

Software Engineering 2 - Klausur SoSe 07.09.2021 (100 Punkte)

Zum Teil Gedächtnisprotokoll

Aufgabe 1 (20 Punkte)

- 1) Bestandteile einer Serviceorientierten Architektur nennen und wie diese miteinander interagieren
- 2) Sechs Unterschiede zwischen SOAP und REST nennen
- 3) Model Checking Idee erklären und zu traditionellen Testen abgrenzen
- 4) Unterschiede zwischen Parallelen und Nebenläufigen Prozessen erklären

Aufgabe 2 (15 Punkte)

XSD Datei erstellen für

- Film
 - Titel
 - Erscheinungsdatum
 - Mindestens 1 und maximal 2 Regisseure
 - Diese haben Vor- und Nachnamen
 - ID mit positivem Integer

Aufgabe 3 (15 Punkte)

- 1) Program Dependence Graph mit Kontroll- und Datenabhängigkeiten zu einem vorgegebenen Programm anfertigen
- 2) Forward und Backward Slicing zu jeweils einer Programmzeile vom gleichen Programm durchführen

Aufgabe 4 (30 Punkte)

Gegeben ist folgendes WHILE-Programm

```
r = 1;
x = n;
while (x > 0) {
    r = r * x;
    x = x - 1;
}
```

- 1) Kontrollflussdiagramm angeben, die Kanten mit den Anweisungen und die Knoten mit der Programmzeile annotieren
- 2) Partielle Korrektheit mit der Pfadvariante beweisen
- 3) Vollständige Korrektheit beweisen

Aufgabe 5 (20 Punkte)

Das folgende Java-Programm modelliert ein soziales Netzwerk

```
public class SocialNetwork {

    static class User {
        private final String name;
        private final HashSet<User> friends;

        public User(String name) {
            this.name = name;
            this.friends = new HashSet<User>();
        }

        public synchronized void friend(User that) {
            if (friends.add(that)) {
                that.friend(this);
            }
        }
    }

    public static void main(String[] args) {
        final User mark = new User("mark");
        final User eduardo = new User("eduardo");

        new Thread(new Runnable() {
            public void run() { mark.friend(eduardo); }
        }).start();
        new Thread(new Runnable() {
            public void run() { eduardo.friend(mark); }
        }).start();
    }
}
```

Wie Facebook ist dieses soziale Netzwerk bidirektional: Wenn Benutzer A mit Benutzer B befreundet ist, dann ist B auch mit A befreundet. Die Methode friend erhält diese Invariante aufrecht, indem sie die Menge der Freunde in beiden Objekten verändert.

- 1) Was für ein Problem könnte auftreten und wie kommt es zustande?
- 2) Das Programm anpassen, ohne die Invariante zu verändern