Konstruktoren und Vererbung

(A1) Ausprobieren

Experimentiere zunächst mit der Vorlage von https://codeberg.org/qg-info-unterricht/school-net-qg-v2. Die Vorlage enthält eine Version des NewsFeeds, in der Vererbung wie bisher besprochen implementiert ist.

Die Diagramme in der BlueJ-Oberfläche zeigen die Vererbungsbeziehungen an.

- Öffne den Quelltext der Klasse `TextBeitrag()` und entferne "extends Beitrag" in der Klassendefinition. Übersetze die Klasse erneut. Welche Veränderung beobachtest du am Klassendiagramm? Füge anschließend "extends Beitrag" wieder ein.
- Erzeuge ein TextBeitrags-Objekt. Rufe einige seiner Methoden auf. Kannst du auch die geerbten Methoden aufrufen, z.B `erfasseKommentar`? Was beobachtest du bei den geerbten Methoden?

Superklasse

Der Konstruktor der Superklasse initialisiert die in der Superklasse festgelegten Attribute:

```
public class Beitrag
{
    private String username;
    private long timestamp;
    private int likes;
    private ArrayList<String> comments;

    /**
     * Initialisiere die Felder von Beitrag
     */
    public Beitrag(String author)
    {
        username = author;
        timestamp = System.currentTimeMillis();
        likes = 0;
        comments = new ArrayList<>();
    }

    // weitere Methoden
}
```

Was macht der Konstruktor einer abgeleiteten Klasse?

Bei der Instanziierung eines Objekts einer abgeleiteten Klasse wird zunächst immer der Konstruktor der abgeleiteten Klasse aufgerufen.

Problem: Auch der `TextBeitrag` hat einen Autor – bei der Instanziierung eines TextBeitrags muss also das von der Superklasse geerbte Attribut `username` mit dem Parameter `author` initialisiert werden, aber wie?

```
public class TextBeitrag extends Beitrag
{
    private String message;

    /**
     * Konstruktor fuer TextBeitrag-Objekte
     */
    public TextBeitrag(String author, String text)
    {
        // PROBLEM: username muss irgendwie auf author
        // gesetzt werden?!

        // das ist klar.
        message = text;
    }

    // methods omitted
}
```

Lösung: Der Konstruktor der abgeleiteten Klasse ruft **immer** den Konstruktor der Superklasse auf: `super(…parameter…);` So werden alle geerbten Attribute initialisisert, wenn ein Objekt der abgeleiteten Klasse instanziiert wird.

Achtung: Wenn der super-Aufruf nicht explizit angegeben wird, wird super implizit ohne Argumente aufgerufen!

Zusammenfassung

- Die Konstruktoren abgeleiteter Klassen müssen **immer** einen Aufruf des Konstruktors der Superklasse (`super`) beinhalten.
- **Wenn der Programmierer keinen super-Aufruf in seinen Code einfügt, macht das der Compiler**. Dann wird super ohne Parameter aufgerufen – das geht schief, wenn der Konstruktor der Superklasse Parameter benötigt.
- Der `super`-Aufruf muss **das erste Statement** im Konstruktor der abgeleiteten Klasse sein.

Material

03_vererbung_konstruktor.odp	46.0 KiB	10.11.2021 19:41
03_vererbung_konstruktor.pdf	106.8 KiB	10.11.2021 19:41