

Aufgabe 7 (4 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 Veranstalter der Lehrveranstaltungen Beleglisten mit den Teilnehmern. 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 8 (3 Punkte)

Ein Versandhaus möchte fast seine gesamte Software neu entwickeln lassen. Als einziges Legacy-System wird ein Abrechnungsmodul übernommen, das die gesamten Daten der Kunden und der Zulieferer verwaltet und in das neue System integriert werden muss. Das Versandhaus nimmt Bestellungen sowohl Online als auch per Post an. Für die Waren gibt es ein Warenlager, das mit Waren der Zulieferer befüllt wird. Bestellungen werden an die Kunden versandt, die danach bezahlen müssen. Kunden können versandte Artikel auch zurück senden. Das Internet-Frontend zur Online-Bestellung wird getrennt in einem anderen Projekt entwickelt.

Entwickeln Sie nach dem in der Veranstaltung vorgestellten Ansatz zur Use Case-Ermittlung für das neu zu erstellende System ein Use Case Diagramm. Wählen Sie „sprechende Namen“ für Ihre Use Cases und beschreiben Sie mit ca. drei kurzen Sätzen die Funktionalität hinter jedem Use Case. Sie sollten auf ca. 5 bis 15 System-Use Cases kommen.

Aufgabe 9 (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. Nutzer 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 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.

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:

Bilder: [Bild{WM14,123}, Bild{EM16,42} x 3,
Bild{EM16,73}]

- (0) Programm beenden
- (1) Sammelbild hinzufuegen
- (2) Sammelbilder herausgeben
- (3) Nummern einer Serie zeigen
- (4) Gesamtbestand anzeigen:

Welche Serie?

Welche Nummer?

- (0) Programm beenden
- (1) Sammelbild hinzufuegen
- (2) Sammelbilder herausgeben
- (3) Nummern einer Serie zeigen
- (4) Gesamtbestand anzeigen:

Bilder: [Bild{WM14,123}, Bild{EM16,42} x 3,
Bild{EM16,73} x 2]

- (0) Programm beenden
- (1) Sammelbild hinzufuegen
- (2) Sammelbilder herausgeben
- (3) Nummern einer Serie zeigen
- (4) Gesamtbestand anzeigen:

Welche Serie?

[73 x 2, 42 x 3]

- (0) Programm beenden

(1) Sammelbild hinzufuegen

- (2) Sammelbilder herausgeben
- (3) Nummern einer Serie zeigen
- (4) Gesamtbestand anzeigen:

Welche Serie?

Welche Nummer?

Wieviele?

Loeschen erfolgreich

- (0) Programm beenden
- (1) Sammelbild hinzufuegen
- (2) Sammelbilder herausgeben
- (3) Nummern einer Serie zeigen
- (4) Gesamtbestand anzeigen:

Bilder: [Bild{WM14,123}, Bild{EM16,42},
Bild{EM16,73} x 2]

- (0) Programm beenden
- (1) Sammelbild hinzufuegen
- (2) Sammelbilder herausgeben
- (3) Nummern einer Serie zeigen
- (4) Gesamtbestand anzeigen:

Welche Serie?

Welche Nummer?

Wieviele?

Loeschen nicht erfolgreich

- (0) Programm beenden
- (1) Sammelbild hinzufuegen
- (2) Sammelbilder herausgeben
- (3) Nummern einer Serie zeigen
- (4) Gesamtbestand anzeigen: