

7° Esercitazione (da svolgere):

Java RMI: esempio callback

Si progetti un'applicazione Client/Server per la **gestione di un'asta** con un server che controlla lo svolgimento della stessa. In particolare, il **server** realizza una operazione per **l'immissione di nuove offerte** (bid) da parte dei client, mentre il **client** realizza due operazioni, una per la **ricezione delle offerte** immesse da altri client (metodo di callback), e un'altra per la **ricezione della notifica di terminazione dell'asta**.

Il **Server** è realizzato come un oggetto remoto. Esso implementa l'operazione di **immissione di nuove offerte** che accetta in ingresso *il riferimento all'oggetto client remoto e l'offerta*, e restituisce un risultato che indica *l'esito dell'operazione*. Il Server mantiene una struttura dati all'interno della quale memorizza *i riferimenti di tutti i client* che hanno effettuato almeno una offerta. L'asta è suddivisa in tre fasi:

1) **Apertura dell'asta**: durante questa prima fase il server, dopo aver ricevuto le prime 5 offerte, calcola il valore finale di vendita dell'oggetto all'asta aumentando di un ordine di grandezza la media delle 5 offerte (moltiplicando la media per 10).

- 2) **Esecuzione dell'asta**: il server accetta le varie offerte e le invia a tutti i client che stanno prendendo parte all'asta (memorizzati nella struttura dati).
- 3) **Terminazione dell'asta**: l'asta termina quando l'offerta di un client supera il prezzo di vendita deciso al punto 1. Una volta stabilita la fine dell'asta, il server notifica la fine dell'asta a tutti i partecipanti (invocando l'opportuna operazione).

Ogni **Client** è realizzato come **processo ciclico** che continua a fare richieste di offerta sincrone fino alla fine dell'asta (comunicata attraverso l'invocazione dell'operazione di notifica di fine asta). E' la notifica di fine dell'asta che fa terminare i diversi clienti che hanno fatto offerte.

Inoltre, il client realizza i metodi remoti invocati dal server:

- **Ricezione delle offerte**: metodo di callback; accetta come unico parametro di ingresso *l'offerta attuale* e la stampa a video.
- **Notifica della terminazione dell'asta**: metodo invocato dal server alla terminazione dell'asta, accetta come unico parametro di ingresso *l'offerta finale di chiusura dell'asta* e provoca la terminazione del processo client.

Proposta di estensione

Scrittura e notifica modifiche su un file remoto

Si vuole sviluppare un'applicazione C/S basata su RMI e sul meccanismo della callback per la modifica di un file presente sul server remoto (la decisione di quale file condividere è del server) e la notifica dei cambiamenti effettuati sul file ad un gruppo di client interessati all'evento.

In particolare, il server mantiene una *struttura dati contenente tutti i client registrati* per il file e realizza le seguenti operazioni:

- **Sottoscrizione** dei client: accetta come parametro di ingresso *il riferimento al client remoto*, lo inserisce nella struttura dati, e restituisce un valore logico con *l'esito dell'operazione*.
- **De-sottoscrizione** dei client: accetta come parametro di ingresso *il riferimento al client remoto*, lo elimina dalla struttura dati, e restituisce un valore logico con *l'esito dell'operazione*.
- **Write**: accetta in ingresso *la posizione* (all'interno del file) dalla quale si vuole effettuare la modifica e *l'array di byte* che vanno sovrascritti sul server remoto a partire dalla posizione indicata, effettua la scrittura richiesta, e restituisce *l'esito dell'operazione*, true in caso di successo, false altrimenti, ad esempio se il file non esiste.

Il **Client** è realizzato come un **processo ciclico**. Dapprima il client si *sottoscrive* presso il server, quindi, *effettua una serie di richieste di scrittura sincrone* fino a che l'utente non decide la terminazione della sessione di lavoro (fine del file di input al processo client), infine, si *de-sottoscrive* e termina. Inoltre, il client deve realizzare il metodo invocato dal server remoto per la ricezione delle modifiche effettuate sul file:

- **Notifica modifiche**: metodo di callback invocato dal server; accetta in ingresso *la posizione* da cui parte la modifica (all'interno del file) e *l'array di byte* contenente i byte sostituiti, quindi li stampa a video.

Si estenda il tutto a **più file** esportati e visibili dall'esterno.

Si estenda a **più direttori** esportati.

Si estenda a **più file system** esportati.

Consegna

Potete inviare via email l'estensione ai docenti, per discutere la soluzione ed eventualmente pubblicarla sul sito del corso