



UNIVERSITÄT
LEIPZIG

Einführung in die Objekt-Orientierte Modellierung und Programmierung

Wintersemester 2025/2026

Dirk Zeckzer

Institut für Informatik



Teil XXIII

Aufzählungen

Aufzählungen

- ▶ Anwendung
 - ▶ Kleine, endliche Anzahl von Werten
- ▶ Beispiele
 - ▶ Wahrheitswerte: TRUE, FALSE
 - ▶ Wochentage: Montag, Dienstag,
...
...
- ▶ enum
 - ▶ Anstelle von `class` oder `interface`
 - ▶ Sichtbarkeit und Name wie bei `class` oder `interface`

Aufzählungen

```
1  public enum Wochentage {  
2      MONTAG,  
3      DIENSTAG,  
4      MITTWOCH,  
5      DONNERSTAG,  
6      FREITAG,  
7      SAMSTAG,  
8      SONNTAG  
9  }  
  
1  public static void printName(  
2      final Wochentage wochentag  
3  ) {  
4      switch (wochentag) {  
5          case MONTAG:  
6              System.out.println("Montag");  
7              break;  
8          case DIENSTAG:  
9              System.out.println("Dienstag");  
10             break;  
11            case MITTWOCH:  
12                System.out.println("Mittwoch");  
13                break;  
14            case DONNERSTAG:  
15                System.out.println("Donnerstag");  
16                break;  
17            case FREITAG:  
18                System.out.println("Freitag");  
19                break;  
20            case SAMSTAG:  
21                System.out.println("Samstag");  
22                break;  
23            case SONNTAG:  
24                System.out.println("Sonntag");  
25                break;  
26        }  
27    }
```

Aufzählungen

```
1 public enum Wochentage {  
2     MONTAG,  
3     DIENSTAG,  
4     MITTWOCH,  
5     DONNERSTAG,  
6     FREITAG,  
7     SAMSTAG,  
8     SONNTAG  
9 }
```

► ; am Ende der Aufzählung **optional**

Aufzählungen

- ▶ Typsicherheit
- ▶ Kein default branch notwendig, da alle Fälle abgedeckt sind

```
1  public static void printName(
2      final Wochentage wochentag
3  ) {
4      switch (wochentag) {
5          case MONTAG:
6              System.out.println("Montag");
7              break;
8          case DIENSTAG:
9              System.out.println("Dienstag");
10             break;
11            case MITTWOCH:
12                System.out.println("Mittwoch");
13                break;
14            case DONNERSTAG:
15                System.out.println("Donnerstag");
16                break;
17            case FREITAG:
18                System.out.println("Freitag");
19                break;
20            case SAMSTAG:
21                System.out.println("Samstag");
22                break;
23            case SONNTAG:
24                System.out.println("Sonntag");
25                break;
26        }
27    }
```

Aufzählungen

```
1  public enum Wochentage {  
2      MONTAG("Montag"),  
3      DIENSTAG("Dienstag"),  
4      MITTWOCH("Mittwoch"),  
5      DONNERSTAG("Donnerstag"),  
6      FREITAG("Freitag"),  
7      SAMSTAG("Samstag"),  
8      SONNTAG("Sonntag");  
9  
10     private final String description;  
11  
12     private Wochentage(  
13         final String description  
14     ) {  
15         this.description = description;  
16     }  
17  
18     @Override  
19     public String toString() {  
20         return description;  
21     }  
22 }
```

► ; am Ende der Aufzählung **obligatorisch**

Aufzählungen

```
1  public enum Wochentage {  
2      MONTAG("Montag"),  
3      DIENSTAG("Dienstag"),  
4      MITTWOCH("Mittwoch"),  
5      DONNERSTAG("Donnerstag"),  
6      FREITAG("Freitag"),  
7      SAMSTAG("Samstag"),  
8      SONNTAG("Sonntag");  
9  
10     private final String description;  
11  
12     private Wochentage(  
13         final String description  
14     ) {  
15         this.description = description;  
16     }  
17  
18     @Override  
19     public String toString() {  
20         return description;  
21     }  
22 }
```

```
1  public class EnumMain {  
2  
3      public static void main(  
4          String[] args  
5      ) {  
6          printName(Wochentage.MONTAG);  
7      }  
8  
9      public static void printName(  
10         final Wochentage wochentag  
11     ) {  
12         System.out.println(wochentag);  
13     }  
14 }
```

