

Programmierkurs

Einführung in Java Tag 1



Marcel Rettig

Wintersemester 2020

Ablauf

Variablen

Datentypen
Wertebereiche

Variablennamen

Verwendung

Deklaration und
Wertzuweisung
Ausgabe

Arithmetik

Modulorechnung
In- bzw. Dekrement
Integer-Division

Konvertierung

Kommentare

Technologie

Eclipse
Einführung

Quellen & Lizenz

Inhaltsübersicht Vorkurs

- ▶ Tag 1: Variablen, Datentypen, Konvertierungen, Arithmetik, Zustände, Eclipse Livedemo
- ▶ Tag 2: Boolesche Ausdrücke, Kommentare, If-Abfragen, Switch-Case, Debugging
- ▶ Tag 3: Arrays, (Do-)While-Schleife, For-Schleifen, Weiterführung Debugging
- ▶ Tag 4: (statische) Methoden, Klassenvariablen, JavaDoc, Exceptions

Ablauf

Variablen

Datentypen
Wertebereiche

Variablennamen

Verwendung

Deklaration und
Wertzuweisung
Ausgabe

Arithmetik

Modulorechnung
In- bzw. Dekrement
Integer-Division

Konvertierung

Kommentare

Technologie

Eclipse

Einführung

Quellen & Lizenz

Ablauf

- ▶ Vorlesung
- ▶ Mittagspause
- ▶ Gegen 12:30 Übungen



Ablauf

Variablen

Datentypen
Wertebereiche

Variablennamen

Verwendung

Deklaration und
Wertzuweisung
Ausgabe

Arithmetik

Modulorechnung
In- bzw. Dekrement
Integer-Division

Konvertierung

Kommentare

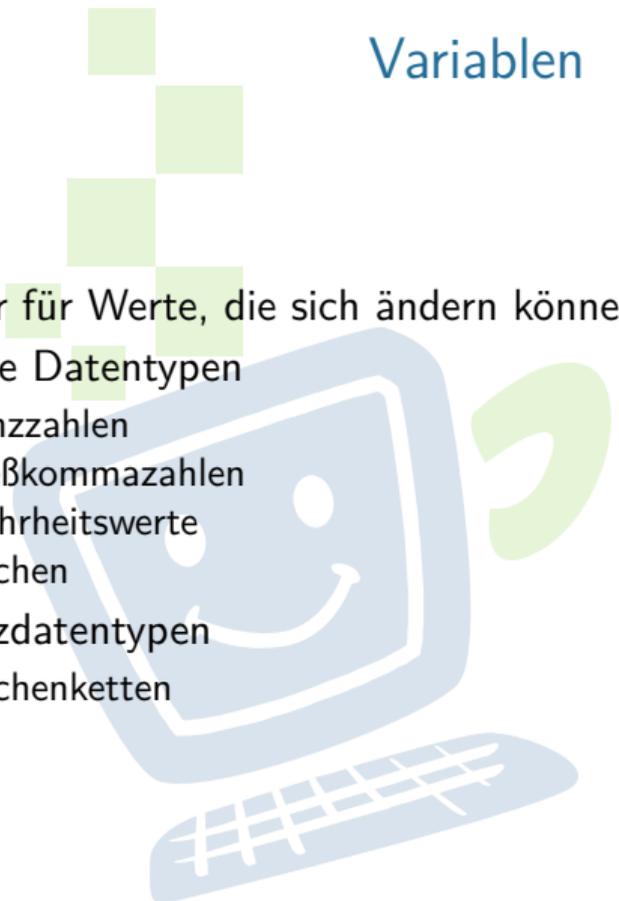
Technologie

Eclipse

Einführung

Quellen & Lizenz

Variablen

- ▶ Speicher für Werte, die sich ändern können
 - ▶ Primitive Datentypen
 - ▶ Ganzzahlen
 - ▶ Fließkommazahlen
 - ▶ Wahrheitswerte
 - ▶ Zeichen
 - ▶ Referenzdatentypen
 - ▶ Zeichenketten
- 

