

Übungen zu Verteilte Systeme

Hochschule Regensburg

02.12.2013, Übung 7

Universitätsstraße 31, 93053 Regensburg

Prof. Dr. Jan Dünneberger

- Implementieren Sie die Klasse `hsr.bank.Account` für ein Bankkonto mit den Methoden
 - ▶ `void deposit(int value)`
 - ▶ `void widthdraw(int value)`
- Machen Sie die das Konto via RMI im Netzwerk verfügbar und überprüfen Sie, dass eine Abfolge mehrerer Einzahlungen (bzw. Abbuchungen) sich dauerhaft auf den Zustand des Kontos (\approx den Kontostand) auswirken
- Überladen Sie die Methode zum Einzahlen, so dass auch Einzahlungen per Scheck (eigene Klasse `Cheque`) möglich sind
- Schreiben Sie einen Testclient, der Überweisungen per Scheck und Bareinzahlung tätigt

- Implementieren Sie eine weitere im Netzwerk verfügbare Klasse `Authority`, deren Methode `approve(Cheque c)` zur Genehmigung von Scheckeinzahlungen dient
- Erweitern Sie die Methode `deposit()`, so dass bei einer Scheckeinzahlung zunächst eine Genehmigung eingeholt wird
- Lagern Sie das Einholen der Genehmigung in einen eigenen Thread aus
(**Hinweis:** der Approval-Prozess muss jetzt in zwei Schritte, `requestApproval()` und `obtainApproval()`, zerlegt werden)
- Stellen Sie mittels `transient`-Deklarationen sicher, dass komplexwertige Attribute eines Schecks (z.B. die ausstellende Bank) nicht über das Netzwerk übertragen werden
- Implementieren Sie für die Klasse `Cheque`, das Interface `Externalizable`, so dass für jedes komplexwertige Attribut nur eine Kennung (z.B. der Name der ausstellenden Bank) übertragen wird