

## II.2.5 Vordefinierte Klassen

Dienstag, 14. November 2017 08:30

Hüllklassen "hüllen" Wert eines prim.  
Datentyps ein.

• besitzen ein (nicht-statisches)  
Attribut des eingehüllten primitiven  
Typs (z.B. int)

- Integer.MAX\_VALUE ist  
int-Wert  $2^{31} - 1$ .

- Integer x = new Integer(123);

Integer y = new Integer("123");

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

String s = Integer.toString(i);

↑  
wird benutzt, wenn  
System.out.print für  
int-Werte aufge-  
rufen wird

String t = x.toString();

" wird benutzt, wenn  
System.out.println für  
Integer-Obj. aufgerufen  
wird.

-  $x == y$  ergibt false, denn  
x und y sind versch.  
Objekte  
 $x.equals(y)$

↑ ergibt true, wenn in  
x und y die gleiche  
int-Zahl gekapselt  
ist.

-  $x.intValue()$  liefert den  
gekapselten int-Wert  
zurück (getter).

## Strings

- ist ebenfalls vordef. Klasse

Erzeugung mit

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

- Es gibt aber auch Kurzform

- Es gibt aber auch Kurzform  
 $s = \text{"wort"}$

- Diese beiden Formen sind  
nicht ganz äquivalent.

Java speichert die Strings,  
die in Kurzform erzeugt wurden.  
Wird ein neuer String mit glei-  
chem Inhalt in Kurzform erzeugt,  
so wird der alte String wieder  
verwendet.

- Vordefinierte Methoden:

- equals

- int length()

- char[] toCharArray()

- char charAt(int x)

- String-Objekte sind  
unveränderbar (deshalb können

Objekte wiederverwendet werden):

$s = \text{"Wort"};$

$t = \text{"Wort"};$

$s = s + \text{"e"};$  ← Hierdurch  
entsteht  
ein neues  
String-Objekt.

D.h.  $t$  hat  
danach immer noch  
den Wert "Wort".