

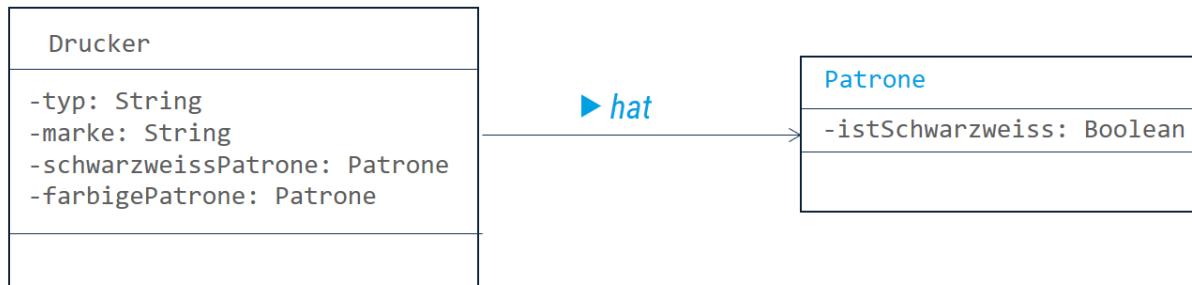
PROGRAMMIER- TECHNIK I ≥ ÜBUNG ≤

11

Die Besprechung der Aufgabenblätter 11 und 12 findet im Neuen Jahr am 13/14 Januar 2021 statt.

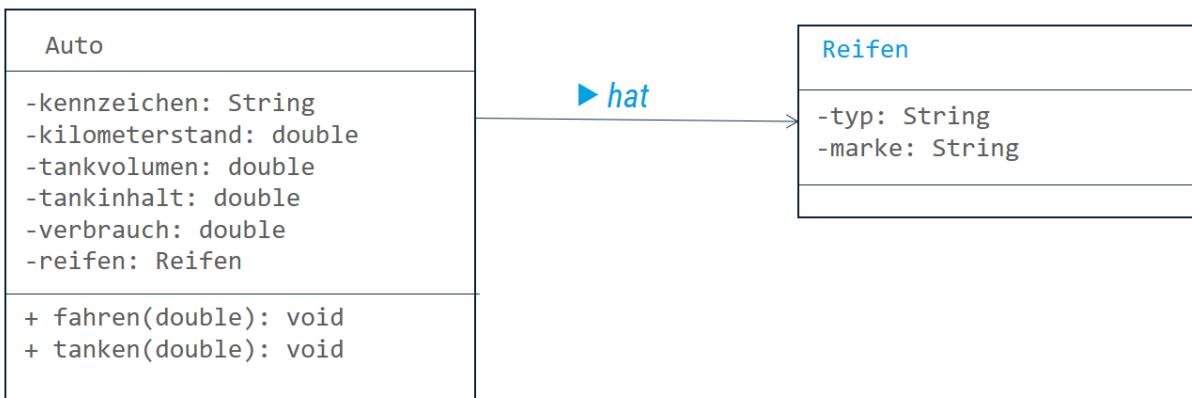
Aufgabe 0 (Wiederholung) – Holen Sie alle Übungen nach, die Sie während des Semesters nicht gelöst haben. Wiederholen Sie diejenigen Übungen, bei denen Sie sich noch nicht sicher fühlen.

Aufgabe 1 (Assoziationen) - Gegeben seien folgende Klassen in der UML-Darstellung.



- Implementieren Sie die beiden Klassen wie grafisch abgebildet.
- Stellen Sie zwei Konstruktoren in der Klasse Drucker bereit:
 - einen Konstruktor, der `typ` und `marke` des Drucker-Objekts initialisiert
 - einen Konstruktor, der alle Instanzvariablen initialisiert
- Stellen Sie `getter` und `setter` für die Instanzvariablen bereit.
- Stellen Sie einen Konstruktor für die Klasse Patrone bereit, der die Instanzvariable initialisiert sowie entsprechende `getter` und `setter`.

Aufgabe 2 (Assoziationen) - Gegeben seien folgende Klassen in der UML-Darstellung.



- Implementieren Sie beide Klassen wie grafisch abgebildet.
- Stellen Sie sinnvolle Konstruktoren in der Klasse Auto und Reifen bereit.
- Schreiben Sie eine Methode `fahren`. Als Parameter wird die Strecke, die zu fahren ist, übergeben. Überprüfen Sie ob der Tankinhalt für die Strecke ausreicht. Wenn ja, passen Sie die Instanzvariablen `kilometerstand` und `tankinhalt` an. Wenn nein, „fahren“ Sie die Strecke für die der Tankinhalt ausreicht. Passen Sie hierfür die Instanzvariablen entsprechend an.
- Schreiben Sie eine Methode `tanken`, die die zu tankende Menge übergeben bekommt. Stellen Sie sicher, dass nicht mehr getankt wird, als der Tankinhalt es zulässt.

PROGRAMMIER- TECHNIK I ≥ ÜBUNG ≤

11

Aufgabe 3 (UML, Assoziationen, Vererbung) – Erstellen Sie eine grafische Darstellung (UML) für die Aufgabe 9.4 (Hafen, Boote).

Aufgabe 4 (Copy-Konstruktor, flache/tiefe Kopie): Gegeben sei folgende Klasse in der UML-Darstellung.

