

JAVA Programmierung

ECLIPSE & JAVA 8



Themenübersicht

01

Datentypen

1. Generelles
2. Summary
3. Typumwandlung

02

Aufgabe

1. Aufgabe
2. Lösung

03

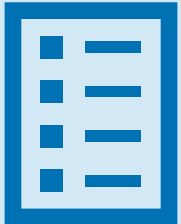
Ausgaben

1. Generelles
2. Beispiele
3. Besonderheiten
bei Zeichenketten

04

Aufgabe

1. Aufgabe
2. Lösung





Java

01 Datentypen

1. *Generelles*
2. *Summary*
3. *Typumwandlung*



1.1

Datentypen

Generelles



1.1 Datentypen

- **Generelles**

Java kennt zwei Datentyp Arten => den primitiven Datentypen und Referenzen auf Objekte.

Mit Referenzen auf Objekte werden wir uns später noch Ausführlich beschäftigen.

Der Grund warum Java theoretisch keine Streng Objekt Orientierte Sprache ist sind die Primitiven Datentypen.

Es wird zwischen 8 primitiven Datentypen unterschieden. Diese grenzen sich durch einem bestimmten Wertebereich und Eigenschaften voneinander ab. Bei Numerischen Typen ist die Größe zu beachten, da dies zu Fehlern führen kann, sollte diese überschritten werden. Zu jedem dieser Typen existiert ein reelles Objekt.

⇒ **Die Wrapper Klassen**

Hierzu zu einem späteren Zeitpunkt mehr

Im Teil Summary finden Sie eine kleine Auflistung der Typen samt Wrapper Klasse und Wertebereich.

Wir beschäftigen uns jetzt mit den primitiven Datentypen.

1.1 Datentypen

- Generelles

```
boolean ersterBoolean = true;
char ersterCharacter = '\u0051'; // Als Unicode => als Q
char zweiterCharacter = 'Q'; // Als Buchstabe
byte ersterByte = 123;
short ersterShort = 12321;
int ersterInteger = 1234567890;
long ersterLong = 922337203685477580L;
float ersterFloat = -3.01f;
double ersterDouble = 16394000000000.987654321;
```

```
System.out.println("erster Boolean: " + ersterBoolean);
System.out.println("erster Character: " + ersterCharacter);
System.out.println("zweiter Character: " + zweiterCharacter);
System.out.println("erster Byte: " + ersterByte);
System.out.println("erster Short: " + ersterShort);
System.out.println("erster Integer: " + ersterInteger);
System.out.println("erster Long: " + ersterLong);
System.out.println("erster Float: " + ersterFloat);
System.out.println("erster Double: " + ersterDouble);
```

Ausgabe auf der Konsole

Zu beachten ist:

Die Datentypen long und float haben eine besondere Kennung:

⇒ Das „L“ am ende des long

⇒ Das „f“ am ende des float

```
long ersterLong = 922337203685477580;
float ersterFloat = -3.01;
```

Ohne Kennung ist es ein Fehler !

```
<terminated> FirstClass [Java Application] C:\Program F
erster Boolean: true
erster Character: Q
zweiter Character: Q
erster Byte: 123
erster Short: 12321
erster Integer: 1234567890
erster Long: 922337203685477580
erster Float: -3.01
erster Double: 1.6394000000000988E13
```

1.2 Datentypen

Summary



1.2 Datentypen

- Summary

Datentyp	Größe	Wrapper-Klasse	Wertebereich
boolean	1 Bit	java.lang.Boolean	true / false (wahr / falsch)
char	16 Bit	java.lang.Character	Unicode Zeichen 16 bit => auf diese verzichten wir hier
byte	8 Bit	java.lang.Byte	-128 - +127
short	16 Bit	java.lang.Short	-32.768 - +32.767
int	32 Bit	java.lang.Integer	ca.-2.000000000 - ca. +2000000000
long	64 Bit	java.lang.Long	ca. - neun Trillionen – ca. +neun Trillionen
float	32 Bit	java.lang.Float	Es gilt ein Wert in 32 Bit Format siehe int => Gleitkommazahl
double	64 Bit	java.lang.Double	Es gilt ein Wert in 64 Bit Format siehe long => Gleitkommazahl (doppelt so genau)

1.3 Datentypen

Typumwandlung



1.3 Datentypen

- Typumwandlung (Implizites Casten)

Generell lassen sich alle primitiven Datentypen in den jeweiligen Datentypen mit höherer Wertigkeit, ohne Werteverlust, packen. Zu beachten ist das auch der char als int dargestellt werden kann.