Ablauf

Variablen

Datentypen
Wertebereiche

Variablennamen

Verwendung

Deklaration und
Wertzuweisung
Ausgabe

Arithmetik

Modulorechnung
In- bzw. Dekrement
Integer-Division

Konvertierung

Kommentare

Technologie

Eclipse

Einführung

Quellen & Lizenz

Datentypen

- ▶ Zahlen im Computer sind endlich
- ▶ Ganzzahlen
 - ▶ **byte** (8 Bit / 1 Byte)
 - ▶ **short** (16 Bit / 2 Byte)
 - ▶ **int** (32 Bit / 4 Byte)
 - ▶ **long** (64 Bit / 8 Byte)
- ▶ Kommazahlen (Fließkommazahlen)
 - ▶ **float** (32 Bit / 4 Byte)
 - ▶ **double** (64 Bit / 8 Byte)
- ▶ Unterscheiden sich jeweils nur in ihrem Wertebereich

Ablauf

Variablen

Datentypen

Wertebereiche

Variablennamen

Verwendung

Deklaration und
Wertzuweisung
Ausgabe

Arithmetik

Modulorechnung
In- bzw. Dekrement
Integer-Division

Konvertierung

Kommentare

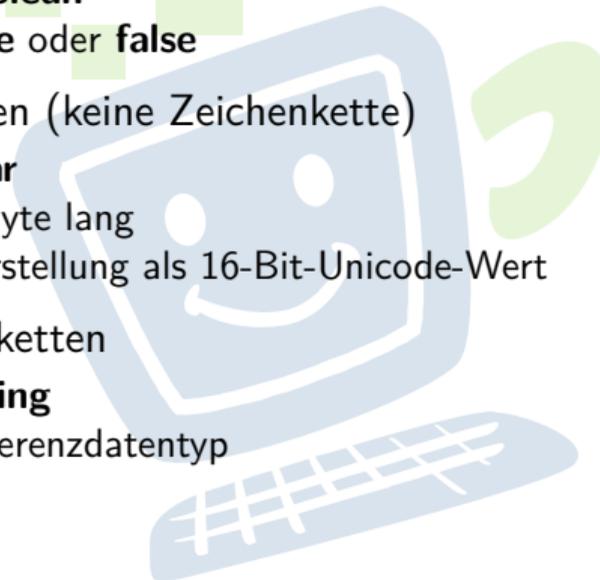
Technologie

Eclipse

Einführung

Quellen & Lizenz

Datentypen

- 
- ▶ Wahrheitswerte
 - ▶ **boolean**
 - ▶ **true** oder **false**
 - ▶ 1 Zeichen (keine Zeichenkette)
 - ▶ **char**
 - ▶ 2 Byte lang
 - ▶ Darstellung als 16-Bit-Unicode-Wert
 - ▶ Zeichenketten
 - ▶ **String**
 - ▶ Referenzdatentyp

Ablauf

Variablen

Datentypen

Wertebereiche

Variablennamen

Verwendung

Deklaration und
Wertzuweisung
Ausgabe

Arithmetik

Modulorechnung
In- bzw. Dekrement
Integer-Division

Konvertierung

Kommentare

Technologie

Eclipse

Einführung

Quellen & Lizenz

Variablen - Wertebereiche

Type	Länge		Wertebereich
	Byte	Bit	
boolean	-	-	true oder false
char	2	16	Unicode Zeichen
byte	1	8	-128 bis 127
short	2	16	-32768 bis 32767
int	4	32	-2.147.483.648 bis 2.147.483.647
long	8	64	-2^{63} bis $2^{63} - 1$
float	4	32	$\pm 1,4^{E-45}$ bis $\pm 3,4^{E+38}$
double	8	64	$\pm 4,9^{E-324}$ bis $\pm 1,7^{E+308}$

