

Note sullo svolgimento della prova

- Non è consentito l'uso di vecchi testi d'esame, libri, o appunti.
- Utilizzare NetBeans o Notepad++ come editor, e finestre DOS per compilare ed eseguire i programmi.
- La cartella C:\prg contiene il JDK da utilizzare e la documentazione delle Java API. Eventuali file ausiliari preparati dal docente devono essere scaricati da elearn.ing.unipi.it
- I file sorgenti relativi all'esercizio 1 devono essere posti nella cartella C:\prg\myapps\esercizio1 mentre quelli relativi all'esercizio 2 devono essere posti nella cartella C:\prg\myapps\esercizio2.
- Al termine della prova, dopo la sottomissione degli elaborati su elearn.ing.unipi.it, il docente mostrerà una possibile soluzione. Dopo aver visto la soluzione, gli studenti avranno la possibilità di riprendere il proprio elaborato.

Esercizio 1

Un sistema per chattare è organizzato a stanze. Ogni stanza consente a due utenti di comunicare scambiandosi messaggi. Il server che gestisce il sistema genera una nuova stanza ogni volta che è necessario. Le stanze non più utilizzate da una coppia di utenti non vengono riutilizzate.

Realizzare una classe *Server* dotata almeno dei seguenti metodi:

- *Room connect()*: restituisce all'utente il riferimento alla stanza da usare per chattare; il metodo restituisce il riferimento ad una nuova stanza ad ogni nuova coppia di utenti (il primo e il secondo utente ottengono un riferimento ad una stanza, il terzo e il quarto a un'altra stanza e così via).

Realizzare una classe *Room* dotata almeno dei seguenti metodi:

- *void enter()*: entra nella stanza; il primo utente che invoca il metodo *enter()* su una stanza si blocca e viene sbloccato dall'esecuzione del metodo *enter()* da parte del secondo utente (in pratica attraverso l'esecuzione di questo metodo i due utenti di una stanza eseguono un rendez-vous).
- *void post(String msg) throws AloneInRoomException*: invia nella stanza il messaggio *msg*; tale messaggio viene conservato all'interno della stanza; il metodo lancia *AloneInRoomException* nel caso in cui l'utente che invoca il metodo sia l'unico utente dentro stanza (realizzare anche l'eccezione).
- *void leave()*: esce dalla stanza.
- *String toString()*: restituisce una stringa che contiene tutti i messaggi scambiati nella stanza. Esempio:

```
Ciao, come stai?  
Bene e tu?  
Anche io bene.  
...
```

Realizzare una classe *User* che

- ottiene dal server il riferimento alla stanza da usare;
- esegue *enter()*;
- invia dieci messaggi;
- esegue *leave()*.

Esercizio 2

Un sistema di *skill search* consente la ricerca di competenze specifiche, attraverso la consultazione di agenzie del lavoro, secondo un dato protocollo. Si consideri un numero di *agenzie del lavoro* non noto a priori, in comunicazione attraverso una configurazione ad anello. Il protocollo inizia con una *RichiestaDiCompetenze*, creata e inviata da una *AgenziaDelLavoro* (richiedente) alla successiva (ricevente). La *RichiestaDiCompetenze* è composta dalla descrizione delle competenze richieste, dalle porte di ascolto delle agenzie richiedente e proponente (quest'ultima stabilita nel primo giro di richiesta). Ogni agenzia ricevente, dopo aver visualizzato la richiesta, la invierà all'agenzia successiva se non ha una proposta, e così via. Non appena una agenzia ricevente ha una proposta, inserisce la propria porta di ascolto nell'oggetto *RichiestaDiCompetenze* e invia il medesimo all'agenzia richiedente invece che all'agenzia successiva. Quando l'agenzia richiedente riceverà la richiesta, la visualizzerà e la rimanderà all'agenzia ad essa successiva, e così via per il secondo giro. Nel secondo giro ogni agenzia riceve l'oggetto, lo visualizza, lo inoltra all'agenzia successiva e termina. Il secondo giro si ferma quando l'agenzia richiedente riceve nuovamente l'oggetto.

