

SoSy Braindump WS 21/22 (02.04.2022)

Dass an den Aufgaben keine Punkte stehen ist Absicht – in der Klausur waren auch keine gegeben

Stand: 6. April 2022 19:49

A1 Theorie

Makieren Sie bei folgenden Aussagen jeweils ob der *kursiv geschriebene Begriff* richtig (✓) oder falsch (✗) ist. Geben Sie zusätzlich bei falschen Aussagen den richtigen Begriff an.

- a) Dokumentationsänderungen sind dem *Perfective Maintenance* zuzuordnen.
Ist diese Aussage richtig?:
Falls nein, ist der korrekte Begriff:
- b) Ein *Fehler* ist die Abweichung zwischen realisiertem Produkt und dem beabsichtigten Verhalten.
Ist diese Aussage richtig?:
Falls nein, ist der korrekte Begriff:
- c) *Sequentielle* Kohäsion ist eine Operation auf gemeinsamen Daten, bei der die Reihenfolge irrelevant ist.
Ist diese Aussage richtig?:
Falls nein, ist der korrekte Begriff:
- d) Jedes Kommunikationsdiagramm lässt sich in ein äquivalentes *Sequenzdiagramm* umwandeln.
Ist diese Aussage richtig?:
Falls nein, ist der korrekte Begriff:
- e) Mit *Datenkopplung* ist gemeint, dass eine Datenstruktur von Modul A mit Modul B geteilt wird aber B nur Teile der Daten daraus verwendet.
Ist diese Aussage richtig?:
Falls nein, ist der korrekte Begriff:
- f) Das Entwurfsmuster *Beobachter* wird verwendet um die Anzahl von Instanzen einer Klasse zu limitieren.
Ist diese Aussage richtig?:
Falls nein, ist der korrekte Begriff:
- g) *Verifikation* ist das Abgleichen des Produkts mit der Anwendungsspezifikation.
Ist diese Aussage richtig?:
Falls nein, ist der korrekte Begriff:

A2 Anforderungsanalyse

Ein Pfandautomat soll Flaschen erkennen und basierend auf der erkannten Flasche Geld zurückgeben. Dabei wird nach folgenden Kriterien unterschieden:

Typ: Mehrweg oder Einweg

Material: Glas oder Kunststoff

Größe: 0,5L oder 1L

Es sind folgende Anforderungen gegeben:

A1: Wird eine Einwegflasche erkannt gibt der Automat 0,25ct zurück.

A2: Wird eine 1L-Mehrwegflasche erkannt gibt der Automat 0,75ct zurück.

A3: Wird eine 0,5L-Glasflasche erkannt gibt der Automat 0,50ct zurück.

A4: Sonst gibt der Automat 0ct zurück.

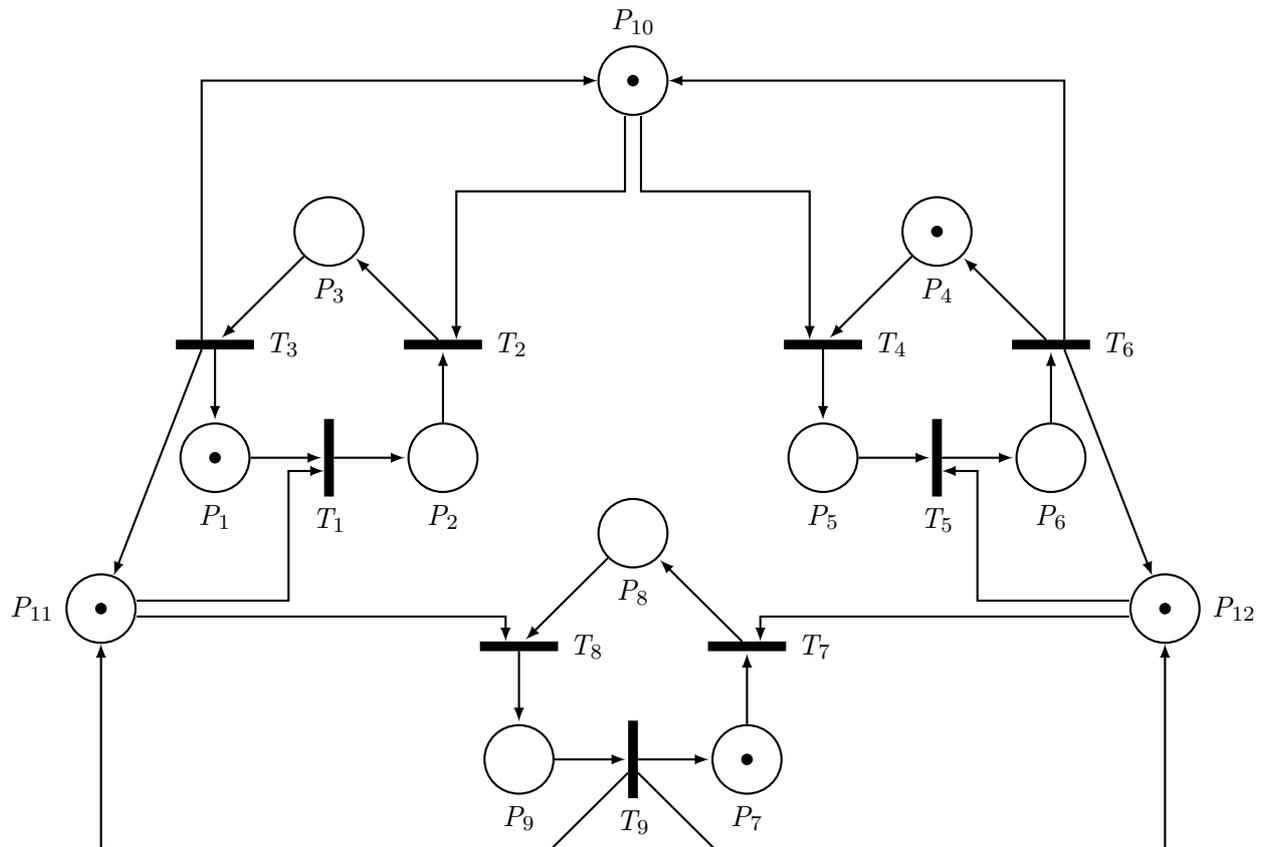
Entscheiden Sie für folgende Anforderungsmengen, ob sie vollständig, konsistent bzw. machbar sind und kennzeichnen Sie dies jeweils mit einem J bzw. N.

