



UNIVERSITÄT  
LEIPZIG

# Einführung in die Objekt-Orientierte Modellierung und Programmierung

Wintersemester 2025/2026

Dirk Zeckzer

Institut für Informatik



# Teil IV

## Klassen Datentypen

# Java: Klassen-Datentypen

primitiv	Klasse (java.lang.)	spezielle Werte
boolean	Boolean	TRUE, .FALSE
char	Character	BYTES
—	String	—

# Java: Klassen-Datentypen

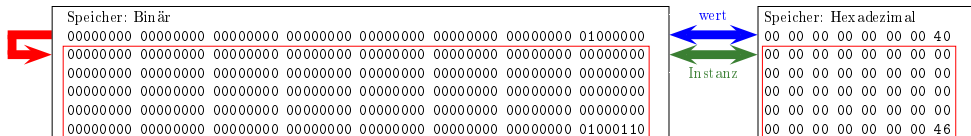
primitiv	Klasse (java.lang.)	spezielle Werte
byte	Byte	[ MIN_VALUE MAX_VALUE BYTES SIZE ]
short	Short	
int	Integer	
long	Long	
float	Float	[ wie Integer MIN_EXPONENT MAX_EXPONENT NaN NEGATIVE_INFINITY ]
double	Double	

# Java: Klassen-Datentypen

- ▶ Alle Klassen-Datentypen sind **immutable** (unveränderlich)
- ▶ → Werte können **nur gelesen** werden
- ▶ Änderung eines Wertes → neue Instanz

# Java: Speicherverwaltung

- ▶ Daten im Speicher im Binärformat
- ▶ Daten als Java Klassendatentyp
  - ▶ `Integer wert = new Integer(70);`
  - ▶ `Character wert = new Character('F');`
  - ▶ ...

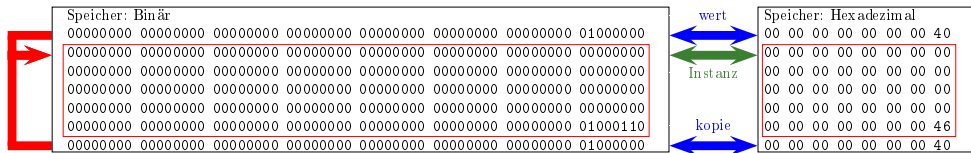


# Java: Speicherverwaltung

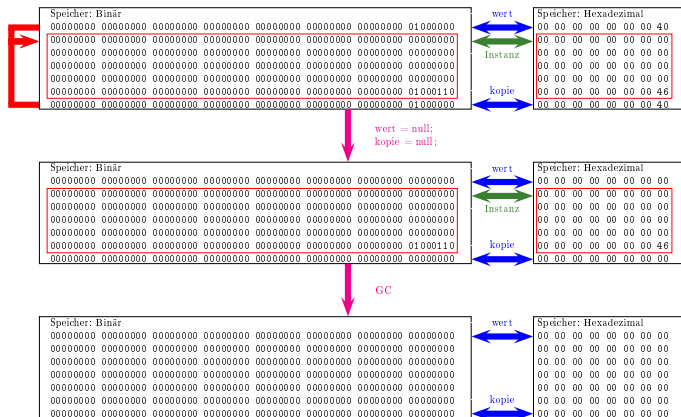
## ► Kopieren von Instanzen

► Integer wert = **new** Integer(70);

► Integer kopie = wert;



# Java: Speicherverwaltung



**Abbildung:** Garbage Collection (GC): Es gibt keinen Verweis mehr auf die Instanz (Variable, Attribut, Parameter) → Instanz wird vom Garbage Collector freigegeben



# Java Datentypen: Automatismen

Source-Code	Aktion	Bezeichnung
<code>Integer i = 5;</code>	<code>Integer i = <b>new</b> Integer(5);</code>	autoboxing
<code><b>int</b> j = i;</code>	<code><b>int</b> j = i.intValue();</code>	unboxing
<code><b>double</b> d = j;</code>	<code><b>double</b> d = (<b>double</b>) j;</code>	typecast
<code><b>long</b> l = i;</code>	<code><b>long</b> l = (<b>long</b>) i.intValue();</code>	unboxing & typecast
<code><b>int</b> k = d;</code>	Fehlermeldung	
<code>Long m = j;</code>	Fehlermeldung	

# Java: Klassen-Datentypen

## ► Nützliche Methoden

- `int i = Integer.parseInt(String s);`
- `Integer i = Integer.valueOf(String s);`

## ► Achtung:

- `Integer i = Integer.parseInt(String s);`  
→  
`Integer i = new Integer(Integer.parseInt(String s));`
- `int i = Integer.valueOf(String s);`  
→  
`int i = Integer.valueOf(String s).intValue();`

# Java: Klassen-Datentypen

## ► Weitere nützliche Methoden

► `int i = Integer.max(int a, int b);`

► `int i = Integer.min(int a, int b);`