Aufzählungen

```
1  public enum Wochentage {  
2      MONTAG("Montag"),  
3      DIENSTAG("Dienstag"),  
4      MITTWOCH("Mittwoch"),  
5      DONNERSTAG("Donnerstag"),  
6      FREITAG("Freitag"),  
7      SAMSTAG("Samstag"),  
8      SONNTAG("Sonntag");  
9  
10     private final String description;  
11  
12     private Wochentage(  
13         final String description  
14     ) {  
15         this.description = description;  
16     }  
17  
18     @Override  
19     public String toString() {  
20         return description;  
21     }  
22  
23     ...  
1         ...  
2  
3     public boolean isWochenende() {  
4         return wochenende.contains(this);  
5     }  
6  
7     private static final  
8         EnumSet<Wochentage> wochenende  
9         = EnumSet.of(Wochentage.SAMSTAG,  
10                      Wochentage.SONNTAG);  
11  
12 }
```

Aufzählungen: Sprachauswahl

```
1  public enum Wochentage {  
2      MONTAG(Messages.getString("Montag")),  
3      DIENSTAG(Messages.getString("Dienstag")),  
4      MITTWOCH(Messages.getString("Mittwoch")),  
5      DONNERSTAG(Messages.getString("Donnerstag")),  
6      FREITAG(Messages.getString("Freitag")),  
7      SAMSTAG(Messages.getString("Samstag")),  
8      SONNTAG(Messages.getString("Sonntag")));  
9  
10     private final String description;  
11  
12     private Wochentage(  
13         final String description  
14     ) {  
15         this.description = description;  
16     }  
17  
18     @Override  
19     public String toString() {  
20         return description;  
21     }  
22 }
```

Anpassung an die Ausgabesprache

- ▶ Oberbegriff: internationalization (i18n)
- ▶ Sprache: 2-Zeichen Abkürzungen (ISO 639)
 - ▶ de: Deutsch
 - ▶ en: Englisch
 - ▶ fr: Französisch
 - ▶ it: Italienisch
 - ▶ zh: Chinesisch
- ▶ Kombinationen:
deutsche Sprache & Land
 - ▶ de_DE
 - ▶ de_AT
 - ▶ de_CH
- ▶ Hier: Locale
- ▶ Land: 2-Zeichen Abkürzungen (ISO 3166)
 - ▶ DE: Deutschland
 - ▶ AT: Österreich
 - ▶ CH: Schweiz
 - ▶ LU: Luxemburg
- ▶ Kombinationen:
Sprache & Schweiz
 - ▶ de_CH
 - ▶ fr_CH
 - ▶ it_CH

Anpassung an die Ausgabesprache

- ▶ ResourceBundle
- ▶ Format: <Schlüssel>=<Wert>
- ▶ Name: <BasisName>[_<Sprache>[_<Land>]].properties
- ▶ Beispiele
 - ▶ messages.properties
 - ▶ messages_de.properties
 - ▶ messages_de_DE.properties

Resource-Dateien (Property Files)

messages_de.properties

```
1 Montag=Montag  
2 Dienstag=Dienstag  
3 Mittwoch=Mittwoch  
4 Donnerstag=Donnerstag  
5 Freitag=Freitag  
6 Samstag=Samstag  
7 Sonntag=Sonntag
```

messages_en.properties

```
1 Montag=Monday  
2 Dienstag=Tuesday  
3 Mittwoch=Wednesday  
4 Donnerstag=Thursday  
5 Freitag=Friday  
6 Samstag=Saturday  
7 Sonntag=Sunday
```

Aufzählungen: Sprachauswahl

```
1 import java.util.MissingResourceException;
2 import java.util.ResourceBundle;
3
4 class Messages {
5
6     private static final String BUNDLE_NAME = "eos.messages"; //NON-NLS-1
7
8     private static final ResourceBundle RESOURCE_BUNDLE = ResourceBundle.getBundle(BUNDLE_NAME);
9
10    private Messages() {
11    }
12
13    public static String getString(
14        final String key
15    ) {
16        try {
17            return RESOURCE_BUNDLE.getString(key);
18        } catch (MissingResourceException e) {
19            return '!' + key + '!';
20        }
21    }
22 }
```

Aufzählungen: Sprachauswahl

```
1  public enum Wochentage {  
2      MONTAG(Messages.getString("Montag")),  
3      DIENSTAG(Messages.getString("Dienstag")),  
4      MITTWOCH(Messages.getString("Mittwoch")),  
5      DONNERSTAG(Messages.getString("Donnerstag")),  
6      FREITAG(Messages.getString("Freitag")),  
7      SAMSTAG(Messages.getString("Samstag")),  
8      SONNTAG(Messages.getString("Sonntag")));  
9  
10     private final String description;  
11  
12     private Wochentage(  
13         final String description  
14     ) {  
15         this.description = description;  
16     }  
17  
18     @Override  
19     public String toString() {  
20         return description;  
21     }  
22 }
```