

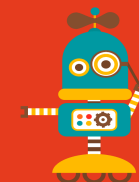


String Formatierung

DI Reinhold Buchinger

Creative Commons-Lizenz CC BY-NC-SA 4.0 AT.

Höhere Abteilung für Mechatronik
Höhere Abteilung für Informationstechnologie
Fachschule für Informationstechnik



Ziele

- » Das Prinzip von `String.format()` und ähnlichen Befehlen verstehen und anwenden können.

Möglicher Anwendungsfall (von vielen)

```
String[] dish = new String[]{"Spätzle", "Palatschinken"};
double[] price = new double[]{8.4, 7.2};
double discount=0.1;
```

```
System.out.println(dish[0]+" "+price[0]*(1-discount)+"€");
System.out.println(dish[1]+" "+price[1]*(1-discount)+"€");
System.out.println();
```

Ausgabe:

```
Spätzle 7.56000000000000000005€
Palatschinken 6.48€
```

Gewünschte Ausgabe:

Spätzle	7,56€
Palatschinken	6,48€



Aufbau String Formatierung

```
String s= String.format("format-String",arg1,arg2,...,argN);
```

Der format-String besteht aus:

- » String Literale (i.e. normaler Text)
- » Format-Angabe ("Platzhalter" für die Argumente)
- » Format-Angaben beginnen mit %

```
int zahl = 4;
```

```
String s= String.format("%d Semmel", zahl); // 4 Semmel
```

Format-Angabe

% [flags] [width] [.precision] conversion-character

- Bis auf den conversion-character sind alle anderen Angaben optional.
- Im angeführten Beispiel ist der conversion-character d

```
int zahl = 4;
```

```
String s= String.format("%d Semmel", zahl); //4 Semmel
```

conversion-character

% [flags] [width] [.precision] conversion-character

» Eine Auswahl an möglichen conversion-character:

d : decimal integer [byte, short, int, long]

f : floating-point number [float, double]

c : character

C : character uppercase

s : String

S : String uppercase

tl: A date & time conversion—hour in 12-hour clock

...

String input = "Semmel";

int zahl = 4;

String s= String.format("%d %s", zahl, input); //4 Semmel

precision

% [flags] [width] [.precision] conversion-character

- » precision Angaben starten mit einem Punkt (.)
- » Bei Kommazahlen gibt precision, die Anzahl der Nachkommastellen an. Die Zahl wird gerundet.
- » Bei Strings die Anzahl der Zeichen, die maximal ausgegeben werden sollen.

String input = "Java";

double zahl = 434.123456;

String s= String.format("%.4f %.2s", zahl, input); //434,1235 Ja

width

% [flags] [width] [.precision] conversion-character

- » Gibt die minimale Anzahl an Zeichen an.
- » Sollte der Wert weniger Zeichen haben, wird der Platz mit Leerzeichen aufgefüllt.

```
String input = "Java";  
double zahl = 434.123456;  
String s= String.format("|%20f %3s", zahl, input);  
// |          434,123456 Java
```


flags

% [flags] [width] [.precision] conversion-character

» Mögliche Flags sind:

- : links ausgerichtet (in Kombination mit width werden die fehlenden Zeichen rechts aufgefüllt)
- + : fügt ein Plus (+) bzw. ein Minus (-) vor numerische Werte
- 0 : füllt den Platz mit 0 statt mit Leerzeichen bei numerischen Werten
- , : Fügt ein Gruppierungszeichen für Zahlen > 1000 ein (Leerzeichen Punkt oder Komma je nach Locale)
- : Leerzeichen - Minus(-) bei negativen Zahlen und Leerzeichen bei positiven Zahlen

String input = "Java";

double zahl = 33434.123456;

String s= String.format("|%+,-20f %s", zahl, input);

// |+33 434,123456 Java

Befehle

- » Verschiedene Befehle in Java verwenden solche Formatierungsstrings.

// liefert einen String zurück

[String.format](#)(format, arg1,..., argN);

[String.format](#)(Locale l, format, arg1,..., argN);

// gibt den String auf System.out aus

[System.out.format](#)(format, arg1,..., argN);

[System.out.format](#)(Locale l, format, arg1,..., argN);

- » Auch in vielen anderen Programmiersprachen kommt das gleiche Konzept zur Anwendung.
- » Komplete Dokumentation: [Java API - Formatter](#)

Anwendungsfall

```
String[] dish = new String[]{"Spätzle", "Palatschinken"};  
double[] price = new double[]{8.4, 7.2};  
double discount=0.1;
```

????

Gewünschte Ausgabe:

Spätzle	7,56€
Palatschinken	6,48€



Anwendungsfall - Lösung

```
String[] dish = new String[]{"Spätzle", "Palatschinken"};
double[] price = new double[]{8.4, 7.2};
double discount=0.1;
```

```
System.out.format("%-20s %.2f€\n", dish[0], price[0]*(1-discount));
System.out.format("%-20s %.2f€", dish[1], price[1]*(1-discount));
```

Ausgabe:

Spätzle	7,56€
Palatschinken	6,48€



Übungsaufgaben

```
String daytime= "Morgen";  
//Gewünschte Ausgabe: "Guten Morgen!"
```

```
String product = "Äpfel";  
float price = 10/3f;  
//Gewünschte Ausgabe: "Äpfel kosten 3,33€/kg"
```

```
int countdown = 233;  
//Gewünschte Ausgabe: "*00233*"  
//ändert man countdown auf 1233: "*01233*"
```

```
String word ="Donaudampfschiff";  
//Gewünschte Ausgabe: "Donau..."
```