II.2.5 Vordefinierte Klassen

Mittwoch, 7. November 2018 09:30

Hüllklassen + Strings

Hüllklassen kapseln einen Wert des entspr. prim. Datentyps.

```
public class Integer {
private int value;
        :
}
```

Integer.MIN_VALUE ist der kleinste mögliche int-Wert (-2^{31}).

```
Integer x = Integer.valueOf(5);
Integer y = Integer.valueOf("5");
Integer u = Integer.valueOf(500);
Integer v = Integer.valueOf(500);
```

x == y ? true

u == v ? false ← denn

Integer-Objekte mit kleinen Zahlen werden wiederverwendet, aber mit großen Zahlen nicht

⇒ Klassenobjekte nicht mit == vergleichen, wenn man an inhaltlicher Gleichheit interessiert ist.

Stattdessen:

x.equals(y) liefert true

- Methoden zur Konversion in Strings
  - Integer.parseInt("123") erzeugt int-Wert 123
  - Integer.toString(123) erzeugt "123"
  - x.toString() erzeugt "5"

- X.intValue ist der gekapselte Wert, d.h. 5

Strings

String u = new String ("Wort");

Stattdessen Kurzschreibweise möglich:

String s = "Wort";

Strings, die in Kurzschreibweise erzeugt werden, werden gespeichert u. wiederverwendet.

String t = "Wort";

⇒ s == t ist true

s == u ist false

Um Strings auf inhaltliche Gleichheit zu untersuchen, sollte man equals verwenden.

```
S = S + "e";
```

Was ist dann der Wert von t? Immer noch "Wat"

String-Objekte sind unveränderbar (immutable) und bei "Wat" + "e" wird ein neues String-Objekt erzeugt.

Weitere Methoden:

- char charAt (int index)
  liefert das Zeichen an der entspr. Position des Strings (beginnend mit 0)
- int length ()
- char[] toCharArray ()
  überführt String in char[]