

Reti di Calcolatori T

Reti di Calcolatori L-A

Appello del 09/01/2013

Compito 2

Cognome:
Nome:
Matricola:

Tempo a disposizione: 2h

È obbligatorio inserire 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 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 dei servizi **NoleggiaSci**, per la gestione del noleggio sci di un negozio di articoli sportivi. I servizi di NoleggiaSci mantengono, per ogni paio di sci, le seguenti informazioni: un **identificatore**, unico all'interno del sistema; il **modello** dello sci; per ogni sci noleggiato riporta **giorno, mese e anno** di inizio noleggio (per semplicità si ipotizzano tutti mesi da 30 giorni) e il **numero di giorni** di noleggio; il **costo giornaliero di noleggio** (la durata minima del noleggio è un giorno).

Si vogliono realizzare le funzionalità di gestione:

1. **inserimento di un nuovo paio di sci**: questa operazione richiede l'identificatore, il modello e il costo giornaliero, quindi inserisce lo sci nella struttura dati e restituisce l'esito dell'operazione;
2. **richiesta di noleggio di un paio di sci**: questa operazione richiede l'identificatore, il giorno, mese e anno di inizio noleggio e la durata in giorni del noleggio, quindi aggiorna la struttura dati e restituisce l'esito dell'operazione;
3. **visualizzazione degli sci noleggiabili di un certo modello**: questa operazione richiede il modello, quindi restituisce la lista dei gli sci noleggiabili per tale modello;
4. **valutazione costo di un noleggio**: questa operazione richiede l'identificatore e restituisce il costo complessivo del noleggio per lo sci richiesto (per tutta la durata del noleggio).

Si progetti con particolare attenzione la **struttura dati** che mantiene lo stato, fino ad un massimo di K sci (L e -1 – o anche -1/-1/-1 – per libero o non presente a default), da implementare e inizializzare nei diversi ambienti richiesti, Java e C.

Identificatore	Data	Giorni	Modello	Costo giornaliero
L	-1/-1/-1	-1	-1	-1
X12AB	12/12/2012	14	Volki Shark	7
ACD14	-1/-1/-1	-1	Fisher Snow	5
Y23CC	23/12/2012	7	Volki Shark	7
...
L	-1/-1/-1	-1	-1	-1

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:

- inserire un nuovo paio di sci;
- richiedere il noleggio di un paio di sci.

Il progetto RPC si basa su un'interfaccia (contenuta nel file *RPC_xFile.x*) in cui vengono definite le procedure invocabili in remoto dal client:

- La procedura **inserisci_sci** accetta come parametro d'ingresso una struttura contenente l'identificatore, il modello e il costo giornaliero e inserisce il nuovo sci nella struttura dati restituendo l'esito dell'operazione, positivo se l'inserimento è andato a buon fine, negativo altrimenti, ad esempio se non c'è più spazio nella struttura dati.
- La procedura **noleggia_sci** accetta come parametro d'ingresso una struttura contenente l'identificatore, il giorno, mese e anno di inizio noleggio e la durata in giorni del noleggio, aggiorna la struttura dati e restituisce l'esito dell'operazione richiesta, positivo se l'operazione è andata a buon fine, negativo altrimenti, ad esempio se gli sci non sono presenti nella struttura dati o se sono già stati prenotati per una qualunque durata.

Si progettino inoltre:

- **RPC_Server** (contenuta nel file *RPC_Server.c*), che implementa le procedure del server invocabili in remoto;
- **RPC_Client** (contenuta nel file *RPC_Client.c*), il processo che realizza l'interazione con l'utente proponendo ciclicamente i servizi che utilizzano le due procedure remote, e stampa a video i risultati, fino alla fine del file di input da tastiera.

Parte Java

Sviluppare un'applicazione C/S basata su **socket stream** che realizzi le operazioni remote per:

- visualizzare gli sci noleggiabili di un certo modello;
- valutare il costo di un noleggio.

Si noti il vincolo di usare un'unica connessione per tutte le interazioni fra client e server.

Più in dettaglio:

- Il **client** è organizzato come un **processo filtro, ossia ciclico che esegue 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 per la intera sessione con il server**; 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 degli sci noleggiabili di un certo modello**, il client richiede all'utente e invia al server il modello, quindi riceve la lista di sci corrispondenti stampandoli a video. Nel caso di **valutazione del costo di un noleggio**, il client invia al server l'identificatore; quindi riceve l'esito dell'operazione e lo stampa a video.
- Il **server** è organizzato come un **unico processo che gestisce in modo parallelo** l'interazione col client generando un figlio all'arrivo di ogni richiesta da un nuovo client. Il figlio che serve quella sessione, 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 della fine sessione, il figlio termina. Per ogni richiesta di **visualizzazione degli sci noleggiabili di un certo modello**, il figlio riceve il modello e trasmette al client la lista degli sci corrispondenti a tale modello e non già noleggiati. Per ogni richiesta di **valutazione del costo di un noleggio**, il figlio riceve l'identificatore e restituisce al client il costo totale del noleggio per tutte le giornate: il costo totale del noleggio nel caso in cui lo sci venga identificato e sia effettivamente noleggiato, oppure un esito negativo, ad esempio se lo sci non è stato noleggiato oppure se non è presente nella struttura dati.