



Neue Themen für die 808

Februar 2025

Plan für die Woche

Montag

- Wiederholung Lambdas und Predicate
- ArrayLists
- ArrayLists und Lambdas

Dienstag

- Wrapper-Klassen
- Block an Quizfragen (15-20)

Mittwoch

- Java class structure
- Command line
- Selbständiges Vorbereiten zur Prüfung anhand Lernplans

Donnerstag

- Statische Variablen und Methoden
- Selbständiges Vorbereiten zur Prüfung anhand Lernplans

Freitag

- Features vergleichen
- Block an Quizfragen (15-20)

Plan für heute

- Wrapper-Klassen
- Interaktive Vorbereitung zur Prüfung

Wrapper-Klassen

BUCHSEITEN S.132-135

Grundlagen

- sind Klassen, die primitive Datentypen in Objekte verpacken („wrappen“)
- dadurch können primitive Datentypen auch für ArrayList genutzt werden

Wrapper-Klassen

Primitiver Datentype	Wrapper-Klasse
byte	Byte
short	Short
int	Integer
long	Long
float	Float
double	Double
char	Character
boolean	Boolean

Boolean-Konstruktor

```
Boolean booleanE2 = new Boolean(true);           //true
Boolean booleanE = new Boolean("TRUE");           //true
Boolean booleanE_1 = new Boolean("tRUe");         //true
Boolean booleanE_2 = new Boolean("false");        //false
Boolean booleanE_3 = new Boolean("FALSE");        //false
Boolean booleanE_4 = new Boolean("false2348");    //false
Boolean booleanE_5 = new Boolean("2348");         //false
```

Byte - Konstruktor

Muss davor geschrieben werden! Sonst Compilerfehler

```
Byte byteE = new Byte((byte)1);  
Byte byteE2 = new Byte("127");  
Byte byteE3 = new Byte("129"); //Exception: Wertebereich zu groß
```


Short - Konstruktor

Muss davor geschrieben werden! Sonst Compilerfehler

```
Short shortE = new Short((short)1);  
Short shortE2 = new Short("10");  
Short shortE3 = new Short("13058230583405"); //Exception
```

Integer und Long - Konstruktor

```
Integer integer = new Integer(10);  
Integer integer2 = new Integer("10");
```

```
Long longE = new Long(10);  
Long longE2 = new Long("10");
```

Float - Konstruktor

Doubles sind auch valide als Parameter

```
Float floatE = new Float(10.0);  
Float floatE2 = new Float(10.0F);  
Float floatE3 = new Float("10");  
Float floatE3_2 = new Float("10.0");  
Float floatE3_3 = new Float("10.0F");  
Float floatE3_4 = new Float("10.0D");
```

Double - Konstruktor

Floats sind auch valide als Parameter

```
Double doubleE = new Double(10.0);  
Double doubleE2 = new Double(10.0F);  
Double doubleE3 = new Double("10.0F");  
Double doubleE3_2 = new Double("10.0D");  
Double doubleE3_3 = new Double("10");  
Double doubleE3_4 = new Double("10.0");
```

Character - Konstruktor

```
Character c = new Character('a'); //muss ein char drinnen stehen
```



Mittels `Integer.intValue()` kriegt man den primitiven Wert der Wrapper-Klasse

Methoden

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));
```

Aufgabe

Schreibe ein Java-Programm, das einen Benutzer nach einer Zahl fragt, diese mit `Integer.parseInt()` umwandelt und diese dann mit sich selbst multipliziert.



Autoboxing und Unboxing

- Autoboxing bedeutet, dass Java automatisch primitive Datentypen in ihre Wrapper-Klassen umwandelt
- Unboxing ist das automatische auspacken des primitiven Werts aus einem Wrapper-Objekt

```
Integer a = 5; // Autoboxing  
int b = a + 10; // Unboxing und Berechnung  
System.out.println(b); // 15
```