

Anmerkungen

- **Abgabe:** Quellcode zu 2-1 und 2-2 auf Papier, Lösung von 2-3 auf Papier, ausserdem Quellcode zu 2-1 und 2-2 über ILIAS.
- Jede Quellcode-Datei muss **Name(n) und Matrikelnummer(n)** enthalten.
- Zweiergruppen geben nur ein Exemplar ab. Notieren Sie aber beide Namen und Matrikelnummern.

Aufgabe 2-1

Sie sollen eine Klasse `RandomISBN` schreiben, mit deren Hilfe man zufällige (vereinfachte) ISBN (s. Wikipedia) erzeugen kann. Eine ISBN besteht aus vier Zahlen, die durch Bindestriche getrennt sind. Ein Beispiel einer ISBN ist 87-784-92-5. Aufgeteilt nach Ziffern hat eine ISBN also das Format

$$L_1L_2 - B_1B_2B_3 - V_1V_2 - C$$

wobei die zweistellige Zahl L_1L_2 für einen Ländercode, $B_1B_2B_3$ für die Titelnnummer, V_1V_2 für die Verlagsnummer und C für eine Prüfziffer stehen. Die Zahl $B_1B_2B_3$ soll ≥ 100 sein. L_1L_2 und V_1V_2 sollen grösser als 0 sein. Die Prüfziffer $0 \leq C \leq 9$ wird wie folgt berechnet:

$$C = L_1\#2 + L_2 + B_1\#2 + B_2 + B_3\#2 + V_1 + V_2\#2 \mod 10$$

wobei

$$i\#2 = \begin{cases} i * 2 - 9 & \text{falls } i * 2 \geq 10 \\ i * 2 & \text{sonst.} \end{cases}$$

Auf ILIAS finden Sie eine Klasse `RandomISBN`, die bereits eine `main`-Methode, eine Methode `makeISBN()` und eine Methode `hashOp` enthält. Die Methode `hashOp(int i)` berechnet die oben beschriebene Operation $i\#2$.

Ihre Aufgabe ist es die Methode `makeISBN()` zu vervollständigen. Die Zahlen L_1L_2 , $B_1B_2B_3$, V_1V_2 und C sollen darin zufällig generiert resp. berechnet (C) und in den vorgegebenen `String`-Variablen abgespeichert werden. Die Zahlen L_1L_2 und V_1V_2 sollen dabei **als Strings** der Länge 2 gespeichert werden. Die Zahl 3 entspricht z.B. dem String "03". Generieren Sie nicht für jede Ziffer einzeln, sondern blockweise für jede (mehrstellige) Zahl eine Zufallszahl.

Vervollständigen Sie anschliessend die `main`-Methode indem Sie die Methode `makeISBN()` dreimal aufrufen und das Ergebnis am Bildschirm ausgeben:

```
1st ISBN: 59-136-68-1
2nd ISBN: 46-704-30-0
3rd ISBN: 66-935-24-2
```

Hinweise:

- Sie können die Klasse `DecimalFormat` verwenden um Zahlen als Strings zu formatieren.
- Achtung: `int x = (int)'5'` liefert nicht 5 sondern den Code des Zeichens 5.
- Konsultieren Sie bei Bedarf die Java API Dokumentation unter <https://docs.oracle.com/javase/8/docs/api>.
- Die Methode `makeISBN()` soll keine Ausgabe auf der Kommandozeile erzeugen!

Aufgabe 2-2

Auf dem ILIAS-System (Übungsserien → Übungsserie 2) 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.
5. Schreiben Sie eine Klasse `BookTest`, die alle Methoden der `Book`-Klasse testet indem Sie ein `Book`-Objekt erstellt und jede Methode mindestens einmal ausführt.

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.
- Es ist möglich die Aufgabe ohne Verwendung von als *deprecated* markierten Methoden zu lösen.

Aufgabe 2-3

Welchen Output produziert das folgende Programm? Begründen Sie Ihre Antwort indem Sie zu jeder Zeile (ausser `System.out...`) der `main`-Methode kurz erklären was diese bewirkt.

```

1 public class A{
2     private int a = 3;
3     public void increment(){ a++; }
4     public String toString(){ return ""+a; }
5 }
6
7 public class B{
8     public static void main(String args[]){
9         A a1 = new A();
10        A a2 = new A();
11        a1.increment();
12        System.out.println(a1+"/"+a2);
13        a2 = a1;
14        a2.increment();
15        System.out.println(a1+"/"+a2);
16
17        String s1 = "ROCK";
18        String s2 = s1;
19        s2 = s2.toLowerCase();
20        System.out.println(s1+"/"+s2);
21
22        int j=1;
23        int i=j;
24        j++;
25        System.out.println(j+"/"+i);
26    }
27 }
```