HHU Düsseldorf
Institut für Informatik
Dr. J. Bendisposto
Dr. M. Brenneis

Daten des Prüflings

Matrikelnummer: _____

Nachname: _____

Vorname: _____

2. Klausur

Programmierpraktikum I
22. September 2025

- Prüfen Sie bitte zuerst, ob die Klausur alle Seiten enthält.
- Sie dürfen auf den leeren Rückseiten schreiben. Bei Bedarf erhalten Sie von uns leere Blätter. Sollten Sie auf diesen Blättern für die Korrektur relevante Teile Ihrer Lösung notieren, markieren Sie dies deutlich an der entsprechenden Aufgabe und auf dem zusätzlichen Blatt. Schreiben Sie auf alle zusätzlichen Blätter Ihren Namen. Geben Sie unten auf dem Deckblatt an, wie viele zusätzliche Blätter Sie zur Korrektur abgeben.
- Antworten dürfen auf Deutsch oder Englisch gegeben werden. Sofern keine ausformulierten Sätze verlangt sind, reichen nachvollziehbare Stichworte als Antwort.
- import-Statements müssen Sie nicht angeben.
- Erlaubte Hilfsmittel: eine beidseitig beschriebene A4-Seite, Wörterbuch (Letzteres muss vor Klausurbeginn der Aufsicht zur Kontrolle vorgelegt werden.)
- Schalten Sie technische Geräte aus. Täuschungsversuche führen zum sofortigen Ausschluss von der Klausur. Die Klausur wird dann als nicht bestanden gewertet.
- Schreiben Sie **nicht** mit radierbaren Stiften und auch **nicht** mit rot!

☐ Falls ich diese Klausur bestehe, möchte ich **nicht** automatisch zum Programmierpraktikum 2 im kommenden Wintersemester angemeldet werden.

Zusätzliche Blätter: _____ **Wir wünschen Ihnen viel Erfolg!**

Diesen Teil bitte nicht ausfüllen:

Aufgabe	1	2	3	4	5	6	Σ
Punkte	5	5	2	8	5	3	28
Erreicht							

Aufgabe 1

_____ / 5 Punkte

Wir wollen eine Methode `anyFileContainsString` schreiben, die eine Liste von Dateien und einen Suchbegriff übergeben bekommt. Die Methode soll genau dann `true` zurückgeben, wenn irgendeine der Dateien eine Zeile beinhaltet, die den Suchbegriff enthält.

Ihre Lösung muss aus genau einem Stream-Ausdruck bestehen. Benutzen Sie überall, wo es möglich ist, **Methodenreferenzen**.

Hinweise:

- Sie können die vorgegebene Methode `safeReadLines` verwenden, die die Text-Zeilen einer Datei zurückgibt (1 String = 1 Zeile).
- Die Klasse `File` besitzt eine Instanzmethode `Path toPath()`, die den Pfad eines File-Objekts zurückgibt.
- Die Klasse `String` enthält eine Instanzmethode `boolean contains(String)`, die prüft, ob der übergebene String ein Teilstring der `String`-Instanz ist.

```
public class Grep {
    private static Stream<String> safeReadLines(Path path) {
        // Implementierung egal
    }

    static boolean anyFileContainsString(List<File> files, String needle) {
        return
```

Aufgabe 2

_____ / 5 Punkte

Manchmal wollen wir eine Funktion über eine Liste mappen, beim Mappen aber auch Zugriff auf die Position des Elements (also den Index) haben. In dieser Aufgabe wollen wir dazu eine statische Methode `mapIndexed` schreiben, die wie folgt verwendet werden kann:

```
List<String> namen = List.of("Marlin", "Charlie");
List<String> namenMitPlaetzen = mapIndexed(namen, (i, name) -> i + ": " + name);
System.out.println(namenMitPlaetzen); // ["0: Marlin", "1: Charlie"]

List<Integer> zahlen = List.of(7, 5, 6);
List<Double> zahlenMitKomma = mapIndexed(zahlen, (i, z) -> z + i / 10.0);
System.out.println(zahlenMitKomma); // [7.0, 5.1, 6.2]
```

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

___/1 (a) Welchen konkreten Typ (inkl. konkrete Typparameter) hat der Lambda-Ausdruck `(i, name) -> i + ": " + name` im Beispielcode (zweite Zeile, Erzeugung der Variable `namenMitPlaetzen`)?

