

6 Übungsaufgabe #6: JXTA

6.1 Allgemeines

In dieser Aufgabe soll das aus den ersten Aufgaben bekannte Bibliothekssystem als *JXTA-Service* implementiert werden. Der Umfang der Applikation kann sich dabei auf die Basisfunktionalität (d.h. Medien registrieren, auflisten, ausleihen und zurückgeben) beschränken. Weitergehende Funktionalität (z.B. Primary/Secondary-Unterscheidung, Item-Duplizierung, -Migration, etc.) ist nicht vorgesehen. Die eigentliche Verwaltung der Medien erfolgt nach wie vor in einer Client/Server-Architektur (→ Code wiederverwenden!). Für das Lokalisieren und Binden eines Servers werden nun jedoch Peer-to-Peer-Mechanismen der JXTA-Plattform verwendet.

Die Kommunikation zwischen Frontend und Server erfolgt über Nachrichten und mittels eines einfachen zeichenbasierten Protokolls. Die Nachrichtenobjekte werden hierfür mittels Instanzen der Klasse *StringMessageElement* befüllt. Das Protokoll sieht folgende Nachrichten vor:

- Registrieren, Ausleihen und Rückgabe (→ <operation>) des Mediums <title>:
Anfrage: [Tag: operation; Value: <operation>],
[Tag: title ; Value: <title>]
Antwort: [Tag: status ; Value: success | failed]
- Auflisten aller registrierten Medien (→ die Antwort kann eine beliebige Anzahl an *title*-Tags beinhalten):
Anfrage: [Tag: operation; Value: list]
Antwort: [Tag: title ; Value: <title>], [...], ...

6.2 Server

Um einen Server im Peer-to-Peer-Netzwerk bekannt zu machen, muss dieser eigene Advertisements veröffentlichen, die auf den Bibliotheks-Service hinweisen. Pro Server sind dazu ein *ModuleClassAdvertisement (MCA)* sowie ein *ModuleSpecAdvertisement (MSA)* bereitzustellen. Der Name des MCA soll „JXTAMOD:LibraryService“ lauten. Das MSA ist mit folgenden Angaben zu versehen:

```
Name: LibraryService
Version: 1.0
Creator: <login>
SpecURI: http://www4.informatik.uni-erlangen.de/Lehre/WS09/V_MW/Uebung/aufgaben/a6.pdf
```

Vervollständigt wird das MSA mit einem *Pipe-Advertisement (PA)*. Hierzu ist die Klasse *PipeAdvertisementCreator* zu implementieren, die ein PA erzeugt und in eine Datei schreibt (vgl. *pipe.adv* im Pub-Verzeichnis). Diese kann dann durch den Server ausgelesen und das PA an eine *JxtaServerPipe* gebunden werden, die auf ankommende Verbindungen wartet. Abschließend sind das MCA sowie das MSA per *DiscoveryService* zu publizieren.

Aufgaben:

- Implementierung der Klasse *PipeAdvertisementCreator* zur Erzeugung eines *Pipe-Advertisement*
- Implementierung des Bibliotheks-Server

Hinweise:

- Die IDs des MCA und des MSA sollen konstant bleiben (→ einmal erzeugen und hartcodieren).
- Die Aufgabenstellung verwendet Advertisements konzeptionell nicht völlig richtig. Dies geschieht aus Gründen des Aufwands und des praktischen Nutzens.

6.3 Frontend

Bei der Initialisierung verbindet sich das Frontend mit der *Net-Peer-Group* von JXTA. Der *Discovery-Service* dieser Gruppe wird dazu verwendet MSAs zu finden, die einen Dienst mit dem Namen „LibraryService“ beschreiben. Die dazu nötige Anfrage soll das Format [Tag: name; Value: LibraryService] aufweisen. Der Aufruf der Discovery-Prozedur erfolgt über ein Kommando *servers*, das alle gefundenen Server auflistet. Das Frontend soll darüber hinaus ein Kommando *connect* anbieten, das es ermöglicht, eine Verbindung mit einem der gefundenen Server-Instanzen aufzubauen. Zu diesem Zweck muss das PA aus dem ausgewählten MSA extrahiert und anschließend an eine bidirektionale *JxtaBiDiPipe* gebunden werden.

Aufgaben:

- Implementierung des Bibliotheks-Frontend

6.4 Abgabe: am 28.01.2010 in der Rechnerübung