Si realizzi un'applicazione Java distribuita composta dalle classi *RichiestaDiCompetenze* e *AgenziaDelLavoro*. La classe *RichiestaDiCompetenze* mantiene ed elabora i dati suddetti, e raccoglie le offerte. La classe *AgenziaDelLavoro* svolge le operazioni di trasmissione di istanze di *RichiestaDiCompetenze* tra una agenzia e la successiva, in accordo al protocollo di cui sopra. La classe *AgenziaDelLavoro* svolge le operazioni di invio, ricezione e visualizzazione di istanze di *RichiestaDiCompetenze* esclusivamente in formato XML, in accordo al protocollo di cui sopra. Si assuma che le visualizzazioni di qualsiasi messaggio o contenuto non avvengano su console, ma tramite archiviazione su un database condiviso, visualizzato tramite MySQL Client. **Attenersi esattamente a quanto espresso dalle seguenti figure.** Fig.1 mostra una configurazione di test con tre agenzie. Fig.2 mostra lo schema del database e i file dell'applicazione. Fig.3 presenta le classi da realizzare (in grigio), con i relativi campi e i metodi da creare, i package e le classi da importare e usare. Gestire la chiusura di flussi e connessioni aperti. Catturare solo le eccezioni obbligatorie e gestirle semplicemente stampando le relative informazioni. Fig.4 mostra il file di testing *make.bat*, con tre casi di test. Si noti che il richiedente si distingue per avere un "?" e il proponente per avere un "!" nelle rispettive descrizioni. Fig.5 mostra la sequenza di passi (ad alto livello) che le istanze di *AgenziaDelLavoro* svolgono in un caso di test. La logica di *AgenziaDelLavoro* e *RichiestaDiCompetenze* dovrà essere valida a prescindere dai numeri di porta e dal numero di agenzie.

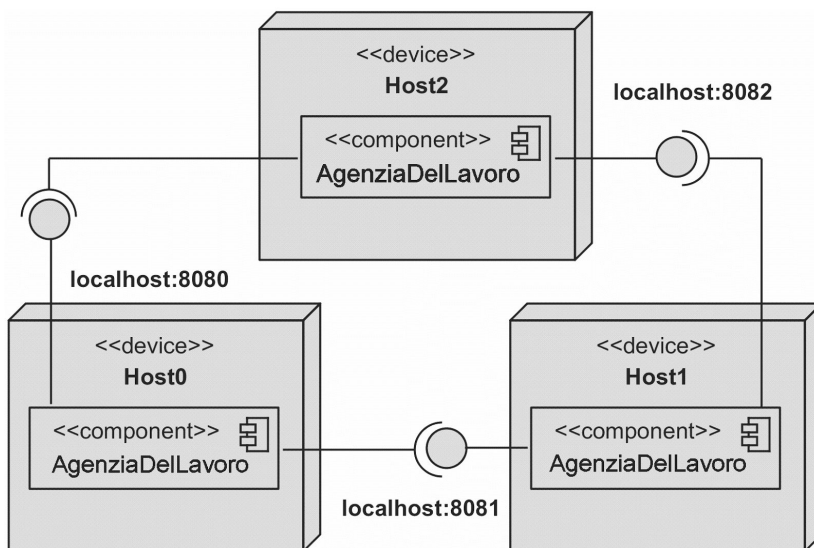


Fig.1 - Tre istanze di *AgenziaDelLavoro* e relative porte di ascolto

Column Name	Datatype
marcatemporale	VARCHAR(12)
idprocesso	INT(11)
operazione	VARCHAR(45)
risultato	TEXT

```
SELECT * FROM agenzia dellavoro.log;
```

marcatemporale	idprocesso	operazione	risultato
06:44:13.106	8080	inizio	*****...
06:44:13.497	8080	nuovaRichiesta	<Richiesta...
06:44:13.712	8080	invia	<Richiesta...
06:44:13.981	8081	ricevi	<Richiesta...

- AgenziaDelLavoro.class
- RichiestaDiCompetenze.class
- AgenziaDelLavoro.java
- RichiestaDiCompetenze.java
- make.bat

Fig.2 - Schema del database, tuple di esempio, e contenuto di c:\prg\myapps\esercizio2 di cui viene fornito solo make.bat

