Übungsaufgabe #5: Callback-Mechanismus

28.05.2003

In Aufgabe 4 wurde eine Call-by-Value-Result-Semantik für primitive Datentypen implementiert. In dieser Aufgabe soll nun eine echte Call-by-Reference-Semantik für Objektreferenzen implementiert werden.

Call-by-Reference bei Objekten
Wenn man auf den Zustand eines Objekts nur mit Methodenaufrufen zugreift (und nicht mit direkten Zugriffen auf Membervariablen), lässt sich eine *Call-by-Reference*-Semantik mittels Fernaufrufe realisieren. Hierzu ist grundsätzliche folgendes zu tun:

- Auf Clientseite wird bei einem Fernaufruf, der eine Objektreferenz als Parameter besitzt, ein Skeleton für das entsprechende Objekt erzeugt und eine geeignete *Remotereferenz* (z.B. eigener Rechnername, Portnummer, Objekt-ID) zum Server geschickt.
- Empfängt der Server eine solche Referenz, so wird dort ein entsprechender Stub erzeugt und mit der Remotereferenz initialisiert.

Lösungsmöglichkeit: Erstellen Sie sich ein Objekt `RemoteReference`, welches die Remote-Adresse eines Objekts darstellt. Um die Adresse für ein Objekt zu erstellen, kann ein Konstruktor vorgesehen werden. Mit Hilfe dieses Konstruktors kann der Client bei einem Fernaufruf Adressen zu den übergebenen Parametern erzeugen. Um diese zu übertragen kann die Klasse `Message` um eine Methode erweitert werden:

```
bool writeReference(RemoteReference r);
```

Auf Seite des Empfängers muss ein `RemoteReference`-Objekt erzeugt werden. Auch hierzu kann eine neue Methode der Klasse `Message` vorgesehen werden:

```
bool readReference(RemoteReference &r);
```

Sie bekommt als Parameter eine Referenz auf ein `RemoteReference` Objekt, welches mit den empfangenen Daten initialisiert werden soll. Ein mögliches Interface für `RemoteReference` könnte wie folgt aussehen:

```
class RemoteReference {
    public:
            RemoteReference(Address addr, int oid);
            RemoteReference();
            setReference(Address addr, int oid);
            int getOID();
            Address getAddress();
};
```

Einfache Beispielanwendung
Der implementierte Callback-Mechanismus soll mit einer einfachen Musteranwendung getestet werden. Hierbei sollen der eigenen Kreativität keine Grenzen gesetzt werden. Ein Vorschlag wird auf der Webseite zur Übung gegeben:

`http://www4.informatik.uni-erlangen.de/Lehre/SS03/V_VS/Uebung/aufgaben/aufgabe5/`

Hinweis: Ihr RPC-System muss gleichzeitig als Client und als Server dienen können. (Einige Details hierzu finden Sie ebenfalls auf der Webseite zur Übung.)

Abgabe: bis 06.06.2003 12:00 Uhr

Abgabe mittels /proj/i4vs/pub/abgabe (Abgabe in 2er oder 3er Gruppen möglich)