Praktikum Verteilte Systeme

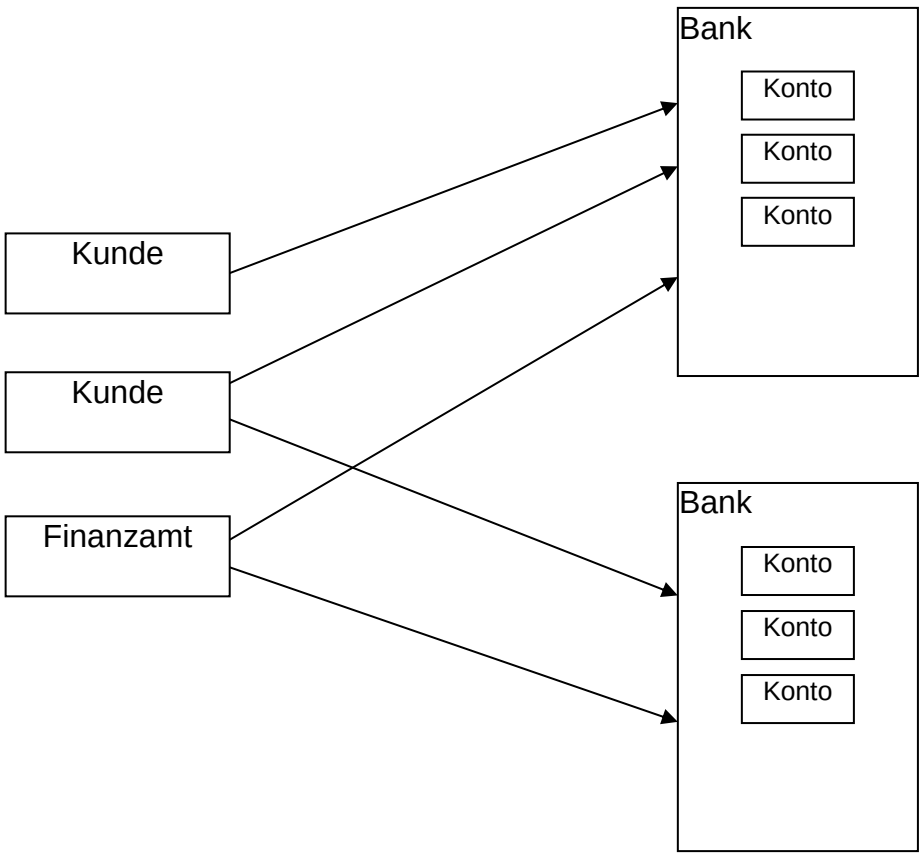
Aufgabe 1: RMI / Java: Banken und Finanzamt

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, die auf Banken über eine entfernte Schnittstelle zugreifen können. 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:

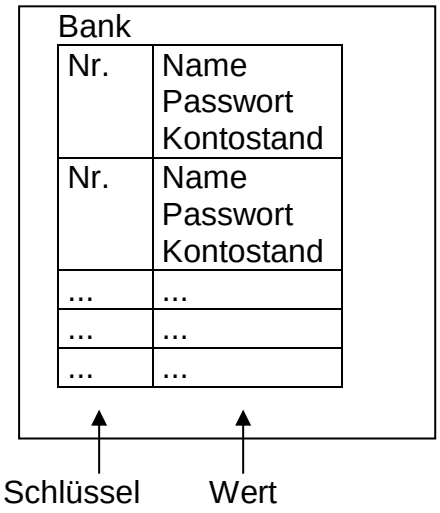
- die Summe aller Kontostände eines Kunden,

- die Nummern der Konten eines Kunden

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

Zur Implementierung:

Eine Bank organisiert ihre Konten in einer Collection, z.B. in einer Hashtabelle mit *Schlüsseln* und zugehörigen *Werten*. Dabei ist der Schlüssel der Hashtabelle die *Nummer* eines Kontos. Der einem Schlüssel zugeordnete Wert eines Hashtabellen-Eintrags ist ein Objekt, das *Name*, *Passwort* und *Kontostand* (Festkommazahl) enthält.



Zur Datenstruktur Hashtable:

Die Hashtabelle wird in der Schnittstellenimplementierung als Variable z.B. vom Typ *Hashtable<Integer, Account>* definiert, wobei *Account* eine serialisierbare Klasse ist, die *Name*, *Passwort*, *Kontostand*, einen Konstruktor und die nötigen *Getter* und *Setter* enthält. Die Hashtable stellt insbesondere mehrere Konstruktoren und z.B. die Methoden *containsKey(), get(), keys(), remove(), put(), values()* zur Verfügung, die zur Kontenverwaltung eingesetzt werden können.

Zum Interface für den Kunden:

- Beim Einrichten eines Kontos müssen Name und Passwort des Kunden eingegeben werden. Die Kontonummer wird von der Bank vergeben, der Kontostand wird auf *0* 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 angegeben. 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 gibt das Finanzamt den Namen des Kontobesitzers an. Zurückgegeben wird einfach ein Betrag.
- Für die Auflistung der Kontonummern eines Kunden gibt das Finanzamt ebenfalls den Namen des Kontobesitzers an. Zurückgegeben wird eine Collection von Kontonummern.

Empfohlene Imports:

Schnittstellendefinition: *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.*
Account-Klasse: java.io.Serializable, java.rmi.*

Nützliche Klassen und Interfaces:

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

Testen Sie ihre RMI-Bankanwendung, indem sie für drei Kunden Konten einrichten, Beträge einzahlen und abheben und Kontostände ausgeben lassen. Lassen Sie das Finanzamt die Kunden überprüfen. Geben Sie gelegentlich falsche Nummern, Namen oder Passwörter an.

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

Vorführung dieser Aufgabe am Ende des Praktikumstermins