

1. Klausur

Programmierpraktikum I
4. August 2025

Aufgabe 1

_____ / 4 Punkte

Zur Auswertung von Klausurergebnissen wollen wir getrennt nach bestandenen und nicht-bestandenen Klausuren wissen, wie die durchschnittliche Punktzahl ist. Eine Klausur ist bestanden, wenn mindestens 15 Punkte erzielt wurden.

Wie in folgendem Beispiel-Code zu sehen, liegen die Klausur-Ergebnissen als Liste vor:

```
record KlausurErgebnis(String matNr, int punkte) {}

public class KlausurStatistik {
    public static void main(String[] args) {
        List<KlausurErgebnis> ergebnisse = List.of(
            new KlausurErgebnis("123", 10),
            new KlausurErgebnis("234", 30),
            new KlausurErgebnis("345", 20),
            new KlausurErgebnis("456", 8));
        Map<Boolean, Double> stats = statistik(ergebnisse);
        System.out.print(stats.get(false)); // Ausgabe 9 [Berechnung: (8+10)/2]
        System.out.print(stats.get(true)); // Ausgabe 25 [Berechnung: (20+30)/2]
    }
}
```

Implementieren Sie die Methode `statistik`, sodass sie wie im Beispiel-Code gezeigt aufgerufen werden kann. Verwenden Sie **genau einen Stream-Ausdruck** und nutzen Sie immer **Methodenreferenzen**, wenn es geht.

Hinweis: In `Collectors` gibt es die Methode

```
public static <T> Collector<T,?,Double> averagingInt(ToIntFunction<? super T> mapper);
```

eine Variante ohne Parameter gibt es *nicht*.

```
public static Map<Boolean, Double> statistik(List<KlausurErgebnis> ergebnisse) {
    return
```

Aufgabe 2

_____ / 5 Punkte

Schreiben Sie eine **generische, statische Methode** mit dem Namen `map2`, die sich ähnlich verhält wie die higher order function, die wir als `map` kennengelernt haben.

Der Unterschied ist, dass wir bei `map2` nicht eine `Function`, sondern eine `BiFunction` übergeben bekommen. Außerdem bekommen wir zwei Instanzen einer Liste, während wir bei `map` nur eine Liste haben.

Die Methode soll die `BiFunction` auf Elemente aus beiden Listen anwenden (also zuerst auf die beiden ersten Elemente der Listen, dann auf die zweiten Elemente usw.) und das Ergebnis in einer neuen Liste speichern. Wir gehen davon aus, dass die beiden Listen immer dieselbe Anzahl von Elementen haben. `map2` kann wie folgt verwendet werden:

```
List<String> c1 = List.of("a", "b", "c");
List<Integer> c2 = List.of(1, 2, 3);
List<Double> c3 = List.of(2.5, 4.0, 9.0);
```

```
BiFunction<String, Integer, String> concat = (a,b) -> a + b;
List<String> r1 = map2(concat, c1, c2);
System.out.println(r1); // [a1, b2, c3]
```

```
BiFunction<Integer, Double, Double> multiply = (a,b) -> a * b;
List<Double> r2 = map2(multiply, c2, c3);
System.out.println(r2); // [2.5, 8.0, 27.0]
```

Hinweis: Die abstrakte Methode im Interface `BiFunction<A,B,C>` heißt `apply`.

Aufgabe 3

_____ / 5 Punkte

Wir setzen einen Druck-Service um, bei dem es möglich ist, mehrere Drucker zu registrieren. Ein Druck-Auftrag wird auf irgendeinem verfügbaren Drucker ausgeführt.

Fertig ist folgender Code:

```
interface Printer {
    boolean isAvailable();
    void print(String text);
}

class PrinterService {
    private final List<Printer> printers = new ArrayList<>();

    void addPrinter(Printer printer) {
        printers.add(printer);
    }

    void print(String text) {
        for (Printer printer : printers) {
            if (printer.isAvailable()) {
                printer.print(text);
                return;
            }
        }
        throw new IllegalStateException("No printer available");
    }
}
```

- ___/2 (a) Kreuzen Sie auf der nächsten Seite an, welche Aussagen über den Code korrekt sind; wenn die Korrektheit einer Aussage anhand des Codeausschnitts nicht entschieden werden kann, ist die Aussage *nicht* anzukreuzen. Nur bei angekreuzten Aussagen: Begründen Sie Ihre Antwort.
- ___/3 (b) Die `print`-Methode verstößt gegen das SLAP. Geben Sie einen Vorschlag in Form von konkretem Code an, wie das Problem behoben werden kann. Sie können dazu Änderungen direkt in die Code-Vorgabe schreiben.