Siehe Ascii Tabelle. => <https://tools.piex.at/ascii-tabelle/>

Dies nennt man => Implizites Casten

```
14 byte ersterByte = 123;
15 short ersterShort = 12321;
16 char zweiterCharacter = 'Q'; // Als Buchstabe
17 int charAlsInt = zweiterCharacter;
18 int byteAlsInt = ersterByte;
19 int shortAlsInt = ersterShort;
20
21 System.out.println("zweiter Character als Int: " + charAlsInt);
22 System.out.println("erster Byte als Int: " + byteAlsInt);
23 System.out.println("erster Short als Int: " + shortAlsInt);
24
```

```
<terminated> FirstClass [Java Application] C
zweiter Character als Int: 81
erster Byte als Int: 123
erster Short als Int: 12321
```

1.3 Datentypen

- Typumwandlung (Explizites Casten)

Datentypen welche in einen Datentypen mit niedriger Wertigkeit gepackt werden sollen, brauchen eine sogenannte Explizite Umwandlung => „cast“ . Hierbei muss der Cast Operator verwendet werden.

```
int ersterInteger = 1234567890;  
short intAlsShort = (short) ersterInteger;  
System.out.println("erster Int als short " + intAlsShort);
```

Der Cast-Operator ist
der jeweilige Datentyp
,in Klammern, des zu
erstellenden Datentypen

Achtung ⚠ => dies verursacht immer einen Werteverlust.

=>

```
erster Int als short 722
```

Aus => 1234567890 wurde 722

1.3 Datentypen

- **Typumwandlung (Besonderheiten)**

1. Der primitive Datentyp „boolean“ kann in keinen anderen Datentyp umgewandelt werden. Da dieser nur die Werte für Wahr und Falsch (True und False) annehmen kann.
2. Vielleicht ist es Ihnen schon aufgefallen. Der Datentyp String => welcher eine Zeichenfolge darstellt <= taucht hier gar nicht auf. Das liegt daran, dass es für diesen Typen lediglich eine Objekt Instanz gibt. Also nur die Wrapper Klasse. Wir werden uns mit dem Thema String noch gesondert auseinandersetzen!



02 Aufgabe

1. *Aufgabe*
2. *Lösung*



2.1

Aufgaben

Aufgabe



2.1 Aufgabe

Erstelle ein Programm welches alle Datentypen einmal ausgibt.

2.2

Aufgabe

Lösung



2.2 Lösung

```
boolean ersterBoolean = true;
char ersterCharacter = '\u0051'; // Als Unicode => als Q
char zweiterCharacter = 'Q'; // Als Buchstabe
byte ersterByte = 123;
short ersterShort = 12321;
int ersterInteger = 1234567890;
long ersterLong = 922337203685477580L;
float ersterFloat = -3.01f;
double ersterDouble = 16394000000000.987654321;

System.out.println("erster Boolean: " + ersterBoolean);
System.out.println("erster Character: " + ersterCharacter);
System.out.println("zweiter Character: " + zweiterCharacter);
System.out.println("erster Byte: " + ersterByte);
System.out.println("erster Short: " + ersterShort);
System.out.println("erster Integer: " + ersterInteger);
System.out.println("erster Long: " + ersterLong);
System.out.println("erster Float: " + ersterFloat);
System.out.println("erster Double: " + ersterDouble);
```

```
erster Boolean: true
erster Character: Q
zweiter Character: Q
erster Byte: 123
erster Short: 12321
erster Integer: 1234567890
erster Long: 922337203685477580
erster Float: -3.01
erster Double: 1.6394000000000988E13
```