Ablauf

Variablen

Datentypen

Wertebereiche

Variablennamen

Verwendung

Deklaration und
Wertzuweisung
Ausgabe

Arithmetik

Modulorechnung
In- bzw. Dekrement
Integer-Division

Konvertierung

Kommentare

Technologie

Eclipse

Einführung

Quellen & Lizenz

Variablennamen (Bezeichner)

► Vorgaben

- So MÜSSEN Namen sein, sonst gibt es Compiler-Fehler
- Erlaubte Zeichen: Buchstaben, Zahlen und `_`
- Erstes Zeichen darf keine Zahl sein

► Gesperrte Namen

- z.B.:

```
true, false, new
```

- Beispiel:

NICHT

```
double 1_variable;
```

richtig:

```
double variable1;
```

Ablauf

Variablen

Datentypen
Wertebereiche

Variablennamen

Verwendung

Deklaration und
Wertzuweisung
Ausgabe

Arithmetik

Modulorechnung
In- bzw. Dekrement
Integer-Division

Konvertierung

Kommentare

Technologie

Eclipse

Einführung

Quellen & Lizenz

Variablennamen

► Konventionen

- So SOLLTEN Namen sein, jedoch kompiliert der Quelltext sobald die Formalen Voraussetzungen erfüllt sind
- sinnvolle, aussagekräftige Namen wählen
- keine Abkürzungen (außer sie sind geläufig)
- Substantive

Beispiel:

NICHT

```
double geldBezahlen;
```

(nicht aussagekräftig, kein Substantiv)

NICHT

```
double ticketBezahlen;
```

(kein Substantiv)

besser:

```
double ticketpreis;
```

Ablauf

Variablen

Datentypen
Wertebereiche

Variablennamen

Verwendung

Deklaration und
Wertzuweisung
Ausgabe

Arithmetik

Modulorechnung
In- bzw. Dekrement
Integer-Division

Konvertierung

Kommentare

Technologie

Eclipse
Einführung

Quellen & Lizenz

Variablennamen

► Konventionen

- Nur lateinische Zeichen (normales Alphabet), Zahlen und _
 - KEIN ä, ö, ü, ß, ...

- Verwendung EINER Sprache, kein Gemisch oder **"Denglisch"**

- lowerCamelCase-Schreibweise

camelCase bedeutet, dass der Bezeichner ohne Trennzeichen wie Leerzeile und Unterstrich angegeben werden und die folgenden Worte groß geschrieben werden. Die lower Variante beginnt das erste Wort klein geschrieben.

Beispiel:

```
int anzahlTiere;  
int koerperHoehe;
```

Ablauf

Variablen

Datentypen
Wertebereiche

Variablennamen

Verwendung

Deklaration und
Wertzuweisung
Ausgabe

Arithmetik

Modulorechnung
In- bzw. Dekrement
Integer-Division

Konvertierung

Kommentare

Technologie

Eclipse

Einführung

Quellen & Lizenz



Zur Verwendung

Ablauf

Variablen

Datentypen
Wertebereiche

Variablennamen

Verwendung

Deklaration und
Wertzuweisung
Ausgabe

Arithmetik

Modulorechnung
In- bzw. Dekrement
Integer-Division

Konvertierung

Kommentare

Technologie

Eclipse

Einführung

Quellen & Lizenz

Deklaration

- ▶ Deklarieren der Variablen:

```
datentyp name;
```

- ▶ Beispiele:

```
int age;  
float weight;  
double height;  
boolean isStudent;  
char gender;  
String name;
```

Ablauf

Variablen

Datentypen
Wertebereiche

Variablennamen

Verwendung

Deklaration und
Wertzuzuweisung

Ausgabe

Arithmetik

Modulorechnung
In- bzw. Dekrement
Integer-Division

Konvertierung

Kommentare

Technologie

Eclipse

Einführung

Quellen & Lizenz

Wertzuweisung

- ▶ Der Variablen einen Wert zuweisen
Beim ersten Mal spricht man von initialisieren

```
name = value;
```

