

Reti di Calcolatori L-A

Prova d'esame

5/12/2006

Cognome:
 Nome:
 Matricola:

Tempo a disposizione: 2h

E' obbligatorio mettere Cognome Nome Matricola e Numero Compito all'inizio di ogni file sorgente, pena la non valutazione del compito, che viene stampato in modo automatico, solo in caso siano presenti gli elementi detti sopra.

Si devono consegnare **tutti i file sorgente e tutti gli eseguibili prodotti singolarmente** (per favore, solo quelli relativi ai file sorgente consegnati!!).

La prova intende valutare le capacità progettuali e di programmazione sia in **ambiente Java** che in **ambiente C**, pertanto è consigliabile sviluppare, *almeno in parte*, **entrambe** le soluzioni richieste. *In entrambi gli esercizi, sia in Java che in C, si effettuino gli opportuni controlli sui parametri della richiesta e si gestiscano le eccezioni, tenendo presente i criteri secondo cui si possa ripristinare il funzionamento del programma oppure si debba forzarne la terminazione.*

Leggete con attenzione le specifiche del problema prima di impegnarvi "a testa bassa" nello sviluppo delle singole parti. Naturalmente, i componenti da consegnare devono essere stati provati.

Si richiede il progetto della gestione dei servizi **GiocoDiRuolo**, per utenti che vogliono realizzare giochi di ruolo distribuiti. Obiettivo è l'insieme delle funzionalità di supporto per il provider dei servizi e non la realizzazione delle funzioni di gioco vere e proprie.

I servizi di GiocoDiRuolo sono organizzati in **livelli** (identificati da un intero crescente) e, per ciascun livello, si prevedono al massimo un certo numero di **utenti** e di **risorse**. Un utente si può iscrivere al più ad un livello e poi può eliminare o aggiungere risorse.

Si vogliono realizzare le funzionalità di gestione:

- aggiunta di una risorsa:** questa operazione richiede il numero del livello, il nome della risorsa e il nome dell'utente, e, se l'utente è fra quelli registrati nel livello, aggiunge la risorsa;
- eliminazione di un utente:** questa operazione richiede il nome dell'utente e scorre la struttura dati eliminando l'utente dal livello in cui si trova;
- visualizzazione dello stato di un livello:** questa operazione richiede il numero del livello e ne visualizza lo stato stampando a video l'elenco delle risorse e degli utenti;
- eliminazione di una risorsa:** questa operazione richiede il nome della risorsa, scorre tutti i livelli eliminando la risorsa, e restituisce la lista di tutti (e soli) i livelli liberati.

Si progetti con particolare attenzione la **struttura dati** che mantiene lo stato degli N livelli ('L', per libero a default) per al massimo di K utenti e M risorse, da implementare opportunamente nei diversi ambienti richiesti, Java e C.

Livello	Utente 1	Utente 2	...	Utente K	Risorsa 1	Risorsa 2	...	Risorsa M
1	Aragorn	L	...	Aron	Borraccia	L	...	Arco
2	Frodo	Pipino	...	L	Anello	Faretra	...	L
3	L	L	...	L	L	L	...	L
...
N	L	Gandalf	...	L	Bastone	L	...	L

Parte Java

Utilizzando **java RMI** sviluppare un'applicazione C/S che consenta di effettuare le operazioni remote per:

- aggiungere un risorsa alla struttura dati;
- eliminare un utente dalla struttura dati.

Il progetto RMI si basa su:

- un'interfaccia (**RemOp**, contenuta nel file *RemOp.java*) in cui vengono definiti i metodi invocabili in remoto dal client:

Il metodo **aggiungi_risorsa** accetta come parametri d'ingresso l'identificatore del livello, il nome della risorsa e il nome dell'utente e, se l'utente è fra quelli registrati nel livello, aggiunge la risorsa; quindi restituisce l'esito dell'operazione, 0 se la registrazione è andata a buon fine, -1 altrimenti, ad esempio, se il livello non è presente nella struttura dati oppure se l'utente non è presente nel livello.

Il metodo **elimina_utente** accetta come parametro d'ingresso il nome dell'utente e restituisce l'esito dell'operazione, 0 in caso di successo, -1 altrimenti, ad esempio, se l'utente non è presente nella struttura dati.

Si progettino inoltre le classi:

- Server** (contenuta nel file *Server.java*), che implementa i metodi del server invocabili in remoto;
- Client** (contenuta nel file *Client.java*), che realizza l'interazione con l'utente proponendo ciclicamente i servizi che utilizzano i due metodi remoti, e stampa a video i risultati, fino alla fine del file di input da tastiera.

Parte C

Usando **socket stream e un'unica connessione**, sviluppare un'applicazione C/S che consenta le operazioni remote per:

- visualizzare lo stato di un livello;
- eliminare una risorsa dalla struttura dati.

Più in dettaglio:

- il **client** è organizzato come un processo ciclico fino alla fine del file di input. Per ogni iterazione del ciclo chiede all'utente quale tipo di operazione vuole effettuare e realizza le interazioni col server utilizzando **una sola connessione**. Alla ricezione del fine file, libera opportunamente le risorse e termina. Per ogni richiesta ricevuta dall'utente, il client prima invia il tipo di servizio al server, poi gestisce gli invii e le ricezioni necessarie alla realizzazione dello specifico servizio.

Nel caso di **visualizzazione dello stato del livello**, il client richiede all'utente e invia al server l'identificativo del livello a cui è interessato, quindi riceve l'elenco di utenti e risorse e lo stampa a video.

Nel caso di **eliminazione della risorsa dalla struttura**, il client richiede all'utente e invia al server il nome della risorsa, quindi riceve la lista dei livelli da cui si è eliminata la risorsa e la stampa a video.

- il **server** è organizzato come un demone e gestisce in modo parallelo l'interazione col client generando, all'arrivo di ogni richiesta da un nuovo client, un figlio. Il figlio, con una prima lettura discrimina il tipo di funzionalità richiesto, poi gestisce opportunamente invii e ricezioni per l'operazione e si pone in attesa di nuove richieste dallo stesso client; alla lettura dell'EOF il figlio termina.

Per ogni richiesta di **visualizzazione dello stato del livello**, il figlio riceve l'identificativo del livello e invia al client l'elenco di tutti gli utenti e le risorse registrati nella riga corrispondente nella struttura dati.

Per ogni richiesta di **eliminazione della risorsa**, il figlio riceve il nome della risorsa, scorre tutti i livelli liberandoli dalla risorsa (riportando lo stato al valore di default 'L'), e restituisce al client l'elenco di tutti (e soli) i livelli liberati.