

Programmierkurs

Einführung in Java

Christian Zimmer
WS 2012/2013

Ablauf

Technologie

Variablen

Datentypen

Wertebereiche

Variablennamen

Verwendung

Deklaration und
Wertzuweisung

Ausgabe

Arithmetik

Modulo

In- bzw. Dekrement

Integer-Division

Verknüpfung von
Zeichenketten

Konvertierung

Demonstration

NetBeans

(Debugging)

Programmiervorkurs für Erstsemester

Ablauf

Technologie

Variablen

Variablennamen

Verwendung

Arithmetik

Konvertierung

Demonstration NetBeans (Debugging)

Ablauf

Technologie

Variablen

Datentypen

Wertebereiche

Variablennamen

Verwendung

Deklaration und

Wertzuweisung

Ausgabe

Arithmetik

Modulo

In- bzw. Dekrement

Integer-Division

Verknüpfung von

Zeichenketten

Konvertierung

Demonstration

NetBeans

(Debugging)

Ablauf

Technologie

Variablen

Datentypen

Wertebereiche

Variablennamen

Verwendung

Deklaration und

Wertzuweisung

Ausgabe

Arithmetik

Modulo

In- bzw. Dekrement

Integer-Division

Verknüpfung von

Zeichenketten

Konvertierung

Demonstration

NetBeans

(Debugging)

- ▶ 09:30 Vorstellung der Lösungen des Vortages
- ▶ ab 10:00 Vorlesung
- ▶ 90 min Mittagspause
- ▶ gegen 12:30 / 13:00 Übungen

Quellcode (Java)

Ablauf

Technologie

Variablen

Datentypen

Wertebereiche

Variablennamen

Verwendung

Deklaration und

Wertzuweisung

Ausgabe

Arithmetik

Modulo

In- bzw. Dekrement

Integer-Division

Verknüpfung von

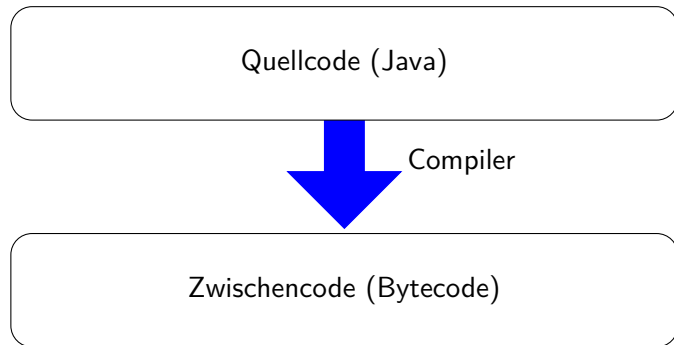
Zeichenketten

Konvertierung

Demonstration

NetBeans

(Debugging)



Ablauf

Technologie

Variablen

Datentypen

Wertebereiche

Variablennamen

Verwendung

Deklaration und

Wertzuweisung

Ausgabe

Arithmetik

Modulo

In- bzw. Dekrement

Integer-Division

Verknüpfung von

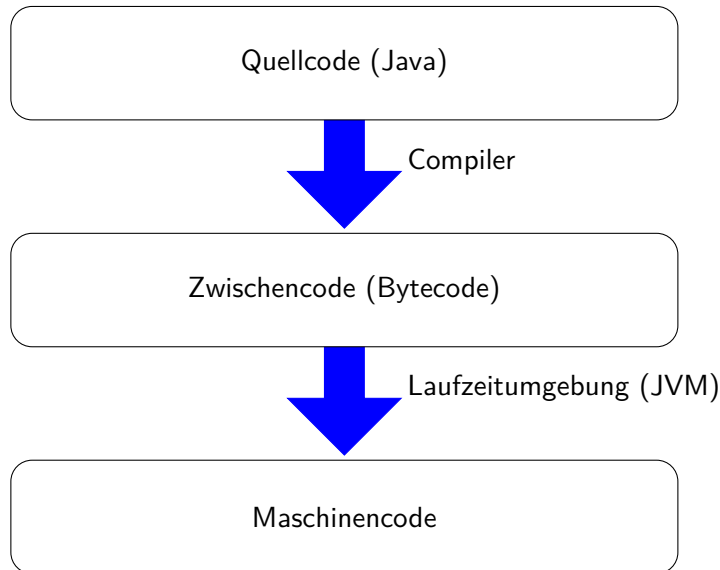
Zeichenketten

Konvertierung

Demonstration

NetBeans

(Debugging)



Ablauf

Technologie

Variablen

Datentypen

Wertebereiche

Variablennamen

Verwendung

Deklaration und

Wertzuweisung

Ausgabe

Arithmetik

Modulo

In- bzw. Dekrement

Integer-Division

Verknüpfung von

Zeichenketten

Konvertierung

Demonstration

NetBeans

(Debugging)

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

Ablauf

Technologie

Variablen

Datentypen

Wertebereiche

Variablennamen

Verwendung

Deklaration und

Wertzuweisung

Ausgabe

Arithmetik

Modulo

In- bzw. Dekrement

Integer-Division

Verknüpfung von

Zeichenketten

Konvertierung

Demonstration

NetBeans

(Debugging)

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

Ablauf

Technologie

Variablen

Datentypen

Wertebereiche

Variablennamen

Verwendung

Deklaration und
Wertzuweisung

Ausgabe

Arithmetik

Modulo

In- bzw. Dekrement

Integer-Division

Verknüpfung von
Zeichenketten

Konvertierung

Demonstration

NetBeans
(Debugging)

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

Ablauf

Technologie

Variablen

Datentypen

Wertebereiche

Variablennamen

Verwendung

Deklaration und
Wertzuweisung

Ausgabe

Arithmetik

Modulo

In- bzw. Dekrement

Integer-Division

Verknüpfung von
Zeichenketten

Konvertierung

Demonstration

NetBeans
(Debugging)

