
2 Übungsaufgabe #2: Library – Java Remote Method Invocation

2.1 Allgemeines

In dieser Aufgabe soll die Bibliotheksverwaltung als Client/Server-Anwendung mit Hilfe der in Java enthaltenen *Remote-Method-Invocation-API (RMI)* gelöst werden. Als Ausgangspunkt dient dabei die Bibliotheksverwaltung aus Aufgabe 1.

2.2 Kommunikation mit der Datenbank

In Aufgabe 1 wurde die Kommunikation zwischen den Frontends (*LibraryFrontends*) und der Datenbank (*SimpleDB*) über einen selbst implementierten Proxy (*SimpleDBProxy*) abgewickelt. Hierfür soll nun statt dem Proxy Java-RMI zum Einsatz kommen.

Zu diesem Zweck ist die Datenbank als Remote-Objekt bereitzustellen, auf das alle *LibraryFrontends* über einen Namensdienst zugreifen können. Als Basis dient dabei die Klasse *RMIServer*, die ein Objekt vom Typ *SimpleDBImpl* erzeugt, es als Remote-Objekt exportiert und anschließend über den Namensdienst von Java-RMI bekannt macht. Das *LibraryFrontend* ist im Zuge dessen so zu verändern, dass es sich über den Namensdienst eine Referenz auf die Datenbank holt und für die Kommunikation mit der Datenbank ausschließlich RMI-Fernaufrufe verwendet.

Aufgaben:

- Veränderung der *SimpleDB*-Schnittstelle, so dass ein Objekt vom Typ *SimpleDB* als Remote-Objekt dient
- Implementierung von `<gruppe>.mwlibrary.server.RMIServer`
- Anpassung von *LibraryFrontend*

Hinweise:

- Die vom Namensdienst benötigte URL soll sowohl dem *RMIServer* als auch den *LibraryFrontends* als Aufrufparameter übergeben werden.
- Zum Testen kann die RMI-Registry auf einem Rechner im CIP-Pool gestartet werden. Dabei ist die eigene *UID* als Port-Nummer zu benutzen.

2.3 Rückrufe

Jedes *LibraryFrontend* soll nun eine kurze Meldung ausgeben, sobald ein neues *Item* in der Datenbank eingetragen wird. Dies ist mit Hilfe des *Observer*-Entwurfsmusters zu realisieren.

Ein *LibraryFrontend* muss hierfür die Möglichkeit haben sich bei der Datenbank zu registrieren. Bei jedem Aufruf von *register()* soll die Datenbank alle registrierten *LibraryFrontends* über die Zustandsänderung informieren. Zu diesem Zweck muss auch ein Remote-Objekt auf der Frontend-Seite implementiert werden. Dieses soll die Remote-Schnittstelle *DBObserver* implementieren, die über eine Methode *newItem()* verfügt.

Aufgaben:

- Implementierung der Rückruf-Funktionalität
- Bereitstellen von `<gruppe>.mwlibrary.DBObserver`
- Erweiterung und Anpassung von *LibraryFrontend* sowie der Datenbankimplementierung

Hinweise:

- Um ein Remote-Objekt wirklich als ein solches weitergegeben zu können, muss es zuvor exportiert werden.

2.4 Abgabe: am 26.11.2009 in der Rechnerübung