

# Reti di Calcolatori T

## Reti di Calcolatori L-A

### Appello del 10/01/2014

#### Compito 1

Cognome: .....  
Nome: .....  
Matricola: .....

Tempo a disposizione: 2h

È obbligatorio inserire Cognome, Nome, e numero di Matricola 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 consigliato sviluppare **entrambe** le soluzioni richieste al meglio.

**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, ci aspettiamo che i componenti da consegnare siano stati provati e siano funzionanti.**

\*\*\*\*\*

Si richiede il progetto della gestione dei servizi **TicketTwo**, per la gestione della compravendita di biglietti per eventi.

I servizi di TicketTwo mantengono, per ogni evento, le seguenti informazioni: la **descrizione** dell'evento, unico all'interno del sistema; il **tipo** di evento che può essere Concerto, Calcio, Formula1; la **data** dell'evento e il **luogo** dell'evento; la **disponibilità**, intesa come i biglietti ancora disponibili; e il **prezzo**, cioè un intero che indica il prezzo unitario per biglietto.

Si vogliono realizzare le funzionalità di gestione:

1. **inserimento di un nuovo evento**: questa operazione richiede la descrizione dell'evento, il tipo, la data e il luogo, la disponibilità e il prezzo, ed inserisce il prodotto nella struttura dati;
2. **visualizzazione di tutti gli eventi di un certo tipo in un determinato luogo**: questa operazione richiede tipo di evento e luogo e restituisce l'elenco di tutti gli eventi di quel tipo il cui luogo è uguale a quello indicato;
3. **visualizzazione di tutti gli eventi disponibili e con prezzo inferiore o uguale a un prezzo dato**: questa operazione richiede il prezzo massimo per biglietto, valuta gli eventi con prezzo accettabile e la disponibilità e li restituisce all'utente;
4. **acquisto di uno o più biglietti per un determinato evento**: questa operazione richiede la descrizione dell'evento e il numero di biglietti da acquistare, valuta se i biglietti sono disponibili e quindi aggiorna la struttura dati.

Si progetti con particolare attenzione la **struttura dati** che mantiene lo stato, fino ad un massimo di N prodotti (L, per libero a default), da implementare opportunamente nei diversi ambienti richiesti, Java e C.

Descrizione	Tipo	Data	Luogo	Disponibilità	Prezzo
L	L	L	L	L	L
String	Concerto	11/01/2014	Verona	40	40
Junentus-Inger	Calcio	03/05/2014	Torino	21	150
...	...	...	...	...	...
GP Bologna	Formula1	07/09/2014	Bologna	10	200
L	L	L	L	L	L

Si considerino e si segnalino le possibilità di interferenze fra le operazioni, evitandole dove necessario.

## Parte C

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

- inserimento di un nuovo evento;
- acquisto di uno o più biglietti per un determinato evento.

Il progetto RPC si basa su un'interfaccia (contenuta nel file *RPC\_xFile.x*) in cui vengono definiti i metodi invocabili in remoto dal client:

- La procedura **inserimento\_evento** accetta come parametro di ingresso la struttura dati **inserimento** contenente la descrizione dell'evento, il tipo, la data, il luogo, la sua disponibilità, il suo prezzo, e aggiunge l'evento alla struttura dati; quindi restituisce l'esito dell'operazione, 0 se la registrazione è andata a buon fine, -1 altrimenti, ad esempio, se la struttura dati è piena.
- La procedura **acquisto\_biglietti** accetta come parametro di ingresso una struttura con contenente la stringa che indica la descrizione dell'evento e un intero con il numero di biglietti da acquistare; quindi il server valuta la richiesta, controlla se esiste l'evento e il numero di biglietti richiesti e aggiorna la struttura restituendo l'esito dell'operazione, 0 se l'acquisto è andato a buon fine, -1 altrimenti, ad esempio, se l'evento non è presente nella struttura dati.

Si progettino inoltre i programmi:

- **RPC\_Server** (contenuta nel file *RPC\_Server.c*), che implementa i metodi del server invocabili in remoto;
- **RPC\_Client** (contenuta nel file *RPC\_Client.c*), il processo filtro 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 Java

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

- visualizzazione di tutti gli eventi di un certo tipo in un determinato luogo;
- visualizzazione di tutti gli eventi disponibili e con prezzo inferiore o uguale a un prezzo dato.

Più in dettaglio:

- Il **client** è organizzato come un processo ciclico fino alla fine dell'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 di tutti gli eventi di un certo tipo in un determinato luogo**, il client richiede all'utente e invia al server il tipo di evento e il luogo, quindi riceve l'elenco degli eventi disponibili e stampa a video.

Nel caso di **visualizzazione di tutti gli eventi disponibili e con prezzo inferiore o uguale a un prezzo dato**, il client richiede all'utente e invia al server la descrizione evento e il prezzo massimo dei biglietti; quindi riceve l'esito dell'operazione e lo 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 di tutti gli eventi di un certo tipo in un determinato luogo**, il figlio riceve il tipo e il luogo, scorre la struttura cercando tutti gli eventi di un certo tipo e luogo e restituisce l'elenco degli eventi trovate al client.

Per ogni richiesta **visualizzazione di tutti gli eventi disponibili e con prezzo inferiore o uguale a un prezzo dato**, il figlio riceve il prezzo massimo per biglietto, e restituisce una lista con tutti gli eventi disponibili con prezzo inferiore a quello richiesto.