- ▶ Die Variable muss deklariert worden sein
- ▶ Beispiele:

```
int age; age = 20;  
float weight; weight = 4.2f;  
double height; height = 1.74;  
boolean isStudent; isStudent = true;  
char gender; gender = 'm';  
String name; name = "Douglas Adams";
```

Ablauf

Variablen

Datentypen
Wertebereiche

Variablennamen

Verwendung

Deklaration und
Wertzuweisung

Ausgabe

Arithmetik

Modulorechnung
In- bzw. Dekrement
Integer-Division

Konvertierung

Kommentare

Technologie

Eclipse
Einführung

Quellen & Lizenz

Initialisierung

- ▶ Wert direkt beim Deklarieren auch initialisieren:

```
datentyp name = value;
```

- ▶ Beispiele:

```
int age = 20;  
float weight = 4.2f;  
double height = 1.74;  
boolean isStudent = true;  
char gender = 'm';  
String name = "Douglas Adams";
```

Ablauf

Variablen

Datentypen
Wertebereiche

Variablennamen

Verwendung

Deklaration und
Wertzuweisung

Ausgabe

Arithmetik

Modulorechnung
In- bzw. Dekrement
Integer-Division

Konvertierung

Kommentare

Technologie

Eclipse
Einführung

Quellen & Lizenz

Ausgabe

- ▶ Sonst wäre es ziemlich langweilig

```
System.out.println(output);  
System.out.print(output);
```

- ▶ Beispiele:

```
int value = 3;  
System.out.println(value);  
  
System.out.println("Hello_ World");  
  
String name = "World";  
System.out.print("Hello_");  
System.out.print(name);  
System.out.println();
```

Ablauf

Variablen

Datentypen
Wertebereiche

Variablennamen

Verwendung

Deklaration und
Wertzuweisung

Ausgabe

Arithmetik

Modulorechnung
In- bzw. Dekrement
Integer-Division

Konvertierung

Kommentare

Technologie

Eclipse

Einführung

Quellen & Lizenz

Arithmetik

Bezeichnung	Operator	Anwendung
Addition	+	$a + b$
Subtraktion	-	$a - b$
Multiplikation	*	$a * b$
Division	/	a / b
Inkrement	++	$a++$
Dekrement	--	$a--$
Modulo	%	$a \% b$

Ergebnis kann Variablen zugewiesen werden:

```
int result = 5 + 3;  
double division = 3.5 / (result - 1);
```

Ablauf

Variablen

Datentypen
Wertebereiche

Variablenamen

Verwendung

Deklaration und
Wertzuweisung
Ausgabe

Arithmetik

Modulorechnung
In- bzw. Dekrement
Integer-Division

Konvertierung

Kommentare

Technologie

Eclipse
Einführung

Quellen & Lizenz

Arithmetik Beispiele

```
int x = 5;  
x = x + 1;
```

Ist äquivalent zu:

```
x++;
```

```
int y = 10;  
y = y + 5;
```

Ist äquivalent zu:

```
y += 5
```



Ablauf

Variablen

- Datentypen
- Wertebereiche

Variablennamen

Verwendung

- Deklaration und Wertzuweisung
- Ausgabe

Arithmetik

- Modulorechnung
- In- bzw. Dekrement
- Integer-Division

Konvertierung

Kommentare

Technologie

Eclipse

Einführung

Quellen & Lizenz

Modulo (Restwertbestimmung)

- Das Ergebnis des Modulo ist der Rest der Division:

$$26/5 = 5 \text{ Rest } 1 \quad \Rightarrow \quad 26 \% 5 = 1$$

$$30/2 = 15 \text{ Rest } 0 \quad \Rightarrow \quad 30 \% 2 = 0$$

Ablauf

Variablen

Datentypen
Wertebereiche

Variablennamen

Verwendung

Deklaration und
Wertzuweisung
Ausgabe

Arithmetik

Modulorechnung

In- bzw. Dekrement
Integer-Division

Konvertierung

Kommentare

Technologie

Eclipse

Einführung

Quellen & Lizenz

In- bzw. Dekrement

- ▶ Erhöht bzw. verringert den Wert einer Variablen um 1.

```
int x = 42;
x++; // Identisch: x=x+1
x--; // Identisch: x=x-1
```

Welchen Wert beinhaltet x?

