

Laurea Specialistica in Informatica per la Gestione d'Azienda A.A. 2007-2008
Tecnologie Informatiche per la Gestione Aziendale
Appello del 04 febbraio 2008

In un sistema informativo distribuito, un server remoto offre il servizio di calcolo di un intervallo temporale, che viene usato dai client per calcolare la durata di certe operazioni.

Il candidato progetti *il time server* e lo realizzi nel linguaggio Java come oggetto remoto (e relativo server). Il time server implementa le seguenti operazioni dell'interfaccia **TimeInterval**:

- **void begin(int clientid)** richiede al server di memorizzare il tempo di sistema attuale (t_{start}), e considerarlo come istante di inizio dell'operazione. Se un client richiede più volte di seguito l'esecuzione di questo metodo, senza che prima richieda l'esecuzione del metodo **time()**, il server considera buona la prima richiesta ignorando le successive.
- **void end(int clientid)** di memorizzare il tempo di sistema attuale (t_{end}), e considerarlo come istante di fine dell'operazione. Se un client richiede più volte di seguito l'esecuzione di questo metodo, senza che prima richieda l'esecuzione del metodo **time()**, il server considera buona l'ultima richiesta ignorando le precedenti.
- **int time(int clientid)** per mezzo della quale un client richiede il calcolo di $t_{end} - t_{start}$ ed il ritorno del risultato.

Le operazioni prendono in ingresso un intero che rappresenta l'identificatore del client, si supponga inoltre per semplicità che il numero di client è limitato ed è pari a 50.

Si suggerisce di utilizzare il metodo **System.currentTimeMillis()** per avere il tempo di sistema attuale.

Il candidato realizzi anche un cliente multi-thread per il testing del gestore completando il seguente scheletro:

```
public class Client {  
    // completare  
    public static void main(String[] args) {  
        int quanti = 50;  
        for(int i = 0 ; i < quanti ; i++){  
            new MyThread( ).start();  
        }  
    }  
}
```

```
class MyThread extends Thread {  
    public MyThread( ) {  
        // completare  
    }  
    public void run() {  
        // completare  
    }  
}
```

