

Übungen zur Vorlesung
Objektorientierte Programmierung: Wintersemester 2021/2022

Nr. 12, Abgabe bis 14.02.2022

Aufgabe 12.1: Alyx

6 Punkte

Verwenden Sie Lambda-Ausdrücke zum Lösen der folgenden Aufgaben in einer Klasse Lambda:

- a) Schreiben Sie eine Methode `public static void removeIf (ArrayList<T> list, Predicate<T> filter)`, die mithilfe der Methode `public boolean removeIf (Predicate<? super E> filter)` der Klasse `ArrayList` Elemente aus der Liste `list` entfernt. 1
- b) Schreiben Sie eine Methode `public static <T> void sortBy (ArrayList<T> list, Comparator<T> comparator)`, die eine übergebene Liste `list` mit `public void sort (Comparator<? super E> c)` der Klasse `ArrayList` sortiert. 1
- c) Schreiben Sie eine Methode `public static <T> String listToString (ArrayList<T> list)`, die eine übergebene Liste unter Verwendung der Methode `public void forEach (Consumer<? super E> action)` in einen `String` umwandelt. Stellen Sie sicher, dass jedes Element eine eigene Zeile erhält. 2
Hinweis: Verwenden Sie `StringBuilder`, um Ihren `String` zu erstellen.
- d) Schreiben Sie für jede der oben genannten Methoden einen geeigneten JUnit-Test. 2
Verwenden Sie dabei `ArrayList` und einen beliebigen Typen. Achten Sie darauf, dass Ihre Liste mindestens 10 Elemente aufweist.

Aufgabe 12.2: Azarath Metrion Zinthos!

6 Punkte

In dieser Aufgabe sollen Sie eine Textdatei mithilfe der Streams-API einlesen und analysieren. Die Klasse `BufferedReader` verfügt hierfür über die Methode `Stream<String> lines()`, die einen Eingabestrom als zeilenweisen Stream bereitstellen kann. Erstellen Sie die Klasse `IOStreams` mit den folgenden Methoden (geben Sie, wenn möglich, etwaige Exceptions an die aufrufende Methode weiter:

- a) **public static** `Stream<String> words(String filePath)`, die eine Datei mithilfe von `BufferedReader` einliest und in einen Stream aus Wörtern umwandelt. Achten Sie bei Ihrer Implementierung darauf, dass Sie lediglich auf Methoden aus `String`, `Arrays`, `Stream` und (natürlich) `BufferedReader` zurückgreifen. 1
- b) **public static** `Map<String, Integer> countWords(Stream<String> words)`, die die Vorkommen eines Wortes in einem Stream zählt und diese in einer Map zurückgibt. Verwenden Sie bei Ihrer Implementierung lediglich Methoden aus dem `Collectors` Interface. 2
- c) **public static void** `writeCountedWords(Map<String, Integer> countedWords, String filepath)`, die eine neue Datei unter `filepath` (der Name der Datei ist in `filepath` inkludiert) anlegt und in diese den Inhalt der übergebenen Map schreibt. Achten Sie darauf, dass für jeden Eintrag in der Map eine neue Zeile geschrieben werden soll. 2

Hinweis: Sie können sich im Ilias die Datei `countedWords.txt` als Schablone anschauen.

- d) Testen Sie Ihre Methoden mit JUnit. Sie können für Ihre Tests die Datei `theRaven.txt` im Ilias verwenden. Die Datei `countedWords.txt` enthält die gewünschte Lösung für gezählte Wörter in `theRaven.txt`. 1