Ablauf

Variablen

Datentypen
Wertebereiche

Variablennamen

Verwendung

Deklaration und
Wertzuweisung
Ausgabe

Arithmetik

Modulorechnung
In- bzw. Dekrement
Integer-Division

Konvertierung

Kommentare

Technologie

Eclipse
Einführung

Quellen & Lizenz

In- bzw. Dekrement

- ▶ Erhöht bzw. verringert den Wert einer Variablen um 1.

```
int x = 42;  
x++; // Identisch: x=x+1  
x--; // Identisch: x=x-1
```

Welchen Wert beinhaltet x?

x = 42

Ablauf

Variablen

Datentypen
Wertebereiche

Variablennamen

Verwendung

Deklaration und
Wertzuweisung
Ausgabe

Arithmetik

Modulorechnung
In- bzw. Dekrement
Integer-Division

Konvertierung

Kommentare

Technologie

Eclipse
Einführung

Quellen & Lizenz

Prä- bzw. Postinkrement sind zu beachten

Präinkrement (Prä → Davor)

```
int x = 23;  
System.out.println(++x);
```

Wie lautet die Ausgabe und welchen Wert hat x?

Postinkrement (Post → Danach)

```
int y = 23;  
System.out.println(y++);
```

Wie lautet die Ausgabe und welchen Wert hat y?

Ablauf

Variablen

Datentypen
Wertebereiche

Variablennamen

Verwendung

Deklaration und
Wertzuweisung
Ausgabe

Arithmetik

Modulorechnung
In- bzw. Dekrement

Integer-Division

Konvertierung

Kommentare

Technologie

Eclipse

Einführung
Quellen & Lizenz

Prä- bzw. Postinkrement sind zu beachten

Präinkrement (Prä → Davor)

```
int x = 23;  
System.out.println(++x);
```

Wie lautet die Ausgabe und welchen Wert hat x?

Ausgabe: **24**; x = 24

Postinkrement (Post → Danach)

```
int y = 23;  
System.out.println(y++);
```

Wie lautet die Ausgabe und welchen Wert hat y?

Ablauf

Variablen

Datentypen
Wertebereiche

Variablennamen

Verwendung

Deklaration und
Wertzuweisung
Ausgabe

Arithmetik

Modulorechnung
In- bzw. Dekrement

Integer-Division

Konvertierung

Kommentare

Technologie

Eclipse

Einführung

Quellen & Lizenz

Prä- bzw. Postinkrement sind zu beachten

Präinkrement (Prä → Davor)

```
int x = 23;  
System.out.println(++x);
```

Wie lautet die Ausgabe und welchen Wert hat x?

Ausgabe: **24**; x = 24

Postinkrement (Post → Danach)

```
int y = 23;  
System.out.println(y++);
```

Wie lautet die Ausgabe und welchen Wert hat y?

