

## Übungsaufgabe #7: Library - .NET

10.01.2006

In dieser Aufgabe soll eine vereinfachte Client/Server-Anwendung ähnlich Aufgabe 2 in C# für die .NET-Architektur implementiert werden.

- a) Als erster Schritt soll die Klasse `Library.LibraryDB` implementiert und als Bibliothek erstellt werden (`LibraryDB.dll`). Items werden durch die Klasse `Item` mit den beiden Properties `Title` und `Borrowed` repräsentiert. Die Klasse `LibraryDB` soll das Interface `ILibraryDB` (in `/proj/i4mw/pub/aufgabe7`) implementieren. Fehler sind als Exceptions zu signalisieren (`AlreadyRegisteredException`, `NotFoundException`, `AlreadyBorrowedException`, `NotBorrowedException`); als Vorlage kann die Datei `AlreadyRegisteredException.cs` im `pub/aufgabe7`-Verzeichnis dienen. Zum Übersetzen und Ausführen im CIP-Pool bitte "Mono" verwenden (Binaries liegen unter `/usr/bin`) und ein geeignetes Makefile bereitstellen:

```
> mcs -target:library LibraryDB.cs Item.cs NotFoundException.cs
    AlreadyRegisteredException.cs NotBorrowedException.cs ...
```

Getestet werden kann die Bibliothek mit der Klasse `LibraryTest` (ebenfalls in `pub`):

```
> mcs -r:LibraryDB.dll LibraryTest.cs
> mono LibraryTest.exe
```

- b) Nun soll die Klasse `LibraryFrontend` implementiert werden, mit der man ähnlich wie in Aufgabe 1 über ein einfaches Benutzerinterface auf die Bibliotheksfunktionen zugreifen kann. Insbesondere soll es möglich sein, neue Items zu registrieren, auszuleihen und zurückzugeben und sich alle Items anzeigen zu lassen.
- c) In dieser Teilaufgabe soll die Bibliothek nun in einen Client und einen Server aufgespalten werden. Schreiben sie die Klasse `LibraryServer`, die einen TCP- oder HTTP-Channel erzeugt. Als Port können sie ihre User-Id verwenden (Befehl `id`). Die Klasse `Library.LibraryDB` soll als "well-known" Serverobjekt (mit beliebigen Namen) registriert werden. Achten sie darauf, dass der Server sich nicht sofort beendet (z.B. durch Warten auf eine Tastatureingabe). Sorgen sie auch dafür, dass die Klasse `LibraryDB` als Remote-Objekt verwendet werden kann. Da im Interface `ILibraryDB` keine expliziten lock/unlock-Operationen vorgesehen sind, muss die Datenbank intern für gegenseitigen Ausschluss beim Ausleihen und Zurückgeben von Items sorgen.
- Die Klasse `LibraryFrontend` (der Client) muss noch um die Aktivierung der serverseitigen `LibraryDB` ergänzt werden. Erzeugt werden können Client und Server mit den Befehlen:

```
> mcs -r:LibraryDB.dll, System.Runtime.Remoting.dll
    LibraryFrontend.cs
> mcs -r:LibraryDB.dll, System.Runtime.Remoting.dll LibraryServer.cs
```

### **Bearbeitung: bis zum 27.01.2006/20:00 Uhr**

Alle Dateien sollen im Verzeichnis `/proj/i4mw/loginname/aufgabe7/` abgelegt und mit dem `abgabe`-Programm abgegeben werden.

**Die Bearbeitung ist in 2er Gruppen möglich.**