



# Dungeon Crawl – Tag 5

29. November 2024

# Plan für die Woche

---

Montag

- Entscheidungsbäume
- Packages/Importe
- Abstrakte Klassen

Dienstag

- Logik
- Switch-case

Mittwoch

- Random
- Math Class

Donnerstag

- Scanner
- Rekursion von Methoden

Freitag

- Datentypen II
- Wrapper: Boolean, Double, Integer
- Casten

# Plan für heute

---

- Datentypen II
  - Wrapper: Boolean, Double, Integer
  - Casten

# Datentypen II

---

# Wrapper: Boolean, Double, Integer

---

# Wrapper-Klasse

---

- Spezielle Klassen, die primitive Datentypen umhüllen (wrap) und als Objekte behandeln
- **Autoboxing und Unboxing:** Automatisches konvertieren zwischen primitiven Datentypen und ihren Wrapper-Klassen
- Bieten zusätzliche Methoden



**Info:** In Java sind die primitiven Datentypen nicht Objekte, daher können sie nicht in Collection-Frameworks wie ArrayList verwendet werden, da diese nur Objekte speichern können. **Wrapper-Klassen ermöglichen es, primitive Datentypen als Objekte zu behandeln.**

# Wrapper-Klassen Übersicht

---

Primitive Datentypen	Wrapper-Klasse
byte	Byte
short	Short
int	Integer
long	Long
float	Float
double	Double
char	Char
boolean	Boolean

# Methoden der Wrapper-Klassen

---

- **parseInt(String s):** Wandelt eine String-Repräsentation eines primitiven Datentyps (z. B. int) in den entsprechenden Wert um.

```
int num = Integer.parseInt("123");
```

- **valueOf(String s):** Wandelt eine String-Repräsentation in das entsprechende Wrapper-Objekt um.

```
Integer num = Integer.valueOf("123");
```

- **toString():** Gibt den Wert des Wrapper-Objekts als String zurück.

```
Integer num = 10;  
System.out.println(num.toString());
```

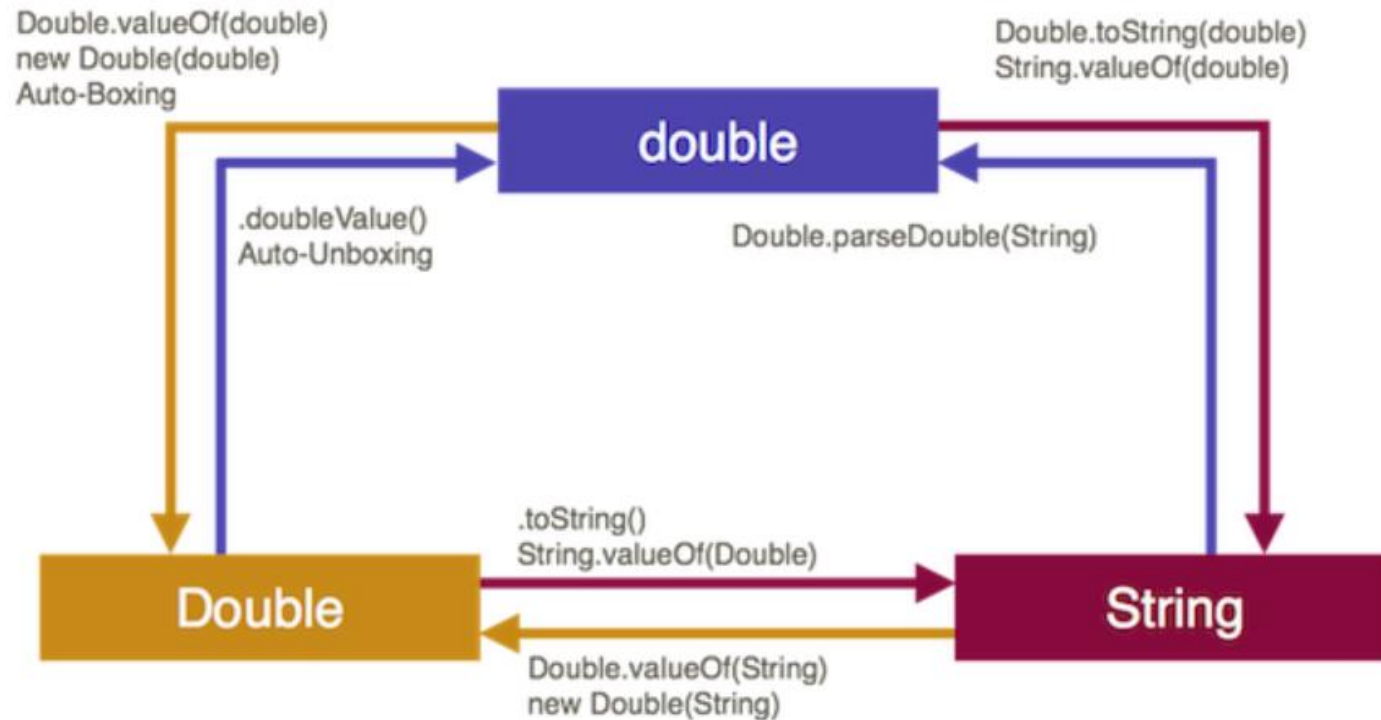
- **compareTo():** Vergleicht zwei Wrapper-Objekte.

```
Integer num1 = 10;  
Integer num2 = 20;  
System.out.println(num1.compareTo(num2));
```