Ausgabe: **23**; y = 24

Ablauf

Variablen

Datentypen
Wertebereiche

Variablenamen

Verwendung

Deklaration und
Wertzuweisung
Ausgabe

Arithmetik

Modulorechnung
In- bzw. Dekrement

Integer-Division

Konvertierung

Kommentare

Technologie

Eclipse

Einführung

Quellen & Lizenz

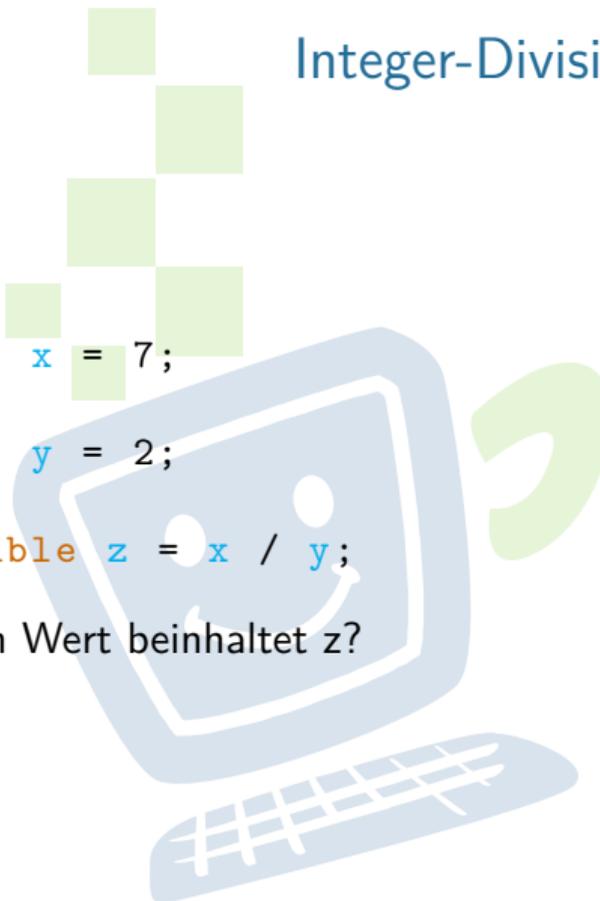
Integer-Division

```
int x = 7;
```

```
int y = 2;
```

```
double z = x / y;
```

Welchen Wert beinhaltet z?



Ablauf

Variablen

Datentypen
Wertebereiche

Variablennamen

Verwendung

Deklaration und
Wertzuweisung
Ausgabe

Arithmetik

Modulorechnung
In- bzw. Dekrement

Integer-Division

Konvertierung

Kommentare

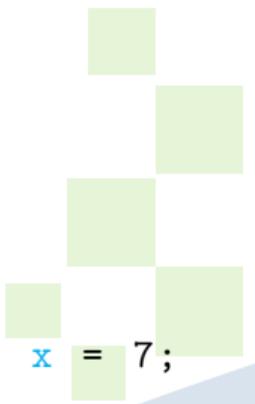
Technologie

Eclipse

Einführung

Quellen & Lizenz

Integer-Division

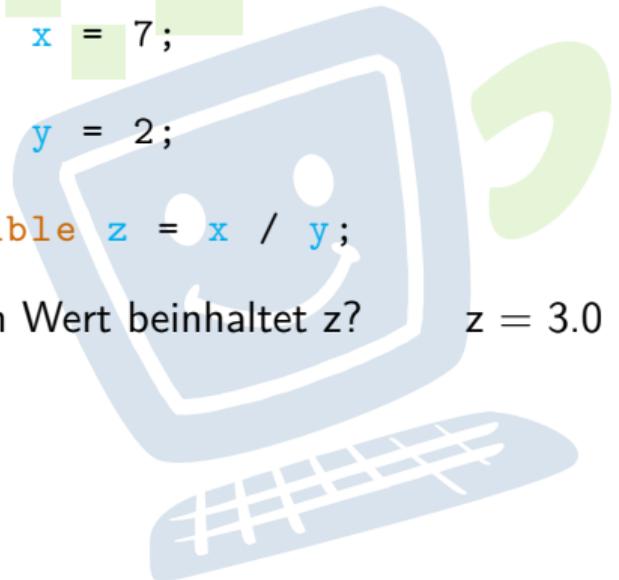


```
int x = 7;
```

```
int y = 2;
```

```
double z = x / y;
```

