



1. Zulassungsklausur (C) zur Veranstaltung

Einführung in die Informatik II - Vertiefung

und Allgemeine Informatik 2

im Sommersemester 2022

Prüfer: Dr. Jens Kohlmeyer

Fakultät Ingenieurwissenschaften, Informatik, Psychologie

4.06.2022, 12 Uhr

Bearbeitungszeit: 45 min

Nachname:	Vorname:	Matrikelnummer:
Studiengang und Abschluss:		Account:
<p>Hiermit erkläre ich, dass ich prüfungsfähig bin. Sollte ich nicht auf der Liste der angemeldeten Studierenden aufgeführt sein, dann nehme ich hiermit zur Kenntnis, dass diese Prüfung nicht gewertet werden wird.</p> <p>_____</p> <p>Datum, Unterschrift des Prüfungsteilnehmers</p>		

Zur allgemeinen Beachtung:

- Füllen Sie das Deckblatt vollständig und korrekt aus.
- Lesen Sie sich zunächst die Klausur sorgfältig durch (sie besteht aus 2 Seiten).
- Zum Erreichen der Vorleistung benötigen Sie **50** der insgesamt **100** Punkte, die Sie über die 2 Zulassungsklausuren hinweg erreichen können.
- Im Ordner `~/materials` finden Sie das zu bearbeitende Projekt sowohl als ZIP als auch entpackt.
- Sie dürfen im Projekt beliebig Hilfsmethoden und Klassen hinzufügen. Die Signaturen der vorgegebenen Methoden dürfen nicht verändert werden.
- **Vor Beginn der Bearbeitungszeit** haben Sie 5 Minuten Zeit um einen Editor/eine IDE Ihrer Wahl zu starten und sich in den Materialien zurecht zu finden.
- **Nach Ablauf der Bearbeitungszeit** haben Sie 5 Minuten Zeit um ihre Lösung im Ordner `~/export` abzulegen. **Bleiben Sie sitzen!**
- Abgaben sind als Projeklexport in Zip-Form abzugeben.

Punkteverteilung				
1	2	3	Σ	Note
von 12	von 7	von 11	von 30	
				Korrektur

Das Projekt

Das Ihnen zur Verfügung gestellte Projekt setzt ein Bankensystem um. In diesem System soll es jeder Bank möglich sein die Vaults aller Banken einzusehen, aber nur die eigenen Accounts zu verändern. Diese Funktionalität wird durch die Verwendung einer zentralen Verwaltungseinheit (VaultingManager) umgesetzt.

Das Projekt besteht aus **4** Klassen und einem Interface. Im folgenden eine kurze Übersicht über die einzelnen Klassen:

`eid12.sose2022.admission_exam.group02.Main`: Diese Klasse enthält keinen funktionalen Inhalt. Sie können diese verwenden um ihre Implementierungen zu testen.

`eid12.sose2022.admission_exam.group01.Bank`: Diese Klasse stellt die zentrale Verwaltungseinheit des Projektes dar. In ihr sollen verschiedene Funktionalitäten umgesetzt werden. Die Klasse besitzt drei Attribute. `knownVaults` repräsentiert alle Bankaccounts die bei der Bank eröffnet wurden. `foreignVaults` repräsentiert alle Bankaccounts die bei anderen Banken eröffnet wurden. `support` stellt die nötige Funktionalität zur Verwendung von `PropertyChangeEvents` zur Verfügung.

`eid12.sose2022.admission_exam.exceptions.NoSuchVaultException`: Diese Klasse wird dazu verwendet eine Laufzeitfehlermeldung für bestimmte Anwendungsfälle zur Verfügung zu stellen.

`eid12.sose2022.admission_exam.storage.IStoreable`: Dieses Interface definiert welche Methoden jedes in Vaults enthaltene Objekt implementieren muss.

`eid12.sose2022.admission_exam.storage.Vault`: Diese Klasse bildet Bankaccounts im System ab. Sie enthält eine eindeutige ID und eine Liste von Objekten die im Account gespeichert werden.

Aufgabe 1 - PropertyChange**4 + 8 = 12 Punkte**

a) Implementieren Sie die Methode `propertyChange(PropertyChangeEvent evt)` der Klasse `Bank`. Die Methode soll prüfen ob die Quelle des `PropertyChange` Events nicht das aktuelle Objekt ist, und in diesem Fall den Vault, der im `newValue` des Events steckt, zu den `foreignVaults` hinzufügen.

Hinweis: Beachten Sie, dass die Werte in *PropertyChange* Events Objects sind und umgewandelt werden müssen. Sie dürfen davon ausgehen, dass diese immer vom Typ `Vault` sind.

b) Implementieren Sie die Methode `addVault(Vault<> vaultToAdd)` der Klasse `Bank`. Die Methode soll einen übergebenen Vault den `knownVaults` hinzufügen und ein `PropertyChangeEvent` feuern bei dem das manipulierte Attribut `knownVaults` ist, der alte Wert `null` und der neue der hinzugefügte Vault. Behandeln Sie illegale Aufrufe mit einer `IllegalArgumentException`.

Aufgabe 2 - AddStoreable**7 Punkte**

Implementieren Sie die Methode `addStoreableToBankVault(long vaultId, IStoreable storeableToAdd)` der Klasse `Bank`. Die Methode soll dem Vault mit der übergebenen ID das übergebene Storeable anfügen. Dabei soll egal sein, ob der Vault der aktuellen Bank angehört oder einer anderen. Die Methode soll `true` zurückgeben wenn das Hinzufügen des Storeables funktioniert hat. Ansonsten soll `false` zurückgegeben werden.

Aufgabe 3 - RemoveValue**11 Punkte**

Implementieren Sie die Methode `removeValue(long vaultId, int valueToRemove)` der Klasse `Bank`. Die Methode soll bei dem Vault mit der übergebenen ID, das erste Element mit dem Wert `valueToRemove` entfernen und `true` zurückliefern. Dies soll nur geschehen, wenn der Vault der aktuellen Bank angehört. Ansonsten soll `false` zurückgegeben werden.

