

Aufgabenblatt 3: CORBA

(Zu bearbeiten bis 09.12.2019)

Client/Server-Programmierung

Wintersemester 2019/20

Aufgabe 1: CORBA Hello World (zur Vorbereitung)

Laden Sie sich zunächst den Code zum Aufgabenblatt von der CSP-Webseite herunter, entpacken Sie das Archiv und wechseln Sie ins Verzeichnis HELLO.

Dort befinden sich die drei Dateien `Hello.idl`, `HelloClient.java` sowie `HelloServer.java`. Ausserdem finden Sie in dem Archiv auch ein HowTo, das Ihnen die notwendigen und allgemeinen Schritte zur Entwicklung einer solchen Anwendung detailliert erklärt. Zu Beginn genügt es, den Client, den Server und den ORB-Daemon (bzw. Name-Server) auf einem Rechner laufen zu lassen. Testen Sie danach die Anwendung, wenn der Client, der Server und der ORB-Daemon (bzw. Name-Server) auf verschiedenen Rechnern ausgeführt werden. Verwenden Sie anstelle des JDK-ORBs auch den JacORB. Bei der Verwendung des JacORB müssen Sie folgendes beachten:

- Einrichten des JacORB: Siehe Vorlesung, Ende des Abschnitts 3.3
- Start Name-Servers: `ns`
- Start des Servers: `jaco HelloServer`
- Start des Clients: `jaco HelloClient`

Starten Sie den Client mehrfach hintereinander, bzw. starten Sie mehrere Clients. Was fällt Ihnen auf?

Aufgabe 2: Factories mit CORBA (zur Vorbereitung)

Laden Sie sich ggf. zunächst den Code zum Aufgabenblatt von der CSP-Webseite herunter, entpacken Sie das Archiv und wechseln Sie ins Verzeichnis HELLO_FACTORY. Dort finden Sie ein erweitertes Beispiel, bei dem jeder Client ein eigenes Objekt auf dem Server erzeugt. Dazu wird eine sog. Factory-Klasse verwendet und die IDL-Datei wie folgt erweitert:

```
module HelloWorld
{
    interface Hello
    {
        string sayHello();
    };
    interface HelloFactory
    {
        Hello createHello(in string Name);
    };
};
```

Die Methode `createHello()` liefert das Objekt, welches vom Client auf dem Server erzeugt wurde, an den Client. Die Methode erzeugt dazu einen Servant der Klasse `HelloServant`, registriert diesen beim Root-POA und gibt die Java-Referenz auf das CORBA-Objekt als Ergebnis zurück. Die Klasse `HelloServant` ist dabei von `HelloPOA` abgeleitet.

Laden Sie sich das Beispiel herunter, übersetzen und starten Sie es, sehen Sie sich die Methode `createHello()` genau an und versuchen Sie, den Code nachzuvollziehen.

Aufgabe 3: Börsenanwendung mit CORBA

Entwickeln Sie die Börsenanwendung zu einer CORBA-Anwendung.

- a) Implementieren Sie zuerst alle Dienste der Klasse `StockExchange` ausser `openDepot`.

Hinweis: Wenn Sie die Börsenanwendung in Aufgabe 2 von Blatt 2 um die Methode `getWinner()` erweitert haben, sollten Sie die Klasse `WinnerInfo` durch ein CORBA-struct realisieren, und die Exception-Klasse `StockException` durch eine CORBA-exception.

- b) Realisieren Sie die Server-Anwendung, die den Servant für `StockExchange` erzeugt und beim Namensdienst registriert.

Passen Sie Ihren Client an und testen Sie die Anwendung.

- c) Erweitern Sie nun die Anwendung mit dem Dienst `openDepot()` und den Diensten der `Depot`-Klasse. Verwenden Sie dazu die Vorgehensweise mit der `Factory`-Klasse aus Aufgabe 2 dieses Übungsblattes.

Anstelle der Klasse `DepotEntry` sollten Sie ein CORBA-struct verwenden.

Passen Sie Ihren Client an und testen Sie die Anwendung erneut.

Aufgabe 4: CORBA IOR (für Interessierte)

Verändern Sie das Beispiel aus Aufgabe 2 dahingehend, dass die Methode `createHello()` eine *Interoperable Object Reference* (IOR) als Zeichenkette an den Client liefert. Die IDL-Datei sieht dazu wie folgt aus:

```
module HelloWorld
{
    interface Hello
    {
        string sayHello();
    };
    interface HelloFactory
    {
        string createHello(in string Name);
    };
};
```

Hinweis: Eine IOR ist eine standardisierte Darstellung einer entfernte Objektreferenz, die auch in Form einer Zeichenkette repräsentiert werden kann. Diese enthält alle Informationen, die der ORB zur Kommunikation mit der Objektimplementierung benötigt. Serverseitig wird die ORB Methode `object_to_string()` verwendet, um die CORBA-Objektreferenz in einen IOR-String umzuwandeln. Clientseitig die Methode `string_to_object()`, um den IOR-String in eine CORBA-Objektreferenz zurück zu wandeln (vgl. Vorlesung, Abschnitt 3.4.7).