Welchen Wert beinhaltet z? $z = 3.0$



Ablauf

Variablen

Datentypen
Wertebereiche

Variablennamen

Verwendung

Deklaration und
Wertzuweisung
Ausgabe

Arithmetik

Modulorechnung
In- bzw. Dekrement

Integer-Division

Konvertierung

Kommentare

Technologie

Eclipse

Einführung

Quellen & Lizenz

Implizite Konvertierung

Einige Typen können ohne Probleme in andere umgewandelt werden

byte → **short** → **int** → **long** → **double**

byte → **short** → **int** → **float** → **double**

```
int x = 42;
```

```
float y = x;
```

```
byte a = 123;
```

```
short b = a;
```

```
long c = b;
```

```
double d = c;
```

Ablauf

Variablen

Datentypen
Wertebereiche

Variablennamen

Verwendung

Deklaration und
Wertzuweisung
Ausgabe

Arithmetik

Modulorechnung
In- bzw. Dekrement
Integer-Division

Konvertierung

Kommentare

Technologie

Eclipse

Einführung

Quellen & Lizenz

Explizite Konvertierung

- ▶ Variablenwerte können umgewandelt werden
 - ▶ explizites „Casten“

```
int x = 42;  
double y = (double)x;
```

Welchen Wert beinhaltet y?

```
double a = 512.6;  
int b = (int)a;
```

Welchen Wert beinhaltet b?

Ablauf

Variablen

Datentypen
Wertebereiche

Variablennamen

Verwendung

Deklaration und
Wertzuweisung
Ausgabe

Arithmetik

Modulorechnung
In- bzw. Dekrement
Integer-Division

Konvertierung

Kommentare

Technologie

Eclipse
Einführung

Quellen & Lizenz

Explizite Konvertierung

- ▶ Variablenwerte können umgewandelt werden
 - ▶ explizites „Casten“

```
int x = 42;  
double y = (double)x;
```

Welchen Wert beinhaltet y? $y = 42.0$

```
double a = 512.6;  
int b = (int)a;
```

Welchen Wert beinhaltet b?

Ablauf

Variablen

Datentypen
Wertebereiche

Variablennamen

Verwendung

Deklaration und
Wertzuweisung
Ausgabe

Arithmetik

Modulorechnung
In- bzw. Dekrement
Integer-Division

Konvertierung

Kommentare

Technologie

Eclipse

Einführung

Quellen & Lizenz

Explizite Konvertierung

- ▶ Variablenwerte können umgewandelt werden
 - ▶ explizites „Casten“

```
int x = 42;
double y = (double)x;
```

Welchen Wert beinhaltet y? $y = 42.0$

```
double a = 512.6;
int b = (int)a;
```

Welchen Wert beinhaltet b? $b = 512$

Ablauf

Variablen

Datentypen
Wertebereiche

Variablennamen

Verwendung

Deklaration und
Wertzuweisung
Ausgabe

Arithmetik

Modulorechnung
In- bzw. Dekrement
Integer-Division

Konvertierung

Kommentare

Technologie

Eclipse
Einführung

Quellen & Lizenz

Verknüpfung von Zeichenketten

- ▶ Verknüpfung durch den + -Operator

```
String name = "Hallo, " + "Welt";
```

- ▶ auch gemischt mit Zahlen möglich

```
int x = 5;  
String text = "x hat den Wert " + x;
```

- ▶ Ausgabe:

```
System.out.println("x ist " + x);  
System.out.print("Hallo, " + "Student");
```

Ablauf

Variablen

Datentypen
Wertebereiche

Variablenamen

Verwendung

Deklaration und
Wertzuweisung
Ausgabe

Arithmetik

Modulorechnung
In- bzw. Dekrement
Integer-Division

Konvertierung

Kommentare

Technologie

Eclipse
Einführung

Quellen & Lizenz

Zurück zum Divisionsproblem

```
int x = 7;
```

```
int y = 2;
```

```
double z = x / y;
```

