

Anmerkungen

- **Abgabe:** Die Abgabe erfolgt online auf ILIAS. Der Quellcode zu den Aufgaben 3-1 und 3-2 soll als *.java Datei und die Lösung von Aufgabe 3-3 soll als *.pdf Datei abgegeben werden. Andere Formate werden nicht akzeptiert.
- Quellcode-Dateien, welche wir nicht kompilieren können, werden nicht akzeptiert.
- Arbeit in Zweiergruppen: Geben Sie jeweils nur ein Exemplar der Lösung pro Gruppe ab. Geben Sie in der Quellcode-Datei die **Namen und Matrikelnummern** beider Gruppenmitglieder in den ersten beiden Zeilen als Kommentar an.
- Einzelarbeit: Geben Sie ebenfalls Ihren Namen und Ihre Matrikelnummer in der ersten Zeile der Quellcode-Datei als Kommentar an.

Aufgabe 3-1

Auf dem ILIAS-System (Übungen → Serie 3) finden Sie eine Datei `Book.java`. Ihre Aufgabe ist es die darin implementierte Klasse `Book` wie folgt zu erweitern:

1. Schreiben Sie mindestens einen Konstruktor sowie Get- und Set-Methoden zu allen Attributen.
2. Implementieren Sie die Methode `public int age()`, welche das Alter eines Buches (Anzahl Tage seit Erscheinungsdatum) berechnet und zurückgibt.
3. Implementieren Sie die Methode `public String toString()`, die alle Informationen eines `Book`-Objekts als String zurückgibt. Beispiel:
123, Die Blechtrommel, Günter Grass, 1.1.1959
4. Vervollständigen Sie die Methode `public void input()`, welche die Attribute `id`, `title`, `author` und `dateOfPublication` von der Kommandozeile einliest und im jeweiligen `Book`-Objekt abspeichert. Ungültige Eingaben müssen Sie nicht abfangen.

Hinweise:

- Zur Konvertierung zwischen `Date`- und `String`-Objekten verwenden Sie bitte die bereits in `Book.java` vorhandenen Methoden `dateToString` und `stringToDate`.
- Objekte der Klasse `java.util.Date` bestehen im Wesentlichen aus einem `long`-Wert, der die Anzahl Millisekunden seit 00:00:00 GMT am 1.1.1970 angibt. Verwenden Sie diesen Wert in Ihrer Implementation der Methode `age()`. Siehe auch Java API Dokumentation.
- Beachten Sie, dass die Anzahl Millisekunden eines Jahres den Wertebereich von `int` übersteigt.

Aufgabe 3-2

Schreiben Sie eine Klasse `Order` für Buchbestellungen. Ein `Order`-Objekt soll aus einer `id`, einem Kundennamen `customerName`, einer Kundenadresse `customerAddress` und maximal 5 `Book`-Objekten (**aus Aufgabe 1**) bestehen. Zudem soll die Klasse `Order` die Methoden `toString()` und `addBook(...)` enthalten.

Schreiben Sie zudem einen Konstruktor `Order()`, der die Instanzvariable `id` automatisch so initialisiert, dass das erste `Order`-Objekt die `id` 1, das zweite die `id` 2, das dritte die `id` 3 usw. erhält. Tipp: verwenden Sie eine `static`-Variable.

Verwenden Sie anschliessend die gegebene Klasse `Test` (ILIAS: Übungen → Serie 3) um Ihre Klasse `Order` zu testen. Die Ausgabe von `Test` soll **exakt** so aussehen:

```
$ java Test
Order id: 1, Customer: Sophie Muster, Mittelstrasse 10, 3011 Bern
1, Homo Faber, Max Frisch, 01.01.1957
2, Harry Potter, J.K. Rowling, 25.07.2000
3, Krieg und Frieden, Leo Tolstoi, 24.01.1867
4, Freedom, Jonathan Franzen, 08.06.2010
4, Freedom, Jonathan Franzen, 08.06.2010

Order id: 2, Customer: Woody Allen, 5th Avenue 7, 10001 New York
5, Goedel, Escher, Bach, Douglas Hofstadter, 05.11.1979
```

Hinweise:

- Die Klasse Order **muss** 5 Book-Instanzvariablen haben.
- Programmieren Sie nur get/set-Methoden, die tatsächlich verwendet werden.
- Die Klasse Test darf **nicht** verändert werden (auch nicht abgegeben).

Aufgabe 3-3

1. Welchen Output erzeugt das folgende Programm? Überlegen Sie sich die richtige Antwort ohne das Programm abzutippen und auszuführen.

```
public class Foo{
    private int a = 0;
    public static int b = 0;

    public void incrementA(){ a++; }
    public void incrementB(){ b++; }

    public String toString(){
        return "a="+a+", b="+b;
    }

    public static void main(String[] args){
        Foo f1 = new Foo();
        Foo f2 = new Foo();
        f2.incrementA();
        f2.incrementB();
        System.out.println(f1);
        System.out.println(f2);
    }
}
```