



# PRIMITIVE DATENTYPEN



Christian Schirmer

# Primitive Datentypen

- Der Datentyp eines Attributes bestimmt, welche Informationen in einem Attribut abgelegt werden dürfen.
- Wird als Datentyp eines Attributes der Name einer Klasse angegeben, dürfen nur Objekte dieser Klasse als Werte diesem Attribut zugewiesen werden
- **Primitiven Datentypen** sind sehr einfache Datentypen, die nicht durch eine eigene Klasse beschrieben werden

# Primitive Datentypen

- Primitive Datentypen speichern
  - *ganze Zahlen (1, 12, 13131)*
  - *Fließkommazahlen (1.123, 21234.1232)*
  - *Wahrheitswerte (true, false)*
  - *einzelne Zeichen (t, w, f, d)*
- Primitive Datentypen sind einfache Standard-Datentypen, die es auch in anderen Programmiersprachen gibt

GANZE ZAHLEN



# Ganze Zahlen

Schlüsselwort	Größe	Wertebereich
byte	8 Bit (1 Byte)	-128 bis 127 $-2^7$ bis $2^7-1$
short	16 Bit (2 Byte)	-32.768 bis 32.767 $-2^{15}$ bis $2^{15}-1$
int	32 Bit (4 Byte)	-2.147.483.648 bis 2.147.483.647 $-2^{31}$ bis $2^{31}-1$
long	64 Bit (8 Byte)	-9.223.372.036.854.775.808 bis 9.223.372.036.854.775.807 $-2^{63}$ bis $2^{63}-1$

# Ganze Zahlen - **byte**

```
byte byte1 = 10;  
System.out.println("byte1: " + byte1);  
// byte1: 10
```

```
byte byteMin = Byte.MIN_VALUE;  
System.out.println("byteMin: " + byteMin);  
// byteMin: -128
```

```
byte byteMax = Byte.MAX_VALUE;  
System.out.println("byteMax: " + byteMax);  
// byteMax: 127
```

# Ganze Zahlen - **short**

```
short short1 = 10;  
System.out.println("short1: " + short1);  
// short1: 10
```

```
short shortMin = Short.MIN_VALUE;  
System.out.println("shortMin: " + shortMin);  
// shortMin: -32768
```

```
short shortMax = Short.MAX_VALUE;  
System.out.println("shortMax: " + shortMax);  
// shortMax: 32767
```

# Ganze Zahlen - `int`

```
int int1 = 30;  
System.out.println("int1: " + int1);  
// int1: 30
```

```
int intMin = Integer.MIN_VALUE;  
System.out.println("intMin: " + intMin);  
// intMin: -2147483648
```

```
int intMax = Integer.MAX_VALUE;  
System.out.println("intMax: " + intMax);  
// intMax: 2147483647
```



# Ganze Zahlen - long

```
long long1 = 10;
System.out.println("long1: " + long1);
// long1: 10

long long2 = 100000000000; //Kompilierfehler
// The literal 100000000000 of type int is out of range

long long3 = 100000000000L;
System.out.println("long3: " + long3);
// long3: 100000000000

long long4 = 100000000000L;
System.out.println("long4: " + long4);
// long4: 100000000000
```

# Ganze Zahlen - long

```
long longMin = Long.MIN_VALUE;  
System.out.println("longMin: " + longMin);  
// longMin: -9223372036854775808
```

```
long longMax = Long.MAX_VALUE;  
System.out.println("longMax: " + longMax);  
// longMax: 9223372036854775807
```

# GANZE ZAHLEN

Stellenwertsysteme



# Ganze Zahlen - Stellenwertsystem

- Ganzzahlige Literale: Angabe in vier Stellenwertsystemen
  - **Dezimalsystem** (Zehnersystem)  
*Literale bestehen aus den Ziffern »0« bis »9«*
  - **Binärsystem** (ab Java 7),  
*Literale bestehen aus den Ziffern »0« und »1«*
  - **Oktalsystem**  
*Literale bestehen aus den Ziffern »0« bis »7«*
  - **Hexadezimalsystem**  
*Literale bestehen aus den Ziffern »0« bis »9« und aus den Buchstaben »a«, »b«, »c«, »d«, »e« und »f«*

# Ganze Zahlen - Oktalsystem

Literale bestehen aus den Ziffern »0« bis »7

Ganze Zahlen beginnen mit dem Präfix 0

```
int zero = 00; // Entspricht decimal 0
System.out.println("Octal 00 = " + zero);
// Octal 00 = 0

int eins = 01; // Entspricht decimal 1
System.out.println("Octal 01 = " + eins);
// Octal 01 = 1

int sieben = 07; // Entspricht decimal 7
System.out.println("Octal 07 = " + sieben);
// Octal 07 = 7
```

# Ganze Zahlen - Oktalsystem

```
int eight = 010; // Entspricht decimal 8
System.out.println("Octal 010 = " + eight);
// Octal 010 = 8
```

```
int neun = 011; // Entspricht decimal 9
System.out.println("Octal 011 = " + neun);
// Octal 011 = 9
```

```
int dreizehn = 015; // Entspricht decimal 13
System.out.println("Octal 015 = " + dreizehn);
// Octal 015 = 13
```

```
int zwanzig = 024; // Entspricht decimal 20
System.out.println("Octal 024 = " + zwanzig);
// Octal 024 = 20
```

