Praktikum Verteilte Systeme

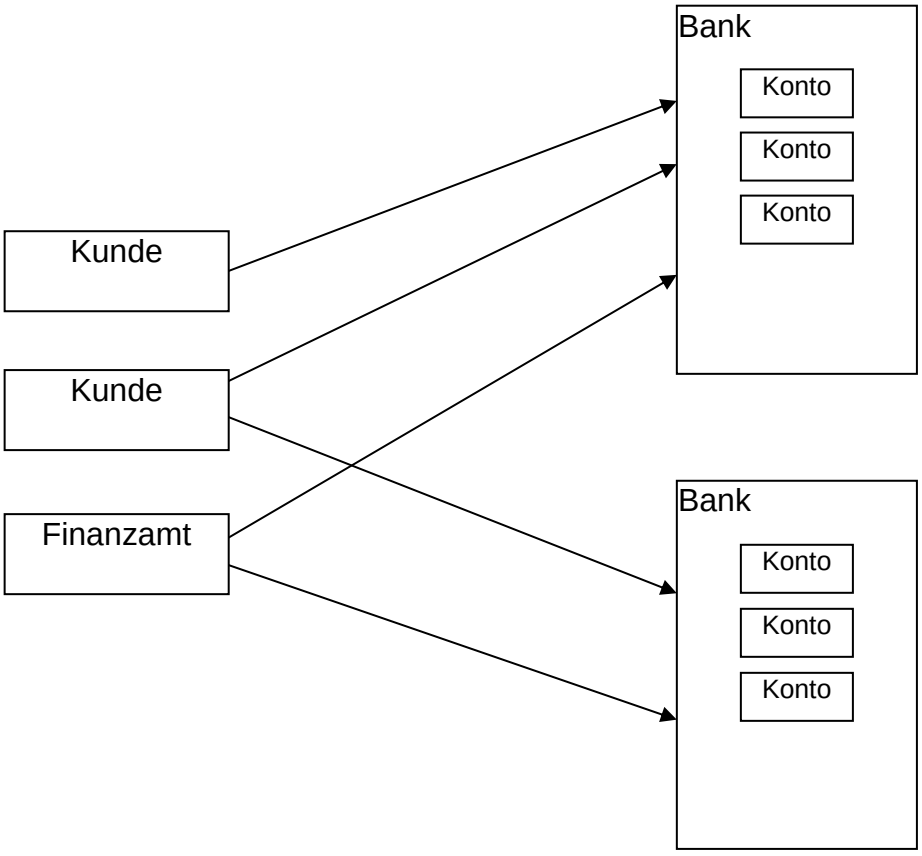
Aufgabe 1: RMI / Java: Banken und Finanzamt

Programmieren Sie mit Hilfe von RMI eine Client-Server-Anwendung, in der:

- Kunden bei mehreren Banken Konten einrichten und verwalten können und
- bei der sich das Finanzamt Informationen über die Konten bei den Banken verschaffen kann.

Dabei sind die Kunden und das Finanzamt Clients der Banken (Server) und können auf die Banken über eine Kunden- bzw. Finanzamts-Schnittstelle zugreifen. Die Banken enthalten die Konten der Kunden.

Allgemeine Architektur:



Ein Kunde kann bei einer Bank.

- ein Konto einrichten,
- einen Betrag auf ein beliebiges Konto einzahlen,
- einen Betrag von einem ihm gehörenden Konto abheben,
- sich den Kontostand eines ihm gehörenden Kontos angeben lassen.

Das Finanzamt kann sich:

- die Summe aller Kontostände eines Kunden,
- die Nummern der Konten eines Kunden

bei einer Bank (und damit natürlich nacheinander bei allen Banken) geben lassen.

Zur Implementierung:

Eine Bank muss ihre Konten organisieren. Hierfür können Sie eine Datenstruktur Ihrer Wahl verwenden, z.B. einen Array, eine Hashtabelle oder jede Ihnen günstig erscheinende Datenstruktur. Java stellt geeignete Collections mit den zugehörigen Methoden bereit. Als Schlüssel für ein Konto soll die Nummer des Kontos verwendet werden, außerdem werden für ein Konto der Name des Besitzers, das Passwort und der Kontostand (Festkommazahl) gespeichert.