Buch
- titel: String
- isbn: long
- preis: double

- Implementieren Sie die Klasse und stellen Sie mindestens einen Konstruktor, der als Parameter alle Instanzvariablen initialisiert, bereit. Stellen Sie für alle Instanzvariablen getter und setter bereit.
- Erstellen Sie eine Testklasse. Was würden folgende Zeilen bewirken? Wie viele Objekte wurden erzeugt?

```
01 Buch b1 = new Buch ("Der Heimweg", 3426281554L, 22.99);
02 Buch b2 = b1;
```

- Was ist die jeweilige Ausgabe in den nächsten Zeilen?

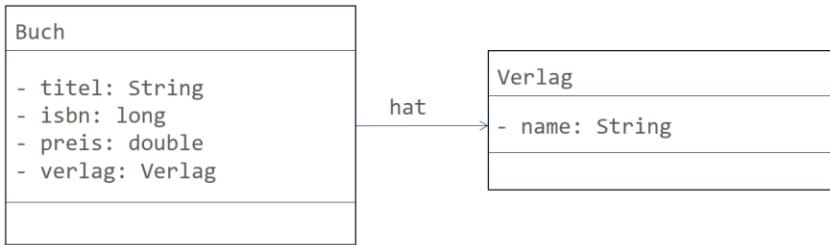
```
b1.setTitel("Die Rückkehr");
b2.setTitel("Der steinige Weg des Kopierens");
System.out.println (b1.getTitel());
System.out.println (b2.getTitel());
```
- Erstellen einen weiteren Konstruktor, der ein Objekt vom Typ Buch als Parameter bekommt. Wir nennen diesen Konstruktor den Copy-Konstruktor. In diesem Konstruktor wird das neue Objekt mit dem Inhalt der Instanzvariablen des übergebenen Objekts initialisiert.
- Ergänzen Sie die Testklasse um die dritte Zeile. Was bewirkt der Aufruf? Wie viele Objekte wurden erzeugt?

```
01 Buch b1 = new Buch ("Der Heimweg", 3426281554L, 22.99);
02 Buch b2 = b1;
03 Buch b3 = new Buch(b1);
```
- Was ist die jeweilige Ausgabe in den nächsten Zeilen?

```
b1.setTitel("Die Rückkehr");
b3.setTitel("Kopieren doch nicht so schwer?");
System.out.println (b1.getTitel());
System.out.println (b3.getTitel());
```
- Erstellen Sie eine weitere Klasse „Verlag“. Stellen Sie mindestens einen Konstruktor bereit, der alle Parameter initialisiert. Stellen Sie weiterhin getter und setter bereit.

PROGRAMMIER- TECHNIK I ≥ ÜBUNG ≤

11



- Erweitern Sie die Klasse Buch um die Instanzvariable verlag. Stellen Sie entsprechende getter und setter bereit und erweitern Sie den Konstruktor, der alle Instanzvariablen initialisiert, um die neue Instanzvariable. Erweitern Sie den Copy-Konstruktor derart, dass die aktuelle Instanzvariable mit dem Inhalt des zu kopierenden Objekts initialisiert wird.

```

public Buch(Buch b) {
    this.titel = b.getTitel();
    this.isbn = b.getIsbn();
    this.preis = b.getPreis();
    this.verlag = b.getVerlag();
}
  
```

- Angenommen wir erstellen eine zweite Testklasse, die folgendermaßen aussieht. Was passiert in den jeweiligen Zeilen? Warum?

```

public class BuchCopyKonstruktorTest2 {

    public static void main(String[] args) {

        Verlag v1 = new Verlag ("Droemer HC");
        Buch b1 = new Buch("Der Heimweg", 3426281554L, 22.99, v1);
        Buch b2 = new Buch(b1);
        System.out.println (b1.getTitel());
        System.out.println (b1.getVerlag().getName());
        System.out.println (b2.getTitel());
        System.out.println (b2.getVerlag().getName());

        b1.getVerlag().setName("Neuer Verlag");
        System.out.println (b1.getTitel());
        System.out.println (b1.getVerlag().getName());
        System.out.println (b2.getTitel());
        System.out.println (b2.getVerlag().getName());
    }
}
  
```

- Erweitern Sie Ihren Copy-Konstruktor derart, dass eine sogenannte „tiefe“ Kopie entsteht, d.h. es muss auch eine Kopie des Verlagsobjekts erzeugt werden. Wie gehen Sie vor?
- Tipps (nur schauen, wenn Sie nicht weiter kommen ;-):*
 - Erstellen Sie in der Klasse Verlag ebenfalls einen Copy-Konstruktor, wie wir ihn in der Klasse Buch erstellt haben.
 - Erweitern Sie den Buch-Copy-Konstruktor derart, dass Sie das Verlagsobjekt durch den Aufruf des Verlags-Copy-Konstruktors initialisieren.
- Wie verändert sich die Ausgabe in der zweiten Testklasse durch diese kleine aber feine Änderung?

PROGRAMMIER- TECHNIK I ≥ ÜBUNG ≤

11

Aufgabe 5: Erstellen Sie für die Klassen Drucker und Auto jeweils einen Copy-Konstruktor, der eine tiefe Kopie erzeugt. Gehen Sie wie in Aufgabe 4 beschrieben vor. Nehmen Sie an den Klassen Patrone und Reifen notwendige Änderungen vor.

Input zu I/O von der Konsole folgt ;-)