Geben Sie bei N zusätzlich eine Begründung mit der jeweiligen Nummer des Feldes an.

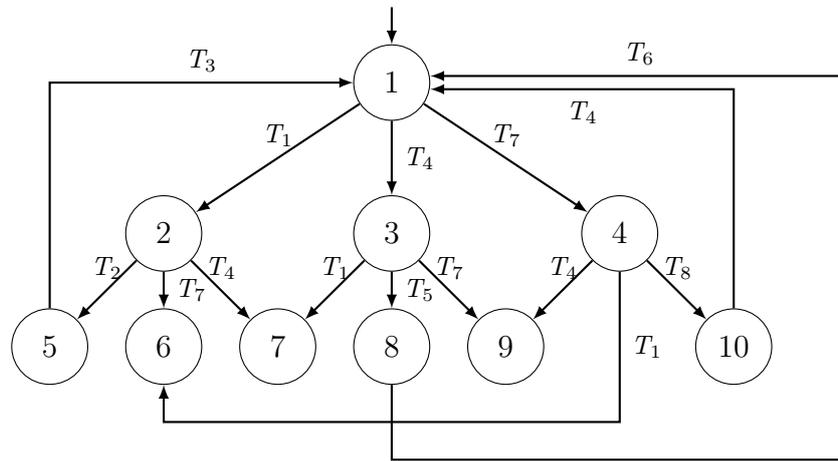
Menge	Vollständigkeit	Konsistenz	Machbarkeit
{A1, A2}	1	2	3
{A1, A2, A3}	4	5	6
{A1, A2, A4}	7	8	9
{A1, A2, A3, A4}	10	11	12

A3 Petri-Netz

Gegeben sei folgendes Petri-Netz:



- Vervollständigen Sie den unvollständigen Zustandsgraphen (nächste Seite).
Hinweis: Um zwölfelementige Vektoren zu vermeiden sind die Zustände (es sind nur die belegten Stellen angegeben) durchnummeriert.
- Untersuchen Sie den vervollständigten Graphen auf Deadlockmakierungen und kennzeichnen Sie diese.



Vorgegebene Zustände:

1: {P1, P4, P7, P10, P11, P12}

6: {P2, P4, P8, P10}

2: {P2, P4, P7, P10, P12}

7: {P2, P5, P7, P12}

3: {P1, P5, P7, P11, P12}

8: {P1, P6, P7, P11}

4: {P1, P4, P8, P10, P11}

9: {P1, P5, P8, P11}

5: {P3, P4, P7, P12}

10: {P1, P4, P9, P10}

Zusätzliche Zustände:

TODO: UML-Diagram 1

A5 Refactoring

Gegeben sei folgender Code:

```
class Index {
    public int value;
}

public class ArrayIterator {
    private String type;
    private Object[] array;
    private Index currentIndex;

    public boolean gotoNextElement() {
        // Protocol
        System.out.println("Start_method_" + "gotoNextElement()");
        System.out.println("Timing:_" + System.nanoTime());

        switch (type) {
            case "ascending":
                if (isNextElementAvailable()) {
                    currentIndex.value = currentIndex.value + 1;
                    return true;
                }
                return false;
            case "descending":
                if (isNextElementAvailable()) {
                    currentIndex.value = currentIndex.value - 1;
                    return true;
                }
                return false;
            default:
                throw new RuntimeException("Unknown_type");
        }
    }

    public boolean isNextElementAvailable() {
        // Protocol
        System.out.println("Start_method_" + "isNextElementAvailable()");
        System.out.println("Timing:_" + System.nanoTime());

        switch (type) {
            case "descending":
                return currentIndex.value > 0;
            case "ascending":
                return currentIndex.value < array.length - 1;
            default:
                throw new RuntimeException("Unknown_type");
        }
    }
}
```

a) Welche drei Refactoring-Techniken lassen sich hier sinnvoll anwenden? Geben Sie zu jeder Technik auch ihre Anwendung an.

1. Technik:
Anwendung:

2. Technik:
Anwendung:

3. Technik:
Anwendung:

b) Zeichnen Sie das UML-Diagramm nach Anwendung des Refactorings

A6 OCL

Gegeben sei folgendes UML-Diagramm:

TODO: UML-Diagramm 2

Formulieren Sie folgende Aussagen in OCL:

- a) Ein Reiseleiter muss Teilnehmer der Reise, die er leitet, sein.

- b) An einer Reise dürfen maximal so viele Personen teilnehmen wie ihr Bus Plätze hat.

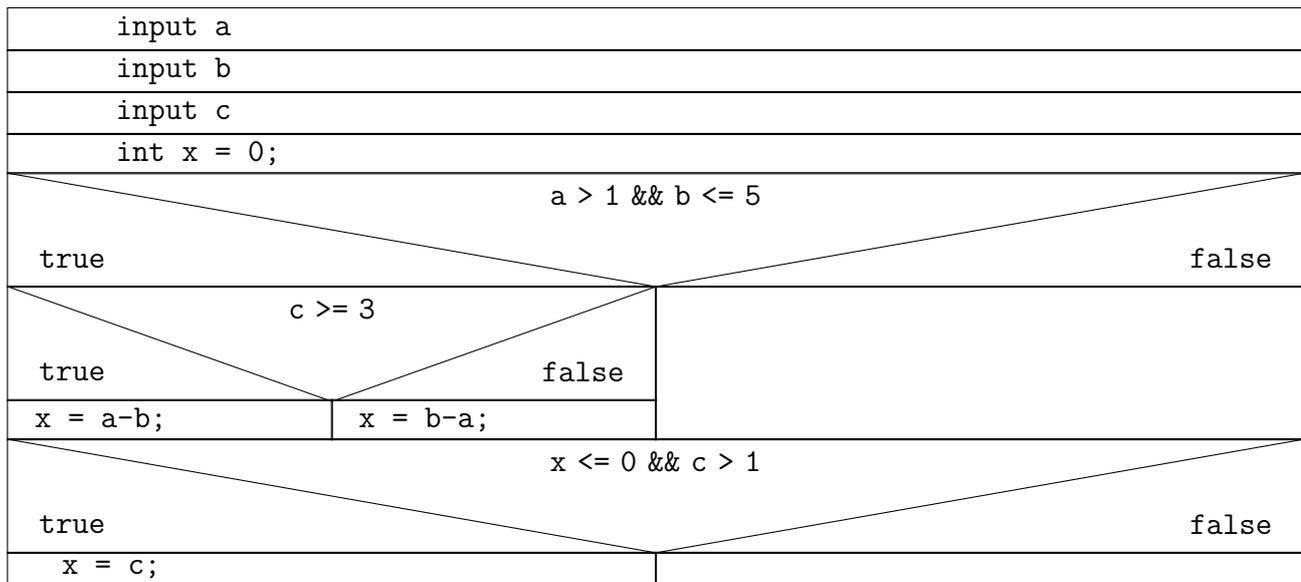
- c) Bei `addPerson(p: Person)` muss das Hinzufügen einer Person mindestens 7 Tage vor Abfahrt erfolgen. Danach müssen alle Mitglieder die davor teilgenommen haben sowie die Person `p` in der Liste der Teilnehmer sein.

- d) Registrationsnummern von Bussen sind eindeutig.

- e) Ein Bus darf nicht für sich überschneidende Reisen verwendet werden.

A7 Testen

Gegeben sei folgendes Strukturdiagramm:



Eingaben seien Tupel der Form (i, j, k) mit $i, j, k \in \{0, 1, 2, 3, 4\}$. Geben Sie minimale Mengen an für

a) Anweisungsüberdeckung:

b) Zweigüberdeckung:

c) Pfadüberdeckung:

d) Handelt es sich bei diesen Tests um White-Box oder Black-Box Tests? Begründen Sie ihre Antwort.