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	$\pm 1,4E - 45$ bis $\pm 3,4E + 38$
double	8	64	$\pm 4,9E - 324$ bis $\pm 1,7E + 308$

Ablauf

Technologie

Variablen

Datentypen

Wertebereiche

Variablennamen

Verwendung

Deklaration und

Wertzuweisung

Ausgabe

Arithmetik

Modulo

In- bzw. Dekrement

Integer-Division

Verknüpfung von

Zeichenketten

Konvertierung

Demonstration

NetBeans

(Debugging)

Ablauf

Technologie

Variablen

Datentypen

Wertebereiche

Variablennamen

Verwendung

Deklaration und
Wertzuweisung

Ausgabe

Arithmetik

Modulo

In- bzw. Dekrement

Integer-Division

Verknüpfung von
Zeichenketten

Konvertierung

Demonstration

NetBeans
(Debugging)

- ▶ 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**

- ▶ Konventionen
 - ▶ So SOLLTEN Namen sein, aber der Compiler würde es auch sonst kompilieren
 - ▶ sinnvolle, aussagekräftige Namen
 - ▶ keine Abkürzungen
 - ▶ Substantive
 - ▶ Nur lateinische Zeichen, Zahlen und _
 - ▶ KEIN ä, ö, ü, ß, ...
 - ▶ camelCase-Schreibweise

Ablauf

Technologie

Variablen

Datentypen

Wertebereiche

Variablennamen

Verwendung

Deklaration und
Wertzuweisung

Ausgabe

Arithmetik

Modulo

In- bzw. Dekrement

Integer-Division

Verknüpfung von
Zeichenketten

Konvertierung

Demonstration

NetBeans
(Debugging)

zur Verwendung

Ablauf

Technologie

Variablen

Datentypen

Wertebereiche

Variablennamen

Verwendung

Deklaration und
Wertzuweisung

Ausgabe

Arithmetik

Modulo

In- bzw. Dekrement

Integer-Division

Verknüpfung von
Zeichenketten

Konvertierung

Demonstration

NetBeans
(Debugging)

- ▶ Bekanntmachen der Variablen:

type name;

- ▶ Beispiele:

int age;

char gender;

boolean isStudent;

- ▶ Der Variablen einen Wert zuweisen:

```
name = wert;
```

- ▶ Die Variable muss deklariert worden sein

- ▶ Beispiele:

```
int age; age = 20;  
float balance; balance = 4.2f;  
char gender; gender = 'm';  
String name; name = "Douglas Adams";
```

- ▶ Wert direkt beim Deklarieren auch zuweisen:

type name = value;

- ▶ Beispiele:

double average = -5.2;

boolean isStudent = **true**;

String answer = "42";

- ▶ Sonst würde es nachher ziemlich langweilig

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

- ▶ Beispiele:

```
System.out.println("Hallo_Welt");
```

```
String name = "Welt";  
System.out.print("Hallo_");  
System.out.print(name);  
System.out.println();
```

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 + 2;
```

```
double division = 3.5 / (result - 1);
```

Ablauf

Technologie

Variablen

Datentypen

Wertebereiche

Variablennamen

Verwendung

Deklaration und

Wertzuweisung

Ausgabe

Arithmetik

Modulo

In- bzw. Dekrement

Integer-Division

Verknüpfung von

Zeichenketten

Konvertierung

Demonstration

NetBeans

(Debugging)

Modulo (Restwertberechnung)

- Das Ergebnis des Modulo ist der Rest der Division:

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

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

- ▶ Erhöht bzw. erniedrigt den Wert einer Variablen

```
int x = 42;
```

```
x++;
```

```
x--;
```

- ▶ Post- bzw. Präinkrement sind zu beachten

```
int x = 23;
```

```
System.out.println(++x);
```

```
int y = 23;
```

```
System.out.println(y++);
```

Integer-Division

```
int x = 7;
```

```
int y = 2;
```

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

Welchen Wert beinhaltet z?

```
int x = 7;
```

```
int y = 2;
```

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

Welchen Wert beinhaltet z? $z = 3.0$

Ablauf

Technologie

Variablen

Datentypen

Wertebereiche

Variablennamen

Verwendung

Deklaration und

Wertzuweisung

Ausgabe

Arithmetik

Modulo

In- bzw. Dekrement

Integer-Division

Verknüpfung von

Zeichenketten

Konvertierung

Demonstration

NetBeans

(Debugging)

- ▶ 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");
```

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

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

Welchen Wert beinhaltet y?

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

Welchen Wert beinhaltet b?

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

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

Welchen Wert beinhaltet y? $y = 42$

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

Welchen Wert beinhaltet b?

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

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

Welchen Wert beinhaltet y?

y = 42

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

Welchen Wert beinhaltet b?

b = 512

Einige Typen können ohne Probleme in andere umgewandelt werden

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

```
int x = 42;
```

```
float y = (float)x;
```

ist äquivalent zu:

```
int x = 42;
```

```
float y = x;
```

Zurück zum Divisionsproblem

```
int x = 7;
```

```
int y = 2;
```

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

- ▶ Bei Rechnungen wird in den bestmöglichen Typen gecastet

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

- ▶ So funktioniert es:

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

Ablauf

Technologie

Variablen

Datentypen

Wertebereiche

Variablennamen

Verwendung

Deklaration und

Wertzuweisung

Ausgabe

Arithmetik

Modulo

In- bzw. Dekrement

Integer-Division

Verknüpfung von

Zeichenketten

Konvertierung

**Demonstration
NetBeans
(Debugging)**

Demonstration NetBeans (Debugging)