# Ganze Zahlen - Hexadezimalsystem

Hexadezimalsystem (Sedezimalsystem)

Literale bestehen aus:

*Ziffern »0« bis »9«*

*und*

*Buchstaben »a«, »b«, »c«, »d«, »e« und »f«*

*beziehungsweise »A« bis »F«*

Ein hexadezimaler Wert beginnt mit »0x« oder »0X«

# Ganze Zahlen - Hexadezimalsystem

```
int x = 0X0001;  
int y = 0x7fffffff;  
int z = 0xDeadCafe;
```

```
System.out.println("x = " + x + " y = " + y + " z = " + z);  
// x = 1 y = 2147483647 z = -559035650
```

```
long l1 = 110599L;  
long l2 = 0xFFFF1; // Note the lowercase 'l'  
long l3 = 0xFFFF;  
System.out.println("l1: " + l1 + " l2: " + l2 + " l3: " + l3);  
// l1: 110599 l2: 65535 l3: 65535
```



FLIEßKOMMAZAHLEN



# Fließkommazahlen

Schlüsselwort	Größe	Wertebereich
float	32 Bit (4 Byte)	$1,40239846 * 10^{-45}$ bis $3,40282347 * 10^{38}$
double	64 Bit (8 Byte)	$4,94065645841246544 * 10^{-324}$ bis $1,79769131486231570 * 10^{308}$

# Fließkommazahlen - float

```
float float1 = 1234.45f;
System.out.println("float1: " + float1);
// float1: 1234.45

float float2 = 1234.45; //Kompilierfehler
// Type mismatch: cannot convert from double to float
System.out.println("float2: " + float2);
// float2: 1234.45

float floatMin = Float.MIN_VALUE;
System.out.println("floatMin: " + floatMin);
// floatMin: 1.4E-45

float floatMax = Float.MAX_VALUE;
System.out.println("floatMax: " + floatMax);
// floatMax: 3.4028235E38
```

# Fließkommazahlen - double

```
double double1 = 1234.45;  
System.out.println("double1: " + double1);  
// double1: 1234.45
```

```
double double2 = 1234.45d;  
System.out.println("double2: " + double2);  
// double2: 1234.45
```

```
double double3 = 1234.45D;  
System.out.println("double3: " + double3);  
// double3: 1234.45
```

# Fließkommazahlen - double

```
double doubleMin = Double.MIN_VALUE;  
System.out.println("doubleMin: " + doubleMin);  
// doubleMin: 4.9E-324  
  
double doubleMax = Double.MAX_VALUE;  
System.out.println("doubleMax: " + doubleMax);  
// doubleMax: 1.7976931348623157E308
```

WAHRHEITSWERTE



# Wahrheitswerte

Schlüsselwort	Größe	Wertebereich
boolean	kann	true oder false

# Wahrheitswerte - **boolean**

```
boolean bool1 = true;
System.out.println("bool1: " + bool1);
// bool1: true

boolean bool2 = false;
System.out.println("bool2: " + bool2);
// bool2: false

boolean boolFalse = Boolean.FALSE;
System.out.println("boolFalse: " + boolFalse);
// boolFalse: false

boolean boolTrue = Boolean.TRUE;
System.out.println("boolTrue: " + boolTrue);
// boolTrue: true
```



# EINZELNE ZEICHEN

Characters



# Einzelne Zeichen

Schlüsselwort	Größe	Wertebereich
char	16 Bit (Unicode Zeichen – Kein Vorzeichen)	0 bis 65535  (Kann jeder zahl aus dem oberen Bereich zugewiesen werden)

# Einzelne Zeichen - **char**

```
char char1 = 'A';
System.out.println("char1: " + char1);
// char1: A

char charMin = Character.MIN_VALUE;
//Smallest value of type char '\u0000'
System.out.println("(int)charMin: " + (int)charMin);
// charMin:
// (int)charMin: 0

char charMax = Character.MAX_VALUE;
//Smallest value of type char '\uFFFF'
System.out.println("charMax: " + charMax);
System.out.println("(int)charMax: " + (int)charMax);
// charMax: ?
// (int)charMax: 65535
```

# ZEICHENKETTEN

String



# Zeichenketten - String

- Ein weiterer Datentyp ist String
  - *Speicherung von Zeichenketten*
- String ist kein primitiver Datentyp
  - *wird durch eine eigene Java-Klasse beschrieben*
- String kann wie ein primitiver Datentyp verwendet werden

# Zeichenketten - String

Schlüsselwort	Beschreibung
String	<p>Wird zum Speichern von beliebig langen Zeichenketten verwendet.</p> <p>Kein primitiver Datentyp, verhält sich aber oft so.</p> <p>Die Klasse String bietet viele Methoden für die Bearbeitung von Zeichenketten.</p>