEN : FirstClass.Farbe - FirstClass.Farbe
- ROT : FirstClass.Farbe - FirstClass.Farbe
- valueOf(String arg0) : Farbe - Farbe
- values() : Farbe[] - Farbe
- valueOf(Class<T> enumType, String name) : T - Enum

1.2

Enum Beispiele



1. Enum

- Beispiel 1

```
public enum WeekDay {  
    MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY, SUNDAY  
}  
  
public static void main(String[] args) {  
    WeekDay day = WeekDay.MONDAY;  
    System.out.println(day);  
}
```

Output:

```
<terminated> ErsteKlasse [Java A  
MONDAY
```

1. Enum

- Beispiel 2 (fortgeschrittene Techniken): Enums mit Values und Streams

```
public enum Programmiersprachen {  
    CSharp("not my World"), Java("so easy"), JavaScript("no types"), Perl("never work with this"), Phython("can be easy"), PHP("What...?"), Ruby("hä...?");  
  
    private String typeOf;  
  
    Programmiersprachen(String typeOf) {  
        this.typeOf = typeOf;  
    }  
    public static Stream<Programmiersprachen> stream() {  
        return Stream.of(Programmiersprachen.values());  
    }  
}  
  
public static void main(String[] args) {  
    Programmiersprachen.stream()  
        .filter(d -> d.typeOf.equals("so easy"))  
        .forEach(System.out::println);  
}
```

Output:

```
<terminated> ErsteKlas  
Java
```



02 Aufgabe

1. *Aufgabe*
2. *Lösung*



2.1 Aufgabe

Aufgabe



2.1 Aufgabe

Erstellen Sie ein Enum, das Programmiersprachen enthält und geben Sie mit einer Schleife alle Sprachen einmal aus.

2.2 Aufgabe Lösung



2.2 Lösung

Erstellen Sie ein Enum, das Programmiersprachen enthält und geben Sie mit einer Schleife alle Sprachen einmal aus.

```
5 public enum Programmiersprachen {  
6     CSharp, Java, JavaScript, Perl, Python, PHP, Ruby  
7 }  
8  
9 public static void main(String[] args) {  
10  
11     for(Programmiersprachen e : Programmiersprachen.values()) {  
12         System.out.println(e);  
13     }  
14 }
```



VIELEN DANK!