Ankreuzen wenn korrekt: Code ...	Begründung (nur wenn angekreuzt)
<input type="checkbox"/> verletzt DIP.	
<input type="checkbox"/> enthält Refused Bequest.	
<input type="checkbox"/> verletzt Tell don't ask.	
<input type="checkbox"/> verletzt OCP.	

Aufgabe 4

_____ / 7 Punkte

Um die Verteilung von Studis auf Klausur-Hörsäle vorzunehmen, haben wir eine Software geschrieben. Folgender Code, der berechnen soll, welche der verfügbaren Hörsäle benutzt werden sollten, ist bereits fertig:

```
record Studi(int matNr) {}

record Hörsaal(String name, Set<String> plätze) {
    public Hörsaal(String name, Set<String> plätze) {
        this.name = name;
        this.plätze = Set.copyOf(plätze);
    }
}

class Aufteilung {
    static List<Hörsaal> benötigteHörsäle(List<Hörsaal> alleHörsäle, List<Studi> studis) {
        // Implementierung ist egal
    }
}
```

___/4

- (a) Schreiben Sie einen Test, der Folgendes überprüft: Wenn wir drei Hörsäle mit jeweils 3, 2 und 1 Plätzen haben und sich 4 Studierende angemeldet haben, werden die Hörsäle mit 3 und 1 Plätzen ausgewählt.

```
@Test
void test_1() {
```

- ___/3 (b) Sie haben sicher bemerkt, dass der Arrange-Schritt etwas unangenehm wird, besonders, wenn wir viele Tests haben. Schreiben Sie eine **statische Factory-Methode**, der wir den Namen eines Hörsaals und die Anzahl der Plätze übergeben und die uns dann ein Hörsaal-Objekt mit der passenden Anzahl von Plätzen liefert. Die Namen der Plätze können beliebig sein.

Aufgabe 5

____ / 6 Punkte

In einer Spring-Anwendung haben wir folgenden Code:

```
public record Data(String id, int p, int v) {}

public interface DataFetch {
    Data receive();
}

@Component
public class CsvFileDataFetch implements DataFetch {

    private String csvFileName;

    Data receive() {
        // Implementierung lädt aus der Datei csvFileName
        // Details sind egal
    }
}

@Component
public class Service {
    private final DataFetch dataFetch;

    public Service(DataFetch dataFetch) {
        this.dataFetch = dataFetch;
    }

    void process() {
        Data d = null;
        while ((d = dataFetch.receive()) != null) {
            if (d.p() > 0) {
                send(d);
            }
        }
    }

    private void send(Data d) {
        // Implementierung ist egal
    }
}
```

- ___/1 (a) Die Implementierung soll die Datei verwenden, die in der `application.properties` unter `data.csvFileName` steht. Ergänzen Sie den Code entsprechend.

___/1 (b) Leider bekommen wir immer eine `ArrayIndexOutOfBoundsException`-Exception, wenn die Methode `process` aufgerufen wird. Wir vermuten, dass einer der Datensätze ungültig ist. Schlagen Sie einen Weg vor, wie wir den fehlerhaften Datensatz identifizieren können, *ohne* den Code zu ändern. Die verarbeitete CSV-Datei enthält über 2 Millionen Datensätze, sodass es nicht sinnvoll ist, mit dem Debugger durch jede Schleifeniteration zu steppen.

___/4 (c) Nachdem wir den fehlerhaften Datensatz gefunden haben, stellen wir fest, dass die `ArrayIndexOutOfBoundsException`-Exception durch einen negativen Wert von `v` ausgelöst wird. Wir entschließen uns, die Methode `process` so zu ändern, dass empfangene Daten, die ungültig sind, ignoriert werden, und folglich keine Exception auslösen.

Schreiben Sie einen möglichst schnellen Test, der prüft, dass ein negativer Wert in `v` keine Exception erzeugt, wenn `process` aufgerufen wird:

```
@Test
void test_1() {
```

Aufgabe 6

_____ / 1 Punkt

Gegeben ist folgende Klasse:

```
class Dose {
    private int barcodeId;
    private String kategorie;
    private String name;

    public Dose(int barcodeId, String kategorie, String name) {
        this.barcodeId = barcodeId;
        this.kategorie = kategorie;
        this.name = name;
    }

    @Override
    public boolean equals(Object o) {
        if (!(o instanceof Dose dose)) return false;
        return barcodeId == dose.barcodeId
            && Objects.equals(kategorie, dose.kategorie)
            && Objects.equals(name, dose.name);
    }

    @Override
    public int hashCode() {
        return Objects.hash(barcodeId, kategorie, name);
    }
}
```

Ist hier Dose als Entity oder Value modelliert? Begründen Sie Ihre Antwort in 1–2 Sätzen.