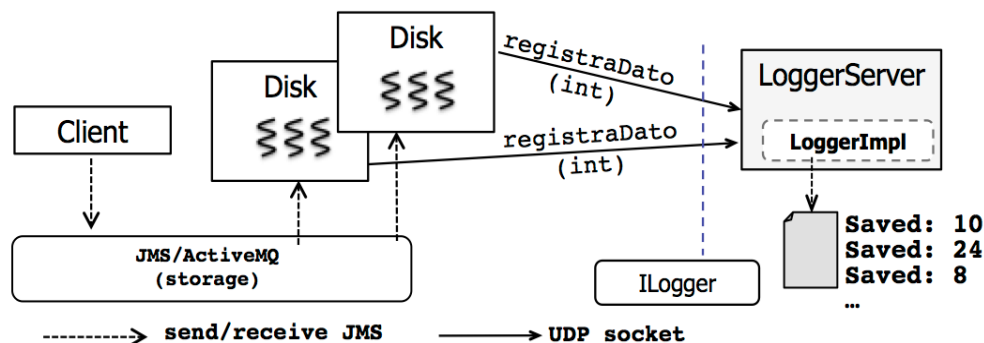


Università degli Studi di Napoli Federico II
Esame di Programmazione II
Prova pratica del giorno 24/07/2019
Durata della prova: 120 minuti

Cognome..... Nome..... Matr.....

Lo studente legga attentamente il testo e produca il programma ed i casi di test necessari per dimostrarne il funzionamento. La mancata compilazione dell'elaborato, la compilazione con errori o l'esecuzione errata dell'algoritmo daranno luogo alla valutazione come **prova non superata**.

Ricordarsi di indicare Cognome, Nome e matricola su questo stesso foglio, che dovrà essere in ogni caso consegnato alla Commissione. Al termine della prova lo studente dovrà far verificare il funzionamento del programma ad un membro della Commissione.



Il candidato realizzi un sistema di storage basato su **JMS** e **socket UDP**. Il sistema si compone di:

Client. Utilizzatore del sistema. Il Client richiede il salvataggio di un dato (ossia un `int` tra 0 e 100) ed indica la porta di “LoggerServer” (ossia l’entità –raggiungibile tramite socket UDP– responsabile in ultima istanza della memorizzazione del dato su file, come descritto nel seguito) mostrato in Figura. Il Client richiede un salvataggio tramite l’invio di un `MapMessage` JMS contenente 1) il dato da salvare e 2) porta del `LoggerServer`, entrambi specificati da prompt e quindi inseriti nel messaggio JMS. Per es.:

```
java nomepackage.Client 25 3000
```

Disk. Implementa un meccanismo di **ricezione asincrona**. Alla ricezione di un `MapMessage` dal Client, il *listener* dei messaggi di `Disk` avvia un thread che estrae dal messaggio *dato* e *porta*. Il thread 1) mostra a video il dato, e 2) invoca il metodo `void registraDato(int dato)` specificato dall’interfaccia **ILogger** fornita dal `LoggerServer`. `Disk` invoca `registraDato` di `LoggerServer` tramite un proxy UDP.

LoggerServer. Registra su file i dati passati da `Disk`. Esso è un server multithread su socket UDP basato su Proxy-Skeleton. Espone l’interfaccia `ILogger` –come in Figura– con il relativo metodo `void registraDato(int dato)`, che mostra il dato a video e lo salva su file. Il metodo `registraDato` è eseguito in mutua esclusione. `LoggerServer` è avviato da prompt, specificando la porta come nell’esempio:

```
java nomepackage.LoggerServer 3000
```

Lo studente produca l’interfaccia `ILogger` e le classi richieste dal pattern Proxy-Skeleton. Il sistema sarà testato da prompt con #1 Client, #2 Disk, #1 LoggerServer. Proprietà naming-JNDI:

```
org.apache.activemq.jndi.ActiveMQInitialContextFactory tcp://127.0.0.1:61616
```

Il candidato indichi la modalità di svolgimento richiesta dalla Commissione tra:

☐ Queue / skeleton per delega

☐ Topic / skeleton per ereditarietà