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

Objektorientierte Analyse und Design

Sommersemester 2026
3. Aufgabenblatt

Aufgabe 0.3 (1 Punkt)
Geben Sie das Lösungswort des Quiz aus der Lernnotiz an.

Aufgabe 7 (1 Punkt)

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

a) Was für Arten von Aktoren kann es geben?
b) Wie hängen Aktoren und Stakeholder zusammen?
c) Erklären anschaulich an einem Beispiel, dass sich die Kritikalität von Use Cases eines Projekts unterscheiden kann.
d) Wenn unerfahrene modellierende Personen zusammen mit ihrer Kundschaft Use Cases erstellen, enden sie häufig mit Diagrammen, die viele <<include>>-Pfeile enthalten. Nennen Sie dafür mögliche Gründe.

Aufgabe 8 (3 Punkte)

Entwickelt werden soll durch Ihr IT-Unternehmen eine einfache Web-Interface-Software für die Belegung von Lehrveranstaltungen durch Studierende namens Belestud. Vor jedem Semester trägt das Studierendensekretariat alle Lehrveranstaltungen mit dem zugehörigen Semester und den Veranstaltungszeiten in das zu entwickelnde System ein. Das System soll das Definieren von Regeln zulassen, mit denen festgelegt werden kann, dass bestimmte Studierende eines Semesters sich nur in bestimmten Zeitintervallen als Teilnehmer eintragen können. Weiterhin soll Belestud mit dem existierenden Studierendeninformationssystem verbunden sein und so z. B. garantieren, dass Studierende, die ein Praktikum oder eine Vertiefung abgebrochen haben (diese Information sei dort gespeichert), sich nur auf die Warteliste zu dieser Veranstaltung setzen lassen können. Studierende können sich bei Belestud anmelden und in den ihrem Semester zugeordneten Zeitintervallen Lehrveranstaltungen belegen. Nach Abschluss der Belegung erhalten die veranstaltenden Personen der Lehrveranstaltungen Beleglisten mit den teilnehmenden Personen. Diese Listen werden über das existierende Mail-System versandt, an das Belestud angeschlossen werden soll.

a) Bestimmen Sie alle möglichen Stakeholder des Projekts (arbeiten Sie die Checkliste dazu ab, lassen Sie ihrer Kreativität etwas Lauf).
b) Strukturieren Sie das System mit minimal drei und maximal sechs System-Use Cases, die Sie zeichnen und zu denen Sie jeweils ein bis drei erklärende Sätze aufschreiben.

Aufgabe 9 (2 Punkte)

Eine kaufende Person hat folgende Beschreibung unserer Software-Firma für das erste Inkrement einer zu erstellenden Software geliefert.
*„(1) Unser Bankhaus benötigt ein neues System zum sicheren Aktienhandel, dabei soll unser altes System zur Verwaltung der Kundschaft weiter genutzt werden. (2) Zunächst sind von bei unserer Bank angestellten Personen die Informationen zu den Aktien selbst zu verwalten, die generell eine ID und einen Namen haben. (3) Weiterhin gehört zu jeder Aktie ein Einkaufspreis, zu dem eine Kund*in diese Aktie erwerben und ein Verkaufspreis mit dem eine Kund*in die Aktie an uns verkaufen kann. (4) Diese beiden Preise werden häufig von einer externen Trading-Software geändert. (5) Kund*innen können sich jederzeit über die aktuellen Preise informieren. (6) Jede Kund*in kann bei uns über die Kundschaftsverwaltung ein oder mehrere Depots anlegen, dabei hinterlegt die Kund*in pro Depot eine Geldmenge als Barreserve, die als Sicherheit für die Kund*in unter keinem Fall negativ werden darf. (7) Kund*innen können neue Aktien zu ihrem Depot hinzufügen und geben dazu eine noch nicht im Depot vorhandene Aktie und die gewünschte Anzahl an. (8) Reicht dann die Barreserve aus, wird das Depot um einen Posten ergänzt, der die hinzugefügten Aktien enthält. (9) Gibt es bereits einen Posten zu einer Aktie im Depot, können diese Aktien zum Verkaufspreis verkauft oder die Aktienanzahl zum Einkaufspreis erhöht werden. (10) Beim Kauf ist wieder die Barreserve zu*

*beachten. (11) Natürlich kann sich die Kund*in jederzeit über ihre Depots und die enthaltenen Posten informieren."*
Leiten Sie aus dem Text ein Use-Case-Diagramm ab und ordnen Sie schriftlich (z. B. im Diagramm) die Sätze den Use Cases zu. Dabei kann ein Satz durchaus mehreren Use Cases, in seltenen Fällen auch keinem Use Case, zugeordnet werden. Bedenken Sie, dass Use Cases eine grobe Gesamtübersicht liefern sollen.