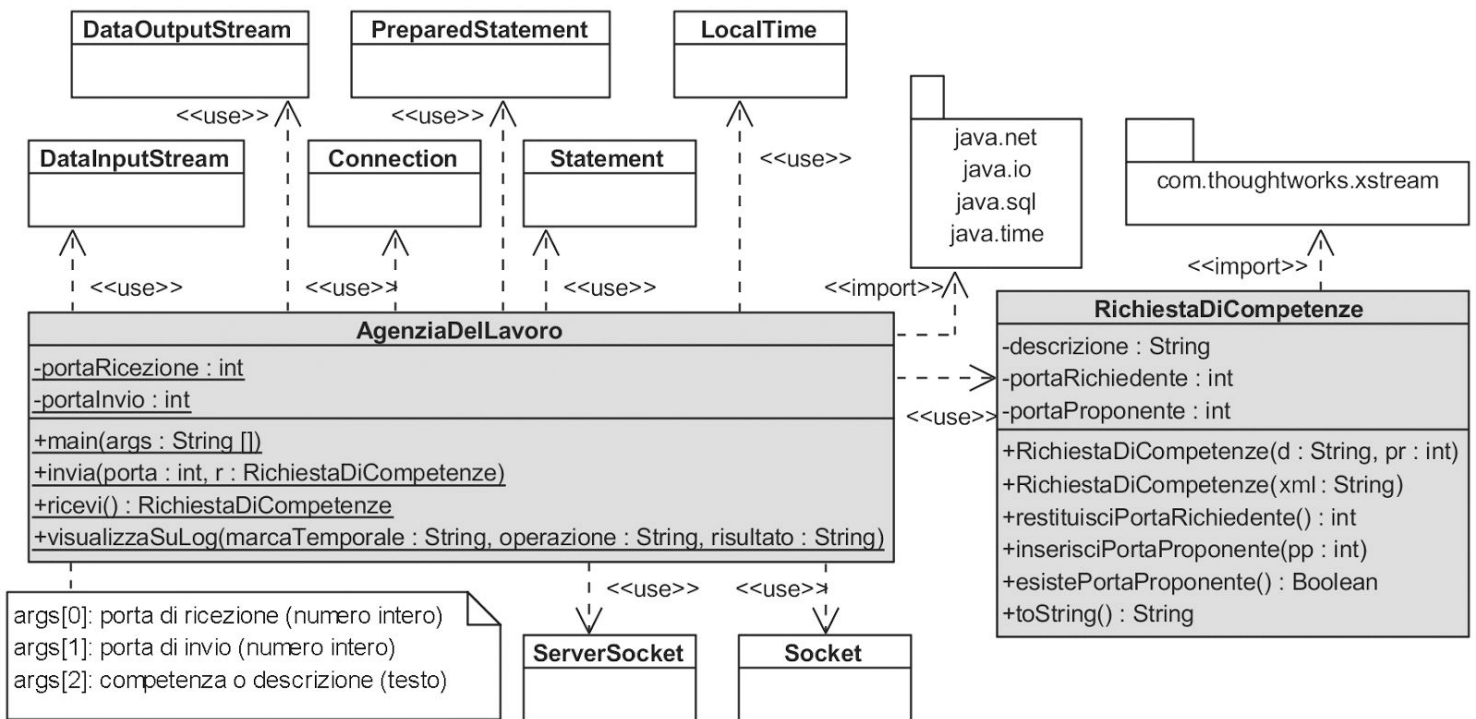


Fig.3 - Classi di cui deve essere composta l'applicazione (in grigio) e classi/package da usare (in bianco)

```
@echo off
c:\prg\jdk8\bin\javac -classpath c:\prg\libs\xstream-1.4.7.jar *.java
pause
set L=c:\prg\libs\
set
LIBS=%L%xstream-1.4.7.jar;%L$xmlpull-1.1.3.1.jar;%L%xpp3_min-1.1.4c.jar;%L:mysql-connector-java-5.1.34-bin.jar;

start cmd /k "color 2F && c:\prg\jdk8\bin\java -classpath %LIBS% AgenziaDelLavoro 8082 8080"
start cmd /k "color 2F && c:\prg\jdk8\bin\java -classpath %LIBS% AgenziaDelLavoro 8081 8082"
start cmd /k "color 2F && c:\prg\jdk8\bin\java -classpath %LIBS% AgenziaDelLavoro 8080 8081 "Programmatore C?"
rem competenza non disponibile, due giri completi di protocollo
pause

start cmd /k "color 3F && c:\prg\jdk8\bin\java -classpath %LIBS% AgenziaDelLavoro 8082 8080"
start cmd /k "color 3F && c:\prg\jdk8\bin\java -classpath %LIBS% AgenziaDelLavoro 8081 8082 "Programmatore C!"
start cmd /k "color 3F && c:\prg\jdk8\bin\java -classpath %LIBS% AgenziaDelLavoro 8080 8081 "Programmatore C?"
rem competenza disponibile in Agenzia1, al primo giro Agenzia2 non viene interpellata
pause

start cmd /k "color 4F && c:\prg\jdk8\bin\java -classpath %LIBS% AgenziaDelLavoro 8082 8080 "Programmatore C!"
start cmd /k "color 4F && c:\prg\jdk8\bin\java -classpath %LIBS% AgenziaDelLavoro 8081 8082"
start cmd /k "color 4F && c:\prg\jdk8\bin\java -classpath %LIBS% AgenziaDelLavoro 8080 8081 "Programmatore C?"
rem competenza disponibile in Agenzia2, due giri completi di protocollo
pause
```

Fig.4 - File make.bat, da non modificare.

