

Aufgabenblatt A06:

1 Java Denksportaufgaben

Die Lernziele der Denksportaufgaben dieses Aufgabenblattes:

- Java API sorgfältig lesen.
 - Immer auf die aktuellen Versionen von Java achten.
 - Grundprinzipien üben.
1. Aus diesem Beispiel habe ich einiges gelernt und hoffe Sie werden auch viel daraus lernen!

Die folgende Methode soll nicht-negative ganze Zufallszahlen liefern:

```
import java.util.Random;
public class RandomNumbers {
    static Random rand = new Random();
    static int random(int n) {
        return Math.abs(rand.nextInt()) % n;
    }
}
```

Ich möchte von Ihnen wissen:

- Wie macht man das besser?
Die Frage ist aber viel zu einfach, ich möchte mehr von Ihnen haben:
- Welche Fehler sind hier gemacht worden? Ohne mathematische Details, Sie brauchen und sollen kein Buch über die Generierung von Pseudo-Zufallszahlen durchzuarbeiten!

- Wie macht man das richtig?
- Wie ist die aktuelle Empfehlung (seit ca. 10 Jahren), wie Sie das machen sollten?

Vielleicht hilft es bei der praktischen Analyse, wenn Sie es mit der folgenden main-Methode ausprobieren (z. B. in obiger Klasse):

```
public static void main(String[] args) {
    int n = 2 * (Integer.MAX_VALUE / 3);
    int low = 0;
    for (int i = 0; i < 1_000_000; i++) {
        if (random(n) < n / 2) {
            low++;
        }
    }
    System.out.println(low);
}
```

Das Ergebnis sollte Sie überraschen! Vielleicht hilft das ja bei der Ursachenanalyse.

2. Was fällt Ihnen an der folgenden Methode auf?

```
public static Comparator<Integer> naturalOrder(){
    return (i, j) ->
        (i < j) ? -1 : (i == j ? 0 : 1);
}
```

Ist sie korrekt? Wenn nicht, was ist falsch und wie macht man das richtig?

Hinweis: Sie sollten die Methode schon mal ausprobieren! Besser wäre noch: Systematisch testen, wenige Testfälle sollten ausreichen!

2 Lambdas und Streams

Die Lernziele dieser Aufgabe sind:

- Üben mit funktionalen Interfaces und Lambda-Ausdrücken.
- Mit unbegrenzten Streams umgehen können.
- Die Java-Klassenbibliothek besser kennen lernen.

1. Erzeugen Sie bitte einen (potentiell) unbegrenzten Stream, der die Jahre ab einem Startjahr liefert. Die Erzeugung soll beendet werden, wenn das erste binäre Palindrom-Jahr erzeugt wurde. Dieses geben Sie bitte auf der Konsole aus!

2. Der 13.03.2020 war ein Freitag. Schreiben Sie bitte einen unbegrenzten Stream, der die folgenden Freitage liefert, die auf den 13. eines Monats fallen! Sie müssen den Stream in irgendeiner Weise begrenzen. Verwenden Sie dafür bitte mehrere Möglichkeiten, nicht nur die Methode *limit*!
3. Verwenden Sie bitte auch Streams, um eine Text-Datei einzulesen und die Häufigkeiten der Worte zu ermitteln! Dies ist eine klassische map-reduce-Anwendung. Als Beispiele können Sie sich selber Beispiele suchen, ggf. kann ich einige zur Verfügung stellen.
4. Sie sollen eine Methode schreiben (z. B. eine Klassenmethode), die zwei Objekte einer Klasse nach einem Kriterium vergleicht, welches „besser“ der beiden ist. Die Strings werden als Parameter übergeben, ebenso das Kriterium.
 - 4.1. Definieren Sie bitte zu diesem Zweck ein geeignetes funktionales Interface.
 - 4.2. Können Sie das auch mit einem funktionalen Interface aus `java.util` erreichen? Welches Interface eignet sich ggf. am besten?
 - 4.3. Schreiben Sie bitte Methoden *anyPredicate* und *allPredicates*, die eine variable Anzahl von Predicates als Parameter erhalten und ein Predicate zurückliefern, dass einem bzw. allen der Predicates genügt!

Zeigen Sie bitte an Beispielen, dass Ihre Methoden für Predicates funktionieren!

5. Machen Sie (sich) bitte an eigenen Beispielen die Unterschiede zwischen `Stream<T>` mit `T = Integer`, oder `Double` und `IntStream` und `DoubleStream` deutlich!

Der Abgabetermin für alle ist:

Mittwoch, 13.05.2020, 08:00