Aufgabe 10 (2 Punkte, Vorgriff Collections)

a) Schreiben Sie eine Definition des Begriffs „Multimenge" auf und grenzen Sie ihn dabei gegen den Begriff „Menge" ab.
b) Multimengen sind leider nicht im Collection-Framework von Java enthalten; statt sie selber zu implementieren, ist es sinnvoll nach einer Umsetzung zu suchen. Diese kann z. B. mit der Bibliothek guava (https://github.com/google/guava) gefunden werden. Schreiben mit Hilfe der Klasse com.google.common.collect.HashMultiset und dem zugehörigen Interface Multiset eine Implementierung für den Tausch von Sammelbildern, von denen jeweils die Bilderserie und die Bildernummer bekannt sind. Benutzende Personen können ein Sammelbild dabei mehrfach zum Tausch anbieten. *Erinnern Sie sich an den Sinn von* `toString`, `equals` *und* `hashCode` *in Java.*
Von der Veranstaltungsseite kann ein Projekt ooadAufgabeMultiSet mit der Bibliothek und einem Nutzungsdialog geladen werden. Weiterhin steht eine Testklasse main.SystemTest zum Testen zur Verfügung. Der folgende Beispielnutzungsablauf sollte möglich sein, Eingaben sind zur Veranschaulichung umrandet, es haben vorher bereits einige Eingaben stattgefunden. Multisets haben bereits eine sinnvolle `toString`-Methode, das Herausgeben von n Bildern soll nur möglich sein, wenn mindestens n Bilder vorhanden sind, geben Sie dazu als Ergebnis einen Text, der entweder die Worte „erfolgreich" oder „nicht erfolgreich" enthält, aus. Orientieren Sie sich sehr eng an den Beispielausgaben, da die Tests hier nur auf diese Inhalte prüfen. Sie sollen im Projekt keine weiteren Klassenvariablen oder Klassenmethoden (static) ergänzen.
Nutzen Sie die Klasse main.SystemTest zum Testen Ihrer Implementierung.

```
(0) Programm beenden
 (1) Sammelbild hinzufuegen
 (2) Sammelbilder herausgeben
 (3) Nummern einer Serie zeigen
 (4) Gesamtbestand anzeigen:
4
Bilder: [Bild{WM22,123},
Bild{EM21,42} x 3, Bild{EM21,73}]
---------------
 (0) Programm beenden
 (1) Sammelbild hinzufuegen
 (2) Sammelbilder herausgeben
 (3) Nummern einer Serie zeigen
 (4) Gesamtbestand anzeigen:
1
Welche Serie? EM21
Welche Nummer? 73
---------------
 (0) Programm beenden
 (1) Sammelbild hinzufuegen
 (2) Sammelbilder herausgeben
 (3) Nummern einer Serie zeigen
 (4) Gesamtbestand anzeigen:
4
Bilder: [Bild{WM22,123},
Bild{EM21,42} x 3, Bild{EM21,73} x
2]
---------------
 (0) Programm beenden
 (1) Sammelbild hinzufuegen
 (2) Sammelbilder herausgeben
 (3) Nummern einer Serie zeigen
 (4) Gesamtbestand anzeigen:
3
Welche Serie? EM21
[73 x 2, 42 x 3]
---------------
 (0) Programm beenden
 (1) Sammelbild hinzufuegen
 (2) Sammelbilder herausgeben
 (3) Nummern einer Serie zeigen
 (4) Gesamtbestand anzeigen:
```

```
2
Welche Serie? EM21
Welche Nummer? 42
Wieviele? 2
Loeschen erfolgreich
---------------
 (0) Programm beenden
 (1) Sammelbild hinzufuegen
 (2) Sammelbilder herausgeben
 (3) Nummern einer Serie zeigen
 (4) Gesamtbestand anzeigen:
4
Bilder: [Bild{WM22,123},
Bild{EM21,42}, Bild{EM21,73} x 2]
---------------
 (0) Programm beenden
 (1) Sammelbild hinzufuegen
 (2) Sammelbilder herausgeben
 (3) Nummern einer Serie zeigen
 (4) Gesamtbestand anzeigen:
2
Welche Serie? EM21
Welche Nummer? 42
Wieviele? 2
Loeschen nicht erfolgreich
---------------
 (0) Programm beenden
 (1) Sammelbild hinzufuegen
 (2) Sammelbilder herausgeben
 (3) Nummern einer Serie zeigen
 (4) Gesamtbestand anzeigen:
0
```