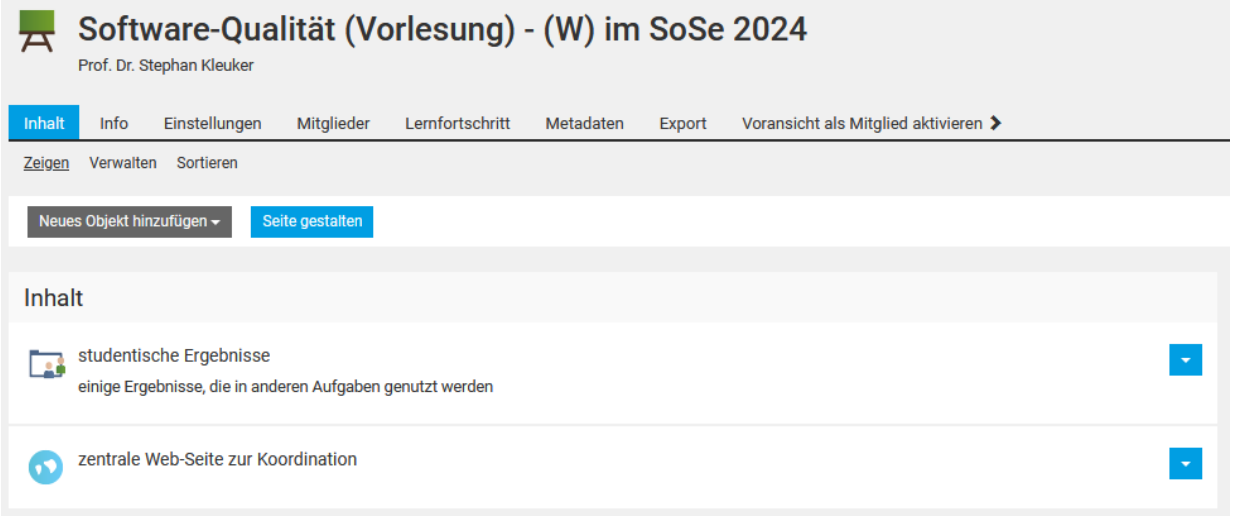
Prof. Dr. Stephan Kleuker
Hochschule Osnabrück
Fakultät Ing.-Wissenschaften und Informatik
- Software-Entwicklung -

Software-Qualität

Sommersemester 2024
3. Aufgabenblatt

Aufgabe 0.3 (0 Punkte)
Geben Sie das Lösungswort des Quiz aus der Lernnotiz an.

Aufgabe 5 (4 Punkte)
Diese Aufgabe setzt die vorherige Aufgabe 4 fort. Schreiben Sie Tests für alle Ihre Methoden aus Aufgabe 4, Testen Sie dabei positive, wie negative Ergebnisse. Machen Sie sich vorher auf Papier Skizzen, welche Möglichkeiten es gibt, wie Intervalle zueinander liegen können. Im Namen der Testklasse sollen Ihre Nachnamen vorkommen. Sie sollten auf 20-35 sinnvolle Tests insgesamt kommen.
Kopieren Sie Ihre Testklasse nach ILIAS in das Verzeichnis „studentische Ergebnisse“, das unter der Vorlesung angelegt ist.



Aufgabe 6 (5 Punkte)
Studierende wurden aufgefordert, eigene Sortieralgorithmen zu implementieren. Um diese flexibel vergleichbar zu machen, wurde ein Interface definiert, dass von den eigenen Sortierklassen realisiert werden soll. Das Interface sieht wie folgt aus und kann generisch beliebige Klassen T sortieren, die das Interface Comparable<T> realisieren.

```
package hilfsklassen;
import java.util.List;
public interface MeinSortierer <T extends Comparable<T>>{
	public List<T> sortieren(List<T> list);
}
```

Die Methode sortieren erhält eine Liste übergeben und soll eine sortierte Liste mit den gleichen Objekten zurückgeben, die absteigend sortiert sind. Eine Klasse ohne Implementierung muss damit wie folgt aussehen.

```
package hilfsklassen;

import java.util.ArrayList;
import java.util.List;

public class Sortierer1<T extends Comparable<T>>
                                    implements MeinSortierer<T> {

	public List<T> sortieren(List<T> list) {
		// hier sortieren
	}
}
```

Von der Webseite der Lehrveranstaltung können im Projekt qsAufgabeSorteier acht Beispielimplementierungen des Interfaces in einem Projekt geladen werden. Mit der Klasse

MainSortierer zeigen die Studierenden vermeintlich, dass zumindest einige ihrer Ideen funktionieren.

a) Überlegen Sie sich zunächst genau, wie man formalisieren kann, dass eine Liste mit beliebigen Elementen, die geordnet werden können, durch eine Methode geordnet wird. Schreiben Sie diese Kriterien als Text auf.
b) Überlegen Sie sich anschaulich verschiedene Szenarien, also anschauliche Äquivalenzklassen, die eine Überprüfung des Sortierverhaltens einer Liste erlauben. Um die Aufgabe einzuschränken, sollten Sie mindestens fünf Szenarien benennen.
c) Formulieren Sie die in b) gefundenen Testfälle für die verschiedenen Sortierer als JUnit-Testfälle. Welche Sortierer scheitern an welchen Testfällen?
d) [ohne Punkte, freiwillig] Die meisten der fehlerhaften Sortierer lassen sich recht einfach korrigieren. Führen Sie diese Korrekturen, z. B. in einem eigenen Eclipse-Projekt, durch und testen Sie diese Lösungen.