



1. Zulassungsklausur (B) 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, 11 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	4 Σ	Note
von 8	von 8	von 14	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 Vaults zu verändern. Diese Funktionalität wird durch die Verwendung einer zentralen Verwaltungseinheit (VaultingManager) umgesetzt.

Das Projekt besteht aus **5** Klassen und einem Interface. Im folgenden eine kurze Übersicht über die einzelnen Klassen:

`eid12.sose2022.admission_exam.group01.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 im System zu verwalten- den Banken dar. Alle Funktionalität ist bereits umgesetzt.

`eid12.sose2022.admission_exam.group01.VaultingManager`: Diese Klasse stellt die zentrale Verwaltungseinheit des Projektes dar. In ihr sollen verschiedene Funktionalitäten umgesetzt werden. Die Klasse besitzt ein einzelnes Attribut `banksAndVaults`. Dieses Attribut soll die Zuordnung aller Banken zu deren Accounts beinhalten.

`eid12.sose2022.admission_exam.util.Tuple`: Diese Klasse wird dazu verwendet eine Zuordnung zwischen den zwei darin enthaltenen Objekten zu ermöglichen.

`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 - Vault**8 Punkte**

Implementieren Sie die Methode `addStoreableToVault(T value)` der Klasse `Vault`. Diese soll das übergebene Element dem Vault hinzufügen und `true` zurückliefern. Sollte das übergebene Element `null` sein, oder einen Wert kleiner oder gleich 0 haben, so soll es nicht hinzugefügt werden und `false` zurückgegeben werden. **Hinweis:** Beachten Sie die Definition von `Vault` und den Typ des `contents` Attributs.

Aufgabe 2 - AddVaultForBank**8 Punkte**

Implementieren Sie die Methode `addVaultForBank(Bank bank, Vault<> vaultToAdd)` der Klasse `VaultingManager`. Die Methode soll den übergebenen Vault (`vaultToAdd`) der Liste, die der übergebenen Bank (`bank`) zugeordnet ist, hinzufügen. Sollte die Bank noch nicht in der Liste aller Banken und Vaults existieren, so soll ein neues Tupel mit der übergebenen Bank und einer leeren Liste erstellt werden. Dann soll der übergebene Vault dort angehängt werden. Die Methode soll eine `IllegalArgumentException` werfen, falls die übergebene Bank oder der übergebene Vault `null` sein sollten.

Aufgabe 3 - AddStoreable**14 Punkte**

Implementieren Sie die Methode `addStoreableToBankVault(long vaultID, IStoreable storeableToAdd)` der Klasse `VaultingManager`. Die Methode soll dem Vault mit der ID `vaultID` das übergebene Storeable anhängen und `true` zurückliefern wenn dies funktioniert hat. Sollte der Vault nicht existieren oder das Hinzufügen aus anderen Gründen scheitern, soll `false` zurückgegeben werden.

