

# Gedächtnisprotokoll

## Software Engineering 1 – Jun. SoSe 2021

### Aufgabe 1: Unified Process (10 Punkte)

- a) Nennen sie die fünf Phasen des Unified Process und ihre Ergebnisse
- b) Nennen sie die Core-Workflows und ihre Ergebnisse
- c) Warum ist der Unified Process: Use-Case Driven  
Architekturzentriert  
Inkrementell
- d)

### Aufgabe 2: Singleton-Pattern (18 Punkte)

gegeben Waren Codefragmente eines Singleton-Patterns welche:

- Klassendeklaration
- instance-Attribut
- Singleton()-Konstruktor
- getInstance()

beschrieben, es war jeweils das richtige anzukreuzen.

So enthielt z.B. getInstance() bei *einem* Ankreuzfeld return this; anstatt return this.instance;

### Aufgabe 3: Kompositum-Pattern

Aufgabe war es Kapitel (z.B. eines Buches) als das Composite und Inhalte eines Kapitels als Leaves (gegeben waren Text, Tabelle, Abbildung, Bild) zu implementieren.

Hierbei war die anzahlBilder()-Methode zu beachten, welche bei Text/Tabelle 0 und bei Abbildung/Bild 1 zurückgeben sollte.

- a) UML-Diagramm zeichnen
- b) Die wichtigen Java-Code stellen aufschreiben

### Aufgabe 4: Aktivitätsdiagramm

Eine Studentin möchte die Software-Engineering 1 Klausur schreiben. An der Eingangskontrolle übergibt sie ihren Studentenausweis. Die Eingangskontrolle überprüft das Lichtbild, die Gültigkeit des Ausweises und ob ihr Name auf der Liste steht, trifft eines nicht zu, wird sie gebeten die Universität zu Verlassen und darf die Klausur nicht mitschreiben. Ist alles in Ordnung, überreicht die Studentin ihr EzP-Formular welches von der Eingangskontrolle überprüft wird, fehlen einige Angaben wird die Studentin von der Einlasskontrolle gebeten diese nachzutragen. Ist das EzP-Formular fertig ausgefüllt, bekommt die Studentin von der Einlasskontrolle ihre Platznummer, begibt sich zum Platz und darf die Klausur mitschreiben.

- a) zeichne ein Aktivitätsdiagramm.
  - parallelisiere soviele Vorgänge wie möglich!
  - verwende nur Konzepte aus der Vorlesung!

#### Aufgabe 4: Zustandsdiagramm

Gegeben war ein Snackautomat, zuerst wählt der Kunde eine Reihe des Snackautomaten aus. Ist die Reihe nicht existent, geht der Automat in den Startzustand zurück. Anschließend wählt der Kunde ein Produkt aus, ist das Produkt nicht existent (fehlerhafte Eingabe) oder Leer, geht der Automat in den Startzustand zurück. Nachdem die Reihe und das Produkt gewählt sind, öffnet der Automat die Geldeingabe. Nachdem der Kunde das Geld eingeworfen hat wird das Rückgeld ausgegeben. Nachdem das Rückgeld entnommen wurde, gibt der Automat das gewünschte Produkt aus und geht in den Startzustand zurück.

a) Zeichne ein Zustandsdiagramm

#### Aufgabe 5: Testen

Der genaue Wortlaut ist mir (dem Autor) nicht mehr geläufig, daher ist das Codefragment einem Gedächtnisprotokoll einer vorangehenden Klausur (SS20) entnommen.

Die Aufgabe war allerdings vom Inhalt her sehr ähnlich, lediglich die Werte und ihre Bezeichnungen weichen ab.

a) 1. Zeichne den Überdeckungsgraphen für folgenden Code:

```
public int berechneDiscount(int base, int points, int discount) {  
    if(base > 600 && points > 50) {  
        discount = discount * 5;  
    }  
    discount *= 1;  
    if(base > 1200 && discount > 100) {  
        discount = discount*2;  
    }  
    return discount;  
}
```

b)

2. Bestimme die Mindestanzahl an Testfällen für eine Anweisungsüberdeckung und gib diese an.
3. Bestimme die Mindestanzahl an Testfällen für eine Zweigüberdeckung und gib diese an.

c) Ein Sensor übergibt in Zentimetern den Wasserstand. Liegt dieser unterhalb von 1,5m leuchtet eine Lampe grün, liegt sie zwischen 1,5m und 2,5m leuchtet sie gelb und über 2,5m rot. Ungültige Eingaben müssen nicht berücksichtigt werden.

1. Bestimme die Äquivalenzklassen
2. Führe eine Grenzwertanalyse durch.
3. Was ist ein Grenzwert? Nenne drei Beispiele für einen Grenzwertfehler.