___/4 (b) Geben Sie eine Implementierung für `mapIndexed` an:

Aufgabe 3

_____ / 2 Punkte

OpenQL ist eine in der Programmiersprache C++ geschriebene Software, um Quanten-Schaltkreise zu simulieren. Ähnlich wie „normale“ Schaltkreise gibt es dort eine feste Menge verschiedener, grundlegender Gatter (X, Y, H, CNOT usw.), die unterschiedliche Funktionalitäten haben.

Wir wollen eine Java-Variante von OpenQL erstellen, die wie folgt verwendet werden kann:

```
Circuit k = new Circuit();
k.gate("x", 0);
k.gate("y", 1);
k.gate("cnot", 0, 1);
String s = k.toString();
// Bit 0: -X--o-
//            |
// Bit 1: -Y--+-
```

Bisher haben wir nur die Methoden `void gate(String name, int... bitNumber)` und `String toString()` implementiert. Als wir in die Original-Dokumentation der Methode `gate` gucken, werden wir etwas stutzig:

> Parameter name (String): The name of the gate. Note that OpenQL currently uses string comparisons with these names all over the place to derive functionality, and to derive what the actual arguments do. This is inherently a bad idea and something we want to move away from, so documenting it all would not be worthwhile. For now, just use common sense, and you'll probably be okay.

___/1 (a) In der Dokumentation steht, dass es eine schlechte Idee sei, dass die Gatter im gesamten Code über Strings identifiziert werden. Erklären Sie in 1–2 ausformulierten Sätzen, warum dies tatsächlich eine schlechte Idee ist.

___/1 (b) Wir wollen diese „schlechte Idee“ in unserer Java-Variante von OpenQL vermeiden. Was sollten wir in unserer Java-Variante anders machen? Antworten Sie in 1–2 ausformulierten Sätzen.

Klausurnr. 0 (0)

Aufgabe 4

_____ / 8 Punkte

Für das Programmierprojekt 1 (Aufteilen von Ausgaben) hat Milena u. a. folgenden Code geschrieben:

```
record Ausgabe(String name, int betrag) {}

class Ausgaben {
    private final List<Ausgabe> ausgaben = new LinkedList<>();

    Ausgaben(String... ausgaben) {
        ausgabenEinlesen(ausgaben);
    }

    private void ausgabenEinlesen(String[] ausgaben) {
        if(ausgaben.length % 2 != 0) {
            throw new IllegalArgumentException();
        }
        for(int i = 0; i < ausgaben.length; i += 2) {
            this.ausgaben.add(new Ausgabe(ausgaben[i],
                                          Integer.parseInt(ausgaben[i+1])));
        }
    }

    Collection<Ausgabe> asCollection() {
        return List.copyOf(ausgaben);
    }
}
```

___/3 (a) Schreiben Sie einen Test dafür, dass die Ausgaben der drei Personen A, B und C in Höhe von jeweils 100, 20 und 10 von der Methode `asCollection` korrekt zurückgeben werden. Achten Sie darauf, dass der Test nicht geändert werden muss, falls sich ind er Zukunft die Reihenfolge der Einträge im Ergebnis ändert.

```
@Test
void test() {
```

Klausurnr. 0 (0)

__/4 (b) Da sich die Art, wie ein `Ausgaben`-Objekt erzeugt wird, möglicherweise in Zukunft noch ändert, wollen wir für den Test-Code einen Builder schreiben, der ein `Ausgaben`-Objekt konstruieren kann. Mit einer Methode `add(String, int)` sollen einzelne Ausgaben hinzugefügt werden. Geben Sie eine mögliche Implementierung für den Builder an.