NOTE SU MYSQL CLIENT

- Per poter rendere editabile la tabella *log* del database opinionista, in MySQL client:
Selezionare "Edit" > "Preferences" > "SQL Queries" > togliere la spunta da "Safe Updates"
Per svuotare il log ad ogni avvio: tasto destro sull'icona della tabella *log* > "Truncate Table"
- Per ordinare le colonne in ordine di inserimento, cliccare sull'icona gialla sopra la tabella "Reset all sorted columns"
- Per visualizzare il testo XML con indentazione cliccare sull'icona sopra la tabella accanto a "Wrap Cell Contents"
- Indirizzo per connettersi al database: "jdbc:mysql://localhost:3306/agenziadellavoro"

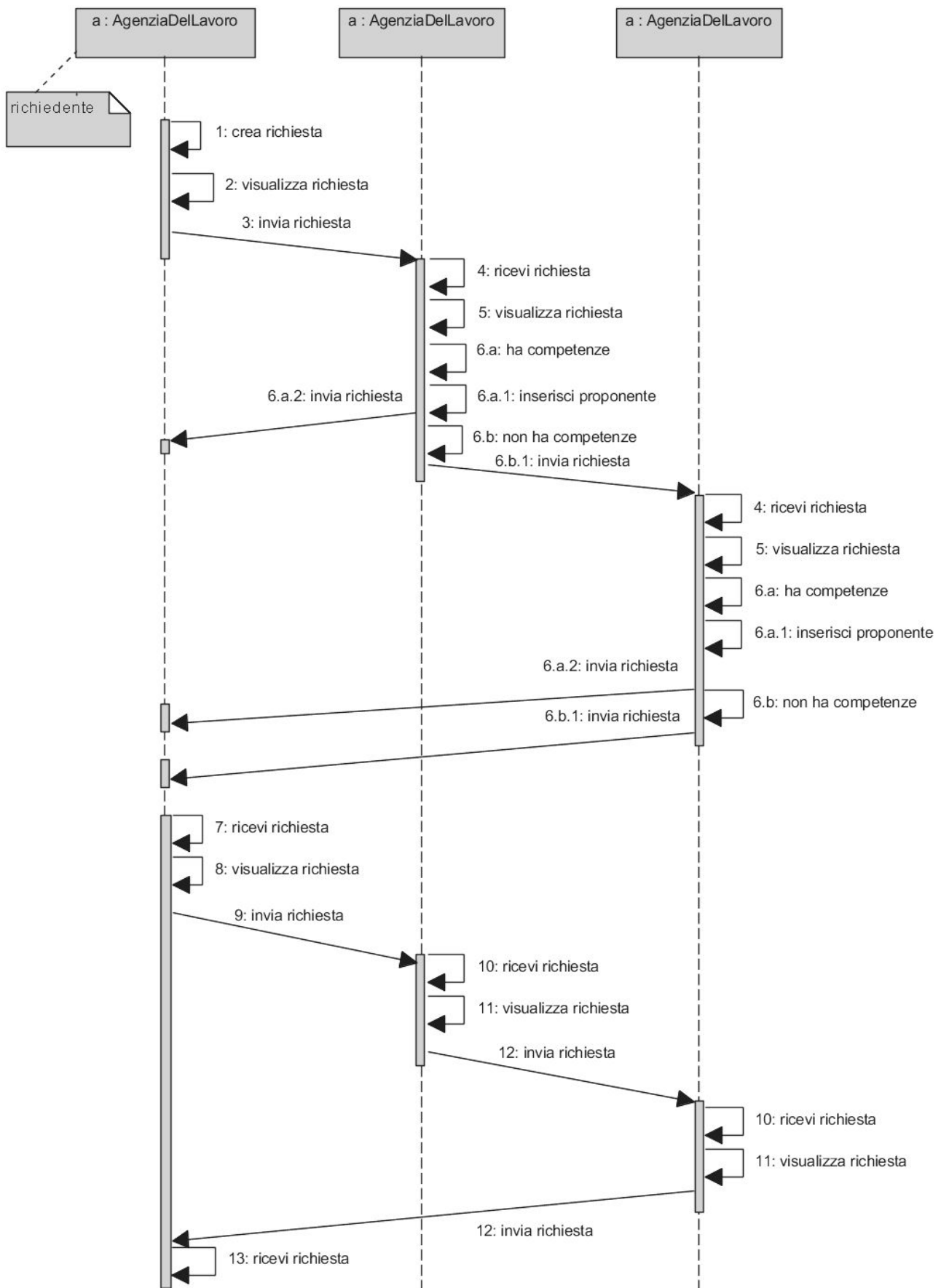


Fig.5 - Sequenza di passi che le istanze di *AgenziaDelLavoro* devono svolgere in un caso di test. Per brevità non viene rappresentata la classe *RichiestaDiCompetenze*.