

Tutorium zur Vorlesung Programmieren

2. Tutorium

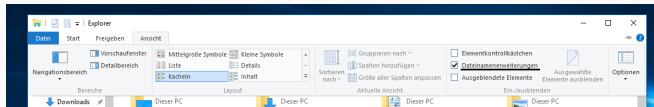
Joshua Gleitze

7. November 2016

IPD Koziolk

FEHLERQUELLEN

- Klasse Foo muss in die Datei Foo.java



- Beachtet genaue Schreibweise in der Ausgabe

Wiederholung

Acht elementare Datentypen:

Acht elementare Datentypen:

`boolean`

`char`

`byte`

`short`

`int`

`long`

`float`

`double`

Acht elementare Datentypen:

boolean Warheitswert: **true** oder **false**

char

byte

short

int

long

float

double

Acht elementare Datentypen:

boolean Warheitswert: **true** oder **false**

char Ein Buchstabe (UTF8)

byte

short

int

long

float

double

Acht elementare Datentypen:

boolean Wahrheitswert: **true** oder **false**

char Ein Buchstabe (UTF8)

byte 8-Bit-Ganzzahl

short

int

long

float

double

Acht elementare Datentypen:

boolean Wahrheitswert: **true** oder **false**

char Ein Buchstabe (UTF8)

byte 8-Bit-Ganzzahl

short 16-Bit-Ganzzahl

int

long

float

double

Acht elementare Datentypen:

boolean Wahrheitswert: **true** oder **false**

char Ein Buchstabe (UTF8)

byte 8-Bit-Ganzzahl

short 16-Bit-Ganzzahl

int 32-Bit-Ganzzahl

long

float

double

Acht elementare Datentypen:

boolean	Warheitswert: true oder false
char	Ein Buchstabe (UTF8)
byte	8-Bit-Ganzzahl
short	16-Bit-Ganzzahl
int	32-Bit-Ganzzahl
long	64-Bit-Ganzzahl
float	
double	

Acht elementare Datentypen:

boolean	Warheitswert: true oder false
char	Ein Buchstabe (UTF8)
byte	8-Bit-Ganzzahl
short	16-Bit-Ganzzahl
int	32-Bit-Ganzzahl
long	64-Bit-Ganzzahl
float	32-Bit-Fließkommazahl
double	

Acht elementare Datentypen:

boolean	Warheitswert: true oder false
char	Ein Buchstabe (UTF8)
byte	8-Bit-Ganzzahl
short	16-Bit-Ganzzahl
int	32-Bit-Ganzzahl
long	64-Bit-Ganzzahl
float	32-Bit-Fließkommazahl
double	64-Bit-Fließkommazahl

WEITERE DATENTYPEN

- String: `String message = "Hallo Welt";`
- Jede Klasse ist ein Typ!

VARIABLEN

Deklaration:

- `int x, y, z;`
- `String name;`

Zuweisung:

- `x = 5;`
- `name = "Annika";`

Initialisierung (Kombination von Deklaration und Zuweisung):

- `int x = 5;`
- `String name = "Annika";`

Variablennamen werden per Konvention in lowerCamelCase geschrieben:

- `firstName`
- `isValidUser`
- `preliminaryMaximallyTolerableTimeWithoutAdditionalTime`

ATTRIBUTE VS VARIABLEN

Was ist der Unterschied zwischen Attributen und Variablen?

Was ist der Unterschied zwischen Attributen und Variablen?

Attribut	Variable
<ul style="list-style-type: none">• Eigenschaft einer Klasse• Modelliert die Programmdaten• Selten: Mittel zum Zweck	<ul style="list-style-type: none">• Nur in der Methode sichtbar, in der sie deklariert ist• Reines Mittel zum Zweck

ENUM

Möglichkeit, eine feste Liste an Werten als Typ zu spezifizieren.

Season.java

```
1 enum Season {  
2     SPRING,  
3     SUMMER,  
4     AUTUMN,  
5     WINTER  
6 }
```

Jacket.java

```
1 class Jacket {  
2     String name;  
3     double price;  
4     Season forSeason;  
5 }
```

Shop.java (Auszug)

```
1 Jacket favouriteJacket = new Jacket();  
2 favouriteJacket.name = "Strellson MATT";  
3 favouriteJacket.price = 299.99;  
4 favouriteJacket.forSeason = Season.WINTER;
```

Quiz!

t1p.de/tut2

Vertiefung und Ausblick

TYPUMWANDLUNG ZU String

elementarer Datentyp x zu String: `String.valueOf(x)`

```
1 String message = "You have " + String.valueOf(messageCount) + "  
↪  new messages.";
```

oder implizit:

```
1 String message = "You have " + messageCount + " new messages.";
```

TYPUMWANDLUNG VON String

Strings können zu anderen Datentypen *geparsed* werden:

- **boolean** `y = Boolean.parseBoolean(x)`
- **char** `y = x.charAt(0)`
- **byte** `y = Byte.parseByte(x)`
- **short** `y = Short.parseShort(x)`
- **int** `y = Integer.parseInt(x)`
- **long** `y = Long.parseLong(x);`
- **float** `y = Float.parseFloat(x)`
- **double** `y = Double.parseDouble(x)`

TYPUMWANDLUNG VON String

Das funktioniert nur, falls die Eingabe das passende Format hat!

```
1 public class Age {  
2     public static void main(String[] args) {  
3         int age = Integer.parseInt(args[0]);  
4         System.out.println("You are " + age + " years old");  
5     }  
6 }
```

```
java Age 21
```


TYPUMWANDLUNG VON String

Das funktioniert nur, falls die Eingabe das passende Format hat!

```
1 public class Age {  
2     public static void main(String[] args) {  
3         int age = Integer.parseInt(args[0]);  
4         System.out.println("You are " + age + " years old");  
5     }  
6 }
```

```
java Age 21  
You are 21 years old
```

TYPUMWANDLUNG VON String

Das funktioniert nur, falls die Eingabe das passende Format hat!

```
1 public class Age {  
2     public static void main(String[] args) {  
3         int age = Integer.parseInt(args[0]);  
4         System.out.println("You are " + age + " years old");  
5     }  
6 }
```

```
java Age x
```

TYPUMWANDLUNG VON String

Das funktioniert nur, falls die Eingabe das passende Format hat!

```
1 public class Age {  
2     public static void main(String[] args) {  
3         int age = Integer.parseInt(args[0]);  
4         System.out.println("You are " + age + " years old");  
5     }  
6 }
```

```
java Age x  
Exception in thread "main" java.lang.NumberFormatException: For input string: "x"
```

KURZSCHLUSSAUSWERTUNG

```
1 int z = a / b == c;
```

KURZSCHLUSSAUSWERTUNG

1 `int z = a / b == c;`

1 `int z = b != 0 & a / b == c;`

KURZSCHLUSSAUSWERTUNG

1 `int z = a / b == c;`

1 `int z = b != 0 & a / b == c;`

1 `int z = b != 0 && a / b == c;`

KURZSCHLUSSAUSWERTUNG

```
1 boolean z = a / b == c;
```

```
1 boolean z = b != 0 & a / b == c;
```

```
1 boolean z = b != 0 && a / b == c;
```

Kurzschlussauswertung ist Bestandteil der Sprachspezifikation
⇒ verwenden!

&, | (fast) nie verwenden!

Übung

Gibt es Fragen zum Übungsblatt?

<https://judge.joshuagleitze.de>

- Aufgaben sind im DOMJudge hinterlegt
- Aufgaben nehmen in der Schwierigkeit zu
- Prüft eure Abgabe bevor ihr sie hochladet