

Aufgabe 11 (1 Punkt)

Beantworten Sie folgende Fragen schriftlich in ganzen Sätzen oder mit Stichpunkten.

- a) Was versteht man unter einer Kano-Analyse von Anforderungen?
- b) Löst die Rupp-Schablone alle Probleme bei der Formulierung von funktionalen Anforderungen?
- c) Welche Anforderungen sind wichtiger, funktionale oder nicht-funktionale?

Aufgabe 12 (2+2 = 4 Punkte)

Zu einem Use Case „Rücksendungen von Kundschaft“ eines Verwaltungssystems eines Versandhauses existiert folgende informelle Beschreibung.

Das zu entwickelnde System soll folgende Abläufe unterstützen. Bei der eingehenden Rücksendung wird die Bestellnummer der ursprünglichen Bestellung eingegeben, danach wird manuell von einer Expert*in geprüft, ob eine Rücksendung erlaubt war und die reklamierten Artikel vollständig in ordnungsgemäßer Form zurückgegeben wurden. Bei Problemen wird die Rücksendung nicht angenommen, zur Kundschaft mit einer Erklärung zurück versandt und in der anzuschließenden Kundschaftsdatei ein Vermerk bei der Kundschaft gesetzt. Wurde die Rücksendung angenommen, wird aus dem anzuschließenden Bestellsystem die Bankverbindung der Kundschaft entnommen, der zugehörige Betrag an die Kundschaft überwiesen, in der Kundschaftsdatei die Rücksendung vermerkt und im anzuschließenden Warensystem die Rückgabe vermerkt. Es wird manuell geprüft, ob die Rücksendung erneut verkauft werden kann, ist dies der Fall, wird im Warensystem die Artikelanzahl passend erhöht.

- a) Leiten Sie aus der Beschreibung ein Aktivitätsdiagramm aus Systemsicht ab. Geben Sie ihren Aktionen „sprechende Namen“.
- b) Wandeln Sie Ihr Diagramm aus a) vollständig in textuelle Anforderungen unter konsequenter Nutzung der Anforderungsschablone von Rupp ab. Überlegen Sie dabei, welche Aktionen vom neuen System und welche wahrscheinlich außerhalb des Systems durchgeführt werden.

Aufgabe 13 (1 Punkt)

Beantworten Sie folgende Fragen schriftlich in ganzen Sätzen oder mit Stichpunkten.

- a) Welche 4 Pfeilarten haben Sie bis jetzt in UML-Klassendiagrammen kennengelernt?
- b) Wieviele Objekte des gleichen Typs können in der obersten Zeile eines Sequenzdiagramms stehen?
- c) Können in Java und der UML Interfaces voneinander erben? Wenn ja, wann könnte dies sinnvoll sein?
- d) Kann in Java und der UML eine Klasse abstrakt sein und trotzdem alle Methoden ausimplementiert sein? Wenn ja, wann ist dies sinnvoll?

Aufgabe 14 (2 Punkte)

Erstellen Sie ein Klassendiagramm, mit dem man folgende Zusammenhänge verdeutlichen kann.

- Es werden Studierende mit Namen und Matrikelnummer verwaltet. Jede studierende Person kann beliebig viele Prüfungen gemacht haben.
- Es werden Module mit Modulnummer, Namen und Inhalt verwaltet.
- Es werden lehrende Personen mit Namen und Personalnummer verwaltet.
- Es werden Lehrveranstaltungen verwaltet, die eine eindeutige Veranstaltungsnummer, das Semester, in dem sie stattfindet, und mindestens eine lehrende Person haben, sowie zu genau einem Modul gehören.
- Es werden Prüfungen verwaltet, mit denen individuelle Noten von Studierenden für genau ein Modul von einer oder zwei lehrenden Personen in der Rolle „Prüfend“

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

Objektorientierte Analyse und Design

Sommersemester 2022

4. Aufgabenblatt

festgehalten werden. Jede Prüfung hat weiter ein Datum, zu dem sie stattfindet und optional eine beisitzende Person deren Namen festgehalten wird.

Für jede der Klassen soll es einen Konstruktor geben, der Werte für die jeweiligen Exemplarvariablen liefert, für die es keinen sinnvollen Default-Wert gibt.

Die Verwaltungsklassen selbst können weggelassen werden.