- ▶ Bei Rechnungen wird in den bestmöglichen Typen gecastet
- ▶ So funktioniert es:

```
double z = (double) x / y;
```

Ablauf

Variablen

Datentypen
Wertebereiche

Variablennamen

Verwendung

Deklaration und
Wertzuweisung
Ausgabe

Arithmetik

Modulorechnung
In- bzw. Dekrement
Integer-Division

Konvertierung

Kommentare

Technologie

Eclipse

Einführung

Quellen & Lizenz

Kommentare

- ▶ Werden verwendet um Code von der Verwendung auszunehmen oder Kommentare zu hinterlassen. Wenn wir euch auffordern etwas auszukommentieren reden wir hiervon. Mehr dazu an Tag 2.
- ▶ Einzeilige Kommentare:

```
int number; //Kommentar.  
//number = 3;  
number = 12;
```

Ablauf

Variablen

Datentypen
Wertebereiche

Variablennamen

Verwendung

Deklaration und
Wertzuweisung
Ausgabe

Arithmetik

Modulorechnung
In- bzw. Dekrement
Integer-Division

Konvertierung

Kommentare

Technologie

Eclipse

Einführung

Quellen & Lizenz

Technologie

Quellcode (Java)



Ablauf

Variablen

Datentypen
Wertebereiche

Variablennamen

Verwendung

Deklaration und
Wertzuweisung
Ausgabe

Arithmetik

Modulorechnung
In- bzw. Dekrement
Integer-Division

Konvertierung

Kommentare

Technologie

Eclipse

Einführung

Quellen & Lizenz

Technologie

Quellcode (Java)

↓
Compiler

Zwischencode (Bytecode)

Ablauf

Variablen

Datentypen
Wertebereiche

Variablennamen

Verwendung

Deklaration und
Wertzuweisung
Ausgabe

Arithmetik

Modulorechnung
In- bzw. Dekrement
Integer-Division

Konvertierung

Kommentare

Technologie

Eclipse

Einführung

Quellen & Lizenz

Technologie

Quellcode (Java)

Compiler

Zwischencode (Bytecode)

Laufzeitumgebung (JVM)

Maschinencode

Ablauf

Variablen

Datentypen
Wertebereiche

Variablennamen

Verwendung

Deklaration und
Wertzuweisung
Ausgabe

Arithmetik

Modulorechnung
In- bzw. Dekrement
Integer-Division

Konvertierung

Kommentare

Technologie

Eclipse

Einführung

Quellen & Lizenz

Eclipse Einführung

- ▶ Live Demo (nach der Pause in der Übung)



Ablauf

Variablen

Datentypen
Wertebereiche

Variablennamen

Verwendung

Deklaration und
Wertzweisung
Ausgabe

Arithmetik

Modulorechnung
In- bzw. Dekrement
Integer-Division

Konvertierung

Kommentare

Technologie

Eclipse

Einführung

Quellen & Lizenz

Quellen und Lizenz

FACHSCHAFT INFORMATIK

HS Karlsruhe



- ▶ Original von Tim Roes
- ▶ Überarbeitet 2012 von Christian Zimmer
- ▶ Überarbeitet 2013 von Benedikt Haug
- ▶ Überarbeitet 2016 von Jakob Ernst
- ▶ Überarbeitet 2018 von Sebastian Glutsch
- ▶ Überarbeitet 2019 von Marcel Rettig
- ▶ Überarbeitet 2020 von Anja Buchmaier
- ▶ Überarbeitet 2020 von Marcel Rettig

Ablauf

Variablen

Datentypen
Wertebereiche

Variablennamen

Verwendung

Deklaration und
Wertzuzuweisung
Ausgabe

Arithmetik

Modulorechnung
In- bzw. Dekrement
Integer-Division

Konvertierung

Kommentare

Technologie

Eclipse

Einführung

Quellen & Lizenz