Hinweis: Wenn Sie eine `List<Integer> zahlen` haben, können Sie mit `zahlen.toArray(Integer[]::new)` ein Array erhalten, das dieselben Elemente wie die Liste beinhaltet.

__/1 (c) Geben Sie einen Beispiel-Aufruf Ihres Builders an, der das gleiche `Ausgaben`-Objekt zurückgibt, das für Aufgabenteil (a) verlangt wurde.

Klausurnr. 0 (0)

Aufgabe 5

_____ / 5 Punkte

Wir entwickeln gerade Klassen für ein Bestell-System. Die Klasse `Order` soll Bestellungen von Produkten verwalten:

```
public class Order {
    private final long id;
    private final List<String> items = new ArrayList<>();
    private final List<Integer> prices = new ArrayList<>();

    public Order(long id) {
        this.id = id;
    }

    public synchronized void addItem(String name, int price) {
        if (name == null || price <= 0.0) {
            throw new IllegalArgumentException();
        }
        items.add(name);
        prices.add(price);
    }

    public synchronized double totalPrice() {
        return prices.stream().mapToDouble(i -> i).sum();
    }

    @Override
    public boolean equals(Object o) {
        if (!(o instanceof Order order)) return false;
        return id == order.id;
    }

    @Override
    public int hashCode() {
        return Objects.hashCode(id);
    }
}
```

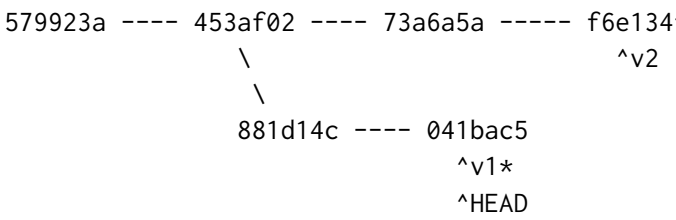
___/1 (a) Der Code hat den Data Clump Smell, weil Preis und Item eigentlich paarweise zusammengehören. Geben Sie ein Beispiel für eine mögliche, zukünftige Code-Änderung an, bei der wir wegen des Smells einfacher versehentlich etwas falsch machen könnten..

Aufgabe 6

_____ / 3 Punkte

Wir entwickeln eine Software zur Verwaltung von Konferenzen. Die aktuelle Version der Software ist Version v2. Wir haben allerdings auch noch Kundschaft, die die ältere Version v1 benutzt und Fehlerkorrekturen von uns erhält.

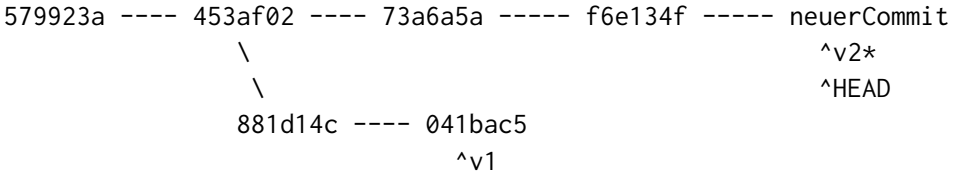
Der Code der Software wird in einem git-Repository verwaltet, wo es für jede Softwareversion einen eigenen Branch gibt:



Mit dem Commit `041bac5` haben wir einen Fehler in Version v1 behoben.

Wir haben festgestellt, dass der Fehler auch in Version v2 vorliegt. Die notwendigen Codeänderungen sind genau dieselben.

___/2 (a) Geben Sie an, mit welchem git-Befehl bzw. welcher git-Befehlfolge wir auf dem Branch v2 einen neuen Commit erzeugen können, der die gleichen Änderungen wie der Commit `041bac5` beinhaltet. Der Commit-Baum soll nach dem Ausführen der Befehle so aussehen:



___/1 (b) Können Sie sagen, welche Commit-ID der neue Commit hat? Begründen Sie Ihre Antwort.