# Methoden der Wrapper-Klassen

Beispiel:



# Konstanten in Wrapper-Klassen

---

- **MIN\_VALUE**: Kleinste Element des Wertebereichs
- **MAX\_VALUE**: Größte Element des Wertebereichs

Für die Wrapper-Klassen Float und Double zusätzlich:

- **NEGATIVE\_INFINITY**: Minus unendlich
- **POSITIVE\_INFINITY**: Plus unendlich
- **NaN** („Not a Number“): undefiniert



# Quiz

---

Was ist der Unterschied zwischen einem primitiven Datentyp und einer Wrapper-Klasse?

- a) Wrapper-Klassen sind keine Objekte und bieten keine zusätzlichen Funktionen.
- b) Primitive Datentypen sind Objekte und Wrapper-Klassen bieten zusätzliche Methoden.
- c) Primitive Datentypen sind einfache Werte, während Wrapper-Klassen Objekte sind, die diese Werte umhüllen und zusätzliche Methoden bieten.
- d) Es gibt keinen Unterschied, sie sind dasselbe.



# Quiz

---

Was bedeutet Autoboxing in Java?

- a) Das manuelle Umwandeln eines primitiven Datentyps in einen Wrapper-Typ.
- b) Der automatische Prozess, bei dem der Compiler einen Wrapper-Typ in einen primitiven Datentyp umwandelt.
- c) Der automatische Prozess, bei dem der Compiler einen primitiven Datentyp in einen Wrapper-Typ umwandelt.
- d) Ein Begriff, der sich auf die Verwendung von Klassen in einem "Auto"-Modus bezieht.



# Aufgabe

---

1. Schreibe ein Programm, das eine Zeichenkette "123" in eine Integer-Zahl umwandelt und diese Zahl dann als String zurückgibt.
2. Schreibe ein Programm, das zwei Integer-Objekte vergleicht. Verwende dazu die Methode `compareTo()` der Integer-Klasse und gib das Ergebnis aus.

# Casten

---

# Casten (Typumwandlung)

---

- Umwandlung von Datentypen in einen anderen
- Verschiedene Arten von Casting:
  - Implizites Casten
  - Explizites Casten

# Implizites Casten (widening casting)

---

- Daten eines kleineren Datentyps werden durch den Compiler **automatisch** dem größeren angepasst

byte -> short -> char -> int -> long -> float -> double

```
int num = 10;  
double doubleNum = num;
```



# Explizites Casten (narrowing casting)

---

- Wechsel von einem größeren in einen kleineren Datentyp
- Es ist eine explizite Anweisung erforderlich
- Es können dabei Daten verloren gehen!

double -> float -> long -> int -> char -> short -> byte

```
double doubleNum = 9.99;  
int intNum = (int) doubleNum;
```

# Casten von Objekten

---

- Umwandlung eines Objekts von einem Typ in einen anderen, der eine Oberklasse oder Unterklasse ist

## **Upcasting:**

Ein Objekt einer Unterklasse wird in ein Objekt der Oberklasse umgewandelt. Dies geschieht automatisch.

## **Downcasting:**

Ein Objekt einer Oberklasse wird in ein Objekt einer Unterklasse umgewandelt.



# Aufgabe

---

Schreibe ein Java-Programm, das eine Sammlung von unterschiedlichen Datentypen verarbeitet, mit denen du sowohl implizites als auch explizites Casting durchführst.

1. Erstelle für die primitiven Datentypen int, double, float, char, boolean je eine Variable mit einer passenden Initialisierung.
2. Konvertiere den int in einen long und den float in einen double.
3. Konvertiere den double in einen int und den long in einen short.



# Dungeon Crawl

---

# Quellen

---

<https://www.programmierenlernenhq.de/wrapper-klassen-in-java/>

<https://freiheit.f4.htw-berlin.de/prog2/wrapper/>

[https://users.informatik.uni-halle.de/~brass/oop14/long\\_beamer/jj\\_wrapp.pdf](https://users.informatik.uni-halle.de/~brass/oop14/long_beamer/jj_wrapp.pdf)

[http://dev.cs.ovgu.de/java/Books/javainsel3/javainsel\\_080001.htm](http://dev.cs.ovgu.de/java/Books/javainsel3/javainsel_080001.htm)

[https://www.w3schools.com/java/java\\_type\\_casting.asp](https://www.w3schools.com/java/java_type_casting.asp)