

### Hüllklassen + Strings

- Hüllklassen hüllen einen Wert eines primitiven Datentyps in ein Klassenobjekt ein (d.h. dies ist ein nicht-statisches Attribut).
- Darüber hinaus besitzen Hüllklassen weitere Methoden + Attribute.

- Konstruktoren von Integer:

Integer x = new Integer(123);

Integer y = new Integer("123");

int z = Integer.parseInt("123");

String s1 = Integer.toString(123);

String s2 = x.toString();

erzeugt  
neues  
Integer-  
Objekt, das  
den Wert 123  
enthält.

s2  
s1 wird der  
String  
"123"

equals: inhaltliche Gleichheit

intValue: Selektor-Funktion, die gekapselten  
int-Wert zurückliefert

### Strings

Kurzform ( s = "Wort" )

und die Erzeugung von Strings  
mit Konstruktoren

(`m = new String("Wort")`)

sind fast äquivalent.

Aber: Java führt Tabelle mit Strings, die bereits in Kurzform erzeugt wurden und verwendet diese wieder, falls der gleiche String in Kurzform erzeugt wird.

⇒ `s = "Wort";`  
`t = "Wort";`

Danach zeigen `s` und `t` auf das gleiche Objekt.

Damit es keine Seiteneffekte aufgrund dieser Realisierung der Klasse `String` gibt, sind `String`-Objekte unveränderbar. Ein `String`-Objekt kann nach seiner Erzeugung nicht mehr verändert werden (immutable).

```
String s = "Wort";  
String t = "Wort";  
s = s + "e";
```

erzeugt ein neues  
String-Objekt "Worte"

Vordef. Methoden für  
Strings:

- `charAt` liefert Zeichen an  
best. Position (startet  
mit Pos. 0)
- `length` liefert Länge  
to Char Array überführt  
String in `char[]`.