Übungsaufgabe #7:CORBA-Client und Server *25 Punkte*

18.12.2002

In dieser Aufgabe soll ein Client und ein Server zur Verwaltung von Büchern einer Bibliothek implementiert werden. Hierzu steht unter `/proj/i4oovs/pub/aufgabe7` eine IDL bereit, die zu verwenden ist.
Generell gilt im Folgenden: Alle Client-Klassen sollen im Paket `library.client` abgelegt werden, alle Server-Klassen in `library.server`.

Teil 1: Client

a) Im ersten Teil soll ein kommandozeilenbasierter Client (wahlweise auch ein GUIbasierter Client (AWT oder Swing)) für ein Bibliotheksystem implementiert werden, der diese Funktionen unterstützt:

- Eingabe eines Namens zum Erzeugen oder Ermitteln eines Ausweises (card)
- Suchen aller Bücher, deren Titel einen angegebenen Suchstring enthält, und Ausgabe der Treffer (jeweils einen Titel anzeigen, vorwärts- und rückwärts blättern)
- Entleihen eines gefunden Buches
- Rückgabe eines Buches
- Erstellen einer Liste aller Bücher, die auf einen bestimmten Ausweis entliehen sind

Ein einfacher Testserver (IOR in `/proj/i4oovs/pub/aufgabe7/Librarian.ior`) steht bereit, mit dem der eigene Client getestet werden kann.

b) Im nächsten Schritt soll der Client so erweitert werden, dass er den CORBA-Namensdienst verwendet, um ein `Librarian`-Objekt zu bekommen. Der Testserver ist dort unter dem Namen `oovs/Librarian` registriert. Der eigene Server aus Teil 2 soll unter <login>/`Librarian` dort registriert werden (wobei <login> der eigene Login-Name sein soll). Unter `/proj/i4oovs/pub/aufgabe7/Nameservice.ior` wird die Adresse des Root-Namensdiensts bekanntgegeben.

Teil 2: Einfacher Server

c) Im zweiten Teil soll nun die Serverseite implementiert werden. Als erster Schritt soll ein ganz einfacher `LibrarianServant` implementiert werden, der

- bei `issueCard` eine `DeniedCard`-Exception erzeugt,
- bei `findCardByOwner` eine `NoSuchCard`-Exception erzeugt,
- bei `findBookByTitle` eine leere Sequenz von `Books` zurückliefert

Ein Klasse library.server.Main soll einen `LibrarianServant` erzeugen, und ihn beim Namensdienst als <login>/`Librarian` anmelden. Der Servant kann nun mit dem eigenen Client getestet werden.

d) Im nächsten Schritt soll nun für Bücher und Ausweise jeweils ein Servant (`BookServant`, `CardServant`) implementiert werden. Im Konstruktor von `LibrarianServant` können zum Testen einige `BookServants` erzeugt werden, die `CardServant`-Objekte werden in der Methode `issueCard` erzeugt.
Der `LibrarianServant` kann nun entsprechend so erweitert werden, dass er die Bücher und Ausweise in einem `Vector` speichert, und die in (c) beschriebenen Methoden sinnvoll implementiert.

Teil 3: Besserer Server

Da eine Bibliothek gewöhnlich eine grosse Anzahl an Büchern enthält, ist es nicht sinnvoll, für jedes Buch einen aktiven Servant zu erzeugen. Der Server sollte daher nur einen Cache mit den zuletzt verwendeten Bücher verwalten, die persistente Speicherung der Daten der Bücher wird zweckmässigerweise in einer Datenbank vorgenommen. Hierzu wird eine einfache Pseudo-Datenbank unter `library.database.BookDatabase` zur Verfügung gestellt.

e) Als erstes ist die Main-Methode anzupassen: Für die Implementierung des Caches muss ein eigener POA (`BookPOA`) unterhalb des RootPOA erstellt werden (Policies: `USER_ID`, `USE_SERVANT_MANAGER` und `NON_RETAIN`). Für das dynamische Erzeugen von Buch-Servants muss beim `BookPOA` ein Servant-Manager (`BookServantLocator`) registriert werden.

f) Nun muss der `BookServantLocator` implementiert werden:

- Als erstes braucht man eine eindeutige ID für Bücher. Diese kann direkt aus dem Index in der Datenbank gewonnen werden kann (Konvertierung zwischen `int` und `byte[]`!)
- In der `preinvoke`-Methode soll der `ServantLocator` testen, ob es zu der angeforderten ID bereits einen `BookServant` im Cache gibt. Wenn vorhanden, wird dieser verwendet, ansonsten wird ein neuer `BookServant` mit den Daten aus der Datenbank erzeugt. Der Cache kann sehr einfach verwaltet werden (Hash über ID oder FIFO). Zum Konvertieren von CORBA-Referenzen in Objekt IDs kann die POA-Methode `reference_to_id` (von dem POA, der die CORBA-Referenz erzeugt hat, siehe (g)) verwendet werden.
- In der `postinvoke`-Methode muss, falls das Buch geändert wurde (Informationen zu "Ausgeliehen an") dieses in die Datenbank zurückgeschrieben werden.

g) Zuletzt ist die Methode `findBooksByTitle` im `LibrarianServant` anzupassen. Diese soll direkt über die Datenbank die IDs der passenden Bücher ermitteln. Aus diesen IDs kann über den `BookPOA` (mittels `create_reference_with_id`) eine CORBA-Objektreferenz erzeugt werden, die als Rückgabe der Methode verwendet werden kann. Hierzu müssen an dieser Stelle keine Daten aus der Datenbank gelesen und auch keine Servants erzeugt werden (das soll später dynamisch durch den `BookServantLocator` geschehen)!

Bearbeitung: bis zum 23.1.2003 17:00 Uhr

Wir wünschen Ihnen ein schönes Weihnachtsfest und einen guten Rutsch in das Jahr 2003