03 Ausgaben

1. *Generelles*
2. *Beispiele*
3. *Besonderheiten bei Zeichenketten*



3.1 Ausgaben

Generelles



3. Ausgaben

- Generelles

Um für den Anfang mit unserem Programm in irgend einer Weise interagieren zu können benötigen wir die Möglichkeit Daten auszugeben. Hierfür stellt uns Java die Funktion/Methode „`System.out.println();`“ zur Verfügung.

```
public static void main(String[] args) {  
    System.out.println();  
}
```

Diese ermöglicht es uns Variablen auf unsere Konsole Sichtbar zu machen.

```
public static void main(String[] args) {  
    System.out.println("Hallo Welt");  
}
```



```
<terminated> FirstClass [Java Application] C:\Program Files\Java\jre1.8.0_321\bin\javaw.exe (25.03.2022, 12:30:17)  
Hallo Welt
```

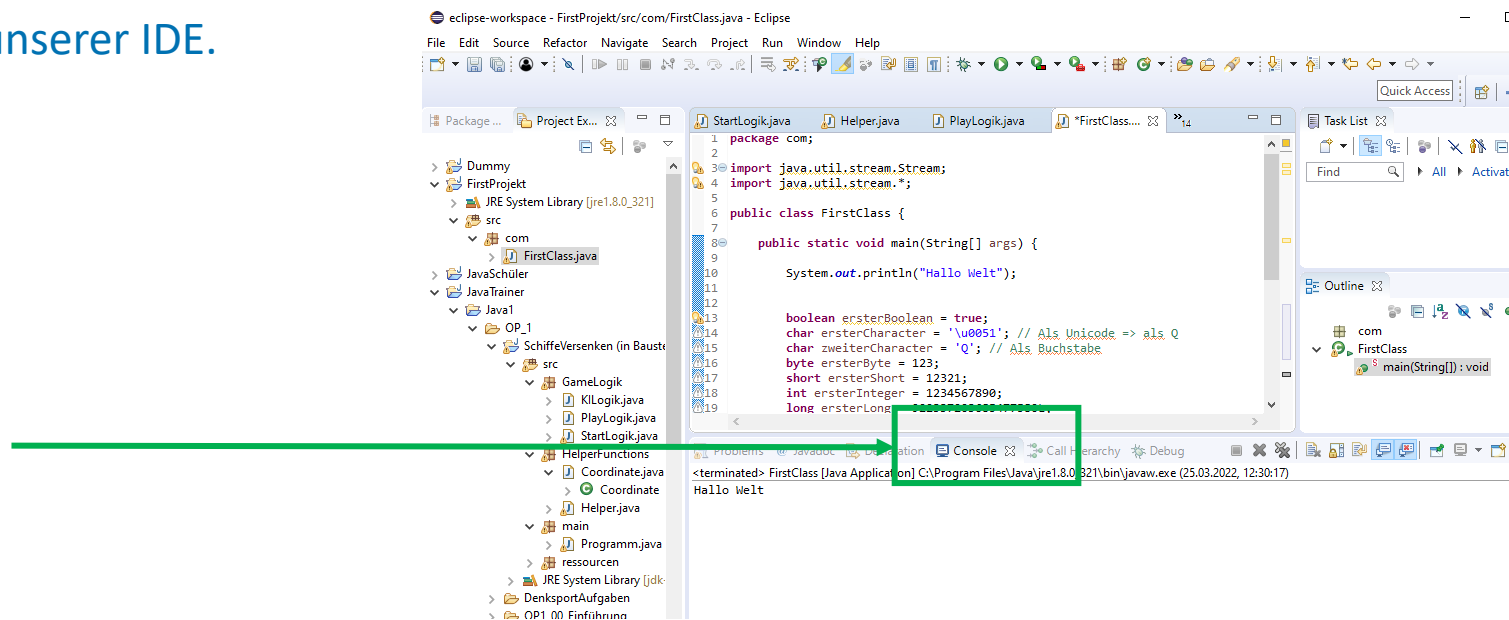
Zum Thema Funktion/Methoden später mehr

3. Ausgaben

- Generelles

Die Konsole finden wir in unserer IDE.

hier



Zeichenketten => eine Abfolge von Buchstaben müssen immer in Hochkommata geschrieben werden.
Variablen können hier direkt übergeben werden.