Zum Interface für den Kunden:

- Beim Einrichten eines Kontos soll der Kunde seinen Namen und ein von ihm auszuwählendes Passwort eingeben. Die Kontonummer wird von der Bank vergeben (zufallsgeneriert oder einfach aufsteigende ganze Zahlen), der Kontostand wird auf *0.00* gesetzt.
- Beim Einzahlen auf das Konto werden Nummer des Kontos und Name des Kontobesitzers angegeben, außerdem der Einzahlungsbetrag. Bei nichtexistierender Nummer oder Name oder wenn Nummer und Name nicht zusammenpassen, wird eine für den Client sichtbare **Exception** geworfen, die die Ursache des Problems angibt.
- Beim Abheben vom Konto werden Nummer des Kontos und Name und Passwort des Kontobesitzers angegeben, außerdem der Abhebungsbetrag. Eine für den Client sichtbare problemspezifische **Exception** wird unter den gleichen Bedingungen wie beim Einzahlen geworfen, außerdem wenn das Konto nicht gedeckt ist oder wenn das Passwort nicht stimmt.
- Bei der Angabe des Kontostandes werden Nummer des Kontos und Name und Passwort des Kontobesitzers eingegeben. Zurückgegeben wird ein **serialisierbares Objekt** mit Kontonummer und Kontostand. Eine für den Client sichtbare **Exception** wird unter den gleichen Bedingungen wie beim Einzahlen geworfen oder wenn das Passwort nicht stimmt.

Zum Interface für das Finanzamt:

- Für die Berechnung der Summe der Kontostände eines Kunden bei einer Bank gibt das Finanzamt den Namen des Kontobesitzers an. Zurückgegeben wird einfach ein Betrag.
- Für die Auflistung der Kontonummern eines Kunden bei einer Bank gibt das Finanzamt ebenfalls den Namen des Kontobesitzers an. Zurückgegeben wird eine Collection von Kontonummern.

Empfohlene Imports:

Schnittstellendefinitionen (für Kunde und Finanzamt): *java.rmi.**
Schnittstellenimplementierung: *java.rmi.*, java.rmi.server.*, java.util.**
Server: *java.rmi.*, java.rmi.server.*, java.rmi.registry.**
BankClient: *java.rmi.**

FinanzClient: java.rmi.Naming, java.util.*
serialisierbare Klasse für Kontostandsangabe: java.io.Serializable, java.rmi.*

Nützliche Klassen und Interfaces:

Collection<E>, Hashtable<K,V>, Iterator<E>, Enumeration<E>, ArrayList<E>
mit: K=Key, V=Value, E=Element

Hinweis:
Es wird Wert darauf gelegt, dass:

- Kunde-Client und Finanzamt-Client **eigene Schnittstellen** verwenden (also zwei Schnittstellendefinitionen erstellen),
- für Einzahlen, Abheben und Kontostandsangabe bei Konten benutzerdefinierte **Exceptions** implementiert werden, *die am Client sichtbar werden*,
- bei der Kontostandsangabe ein **serialisierbares Objekt** zurückgegeben wird.

Materialien:
Zur Orientierung ist auf Moodle die RMI-Anwendung über die Addition komplexer Zahlen aus der Vorlesung mitgegeben.

Abgabe:
Sie können diese Aufgabe in Präsenz am Praktikumstermin entwickeln und / oder auf Moodle mit Programmcode und Ablaufprotokoll hochladen. Das Ablaufprotokoll soll (in etwa) folgendes enthalten:

Richten Sie für drei Kunden bei zwei Banken Konten ein, wobei einzelne Kunden bei der gleichen Bank und/oder bei verschiedenen Banken mehrere Konten unterhalten können. Zahlen Sie ein, heben Sie ab und lassen Sie Kontostände ausgeben. Lassen Sie das Finanzamt die Kunden überprüfen. Geben Sie gelegentlich falsche Nummern, Namen oder Passwörter an und überziehen Sie mal ein Konto.

Die Kunden-Clients und Finanzamt-Clients sollen interaktiv gestaltet werden (in einer Schleife: "Was wollen Sie als nächstes tun?").