3.1 Ausgaben

Beispiele



3. Ausgaben

- Beispiele

```
5 public class FirstClass {  
7     public static int meineErsteVariable = 3;  
8 }  
9  
10 public static void main(String[] args) {  
11     System.out.println("Hallo Welt");  
12     System.out.println(meineErsteVariable);  
13 }
```

Ausgabe von:
„Hallo Welt“ und der Variable meineErsteVariable
auf der Eclipse eigenen Konsole

```
<terminated> FirstClass [Java Application] C:\Program Files\Java\jre1.8.0_321\bin\javaw.exe (25.03.2022, 12:39:40)  
Hallo Welt  
3
```

3. Ausgaben

- Beispiele

```
boolean ersterBoolean = true;
char ersterCharacter = '\u0051'; // Als Unicode => als Q
char zweiterCharacter = 'Q'; // Als Buchstabe
byte ersterByte = 123;
short ersterShort = 12321;
int ersterInteger = 1234567890;
long ersterLong = 922337203685477580L;
float ersterFloat = -3.01f;
double ersterDouble = 16394000000000.987654321;

System.out.println("erster Character: " + ersterCharacter);
System.out.println("zweiter Character: " + zweiterCharacter);
System.out.println("erster Byte: " + ersterByte);
System.out.println("erster Short: " + ersterShort);
System.out.println("erster Integer: " + ersterInteger);
System.out.println("erster Long: " + ersterLong);
System.out.println("erster Float: " + ersterFloat);
System.out.println("erster Double: " + ersterDouble);
```

```
erster Character: Q
zweiter Character: Q
erster Byte: 123
erster Short: 12321
erster Integer: 1234567890
erster Long: 922337203685477580
erster Float: -3.01
erster Double: 1.6394000000000988E13
```


3.1

Ausgaben

Besonderheiten bei Zeichenketten



3. Ausgaben

- Besonderheiten bei Zeichenketten

Achtung

Den Datentypen String werden wir gesondert behandeln. Allerdings gibt es im Bezug auf Ausgaben Besonderheiten welche wir beachten müssen. Hierfür betrachten wir den Operator „+“ etwas genauer. Zum einen dient er als Additionszeichen zum anderen lassen sich mit ihm Zeichenketten und Variablen miteinander Verkettan / Konkatenation => concatenation.

```
public static int meineErsteVariable = 3;

public static void main(String[] args) {
    System.out.println("1223" + " " + "Welt");
    System.out.println("123" + meineErsteVariable);
    System.out.println("3+3" + meineErsteVariable + meineErsteVariable);
    System.out.println("3+3" + (meineErsteVariable + meineErsteVariable));
    System.out.println(meineErsteVariable + meineErsteVariable + "3");
}
```



```
1223 Welt
1233
3+333
3+36
63
```

Die Klammer bekommt die selbe Wertigkeit wie in der Mathematik und wird zuerst ausgewertet

3. Ausgaben

- Besonderheiten bei Zeichenketten

```
public static void main(String[] args) {  
    // Erster Befehl: Konsolenausgabe per System.out.print()  
    System.out.print("Hallo "); // mit print findet kein Zeilenumbruch statt  
    System.out.print("Welt!\n"); // mit " \n " wird jedoch ein Zeilenumbruch erzeugt  
    // Falls wir am Ende einer Konsolenausgabe AUTOMATISCH einen Umbruch haben wollen,  
    // dann können wir auch bequem mit dem Befehl println() statt print() arbeiten:  
    System.out.println("Heute ist die Welt noch in Ordnung...");  
    System.out.println("Wer weiß wie es morgen aussieht?");  
}
```



Java

04 Aufgabe

1. *Aufgabe*
2. *Lösung*



4.1

Aufgaben

Zweite Aufgabe



4.1 Zweite Aufgabe

Erweitere dein Projekt um:

eine Ausgabe, welche auf der Konsole nacheinander

- Hallo Welt
- Ich lerne Java

ausgibt.

4.2

Aufgaben

Lösung



4.2 Lösung

```
9 public static void main(String[] args) {  
10     System.out.println("Hallo Welt");  
11     System.out.println("Ich lerne Java");  
12 }
```

```
<terminated> FirstClass [Java Application] C:\P  
Hallo Welt  
Ich lerne Java
```




VIELEN DANK!

