Programmazione 17/02/2015

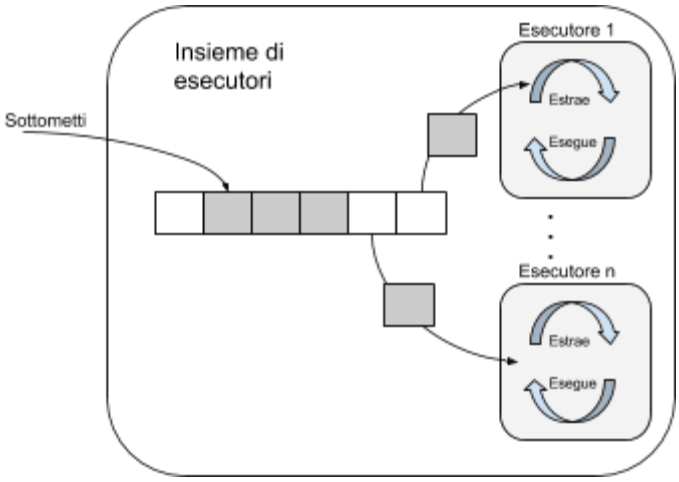
Note sullo svolgimento della prova

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

Esercizio 1

Un *insieme di esecutori* è un sistema che gestisce una coda di *compiti* e opera come segue:

- Quando l'insieme di esecutori viene creato vengono automaticamente generati un certo numero di thread (esecutori).
- Gli esecutori sono tutti equvalenti. Ogni esecutore esegue un ciclo infinito in cui preleva un compito dalla coda e lo esegue. Se nella coda non ci sono compiti gli esecutori si bloccano, in attesa che la coda diventi non vuota.
- E' possibile sottomettere un nuovo compito da svolgere inserendolo nella coda. Se la coda è piena l'operazione di inserimento è bloccante.



Supponiamo che il tipo compito sia definito come segue:

```
public interface Compito {
  public void esegui();
}
```

Realizzare una classe *InsiemeEsecutori* ed eventuali classi accessorie (per esempio la classe *Esecutore*) dotata almeno dei seguenti costruttori e metodi:

- *InsiemeEsecutori(int n, int s)*: crea una istanza in cui ci sono *n* esecutori attivi e la coda è in grado di contenere al più *s* compiti.
- *void sottometti(Compito c)*: inserisce nella coda il compito *c*. Il metodo è bloccante nel caso in cui la coda sia piena. Non appena un esecutore diventa libero, l'esecutore estrae il compito dalla coda e lo esegue.

Tralasciare il problema della terminazione degli esecutori.

Esercizio 2

Un gruppo di commercianti deve calcolare il prezzo medio delle forniture del settore, per consentire a ciascuno di valutare la convenienza del proprio fornitore rispetto alla media. Al fine di ridurre i rischi di spionaggio[1] il calcolo è svolto in modo distribuito e senza trasmettere i singoli prezzi: il primo commerciante genera un prezzo fittizio (random) e lo invia al secondo, che somma il suo prezzo e invia il totale al terzo, e cosi via fino a ritornare al primo, che sottrae il prezzo fittizio iniziale e somma il suo prezzo, dividendo tutto per il numero di commercianti. Si realizzi un'applicazione Java composta dalla classe *SommaPrezzi*, che calcola il valor medio in modo incrementale per un numero qualsiasi di commercianti, e la classe *Commerciante*, le cui istanze sono in grado di trasmettere e ricevere *SommaPrezzi* tra i commercianti, comunicanti secondo una configurazione ad anello. **Attenersi esattamente a quanto espresso dalle seguenti figure.** Fig.1 mostra la configurazione di test con soli tre commercianti (in generale assumere che le porte di ascolto partano dalla **8080** in su). Fig.2 mostra i file dell'applicazione. Fig.3 mostra il file di testing *make.bat*, che dovà produrre su console la stampa del valore 0.47666.... Fig.4 mostra la sequenza di passi (ad alto livello) che le istanze di *Commerciante* devono svolgere nel caso di test. Fig.5 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.

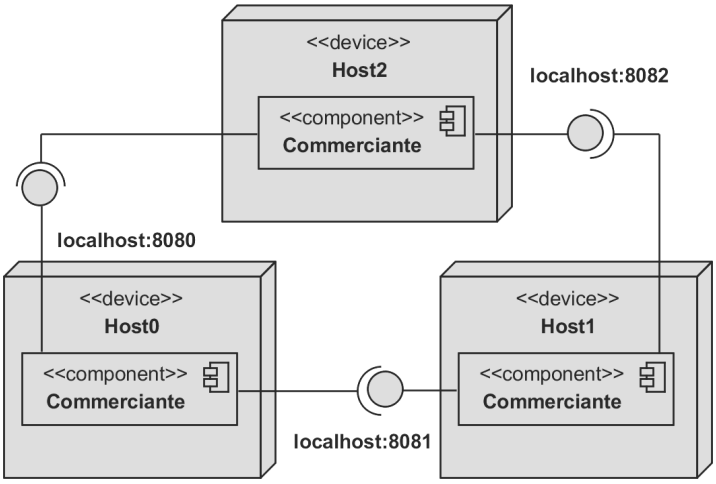
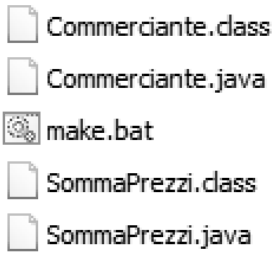


Fig.1 - Tre istanze di *Commerciante* e relative porte di ascolto



(viene fornito solo make.bat)

Fig.2 - Fig.2 - c:\prg\esercizio2

```
c:\prg\jdk8\bin\javac *.java
pause
start cmd /k "c:\prg\jdk8\bin\java Commerciante 8082 8080 .15"
pause
start cmd /k "c:\prg\jdk8\bin\java Commerciante 8081 8082 .81"
pause
start cmd /k "c:\prg\jdk8\bin\java Commerciante 8080 8081 .47"
pause
rem come risultato stampa 0.47666...
```

Fig.3 - File make.bat, da non modificare.

[1] Volto a conoscere il miglior fornitore.

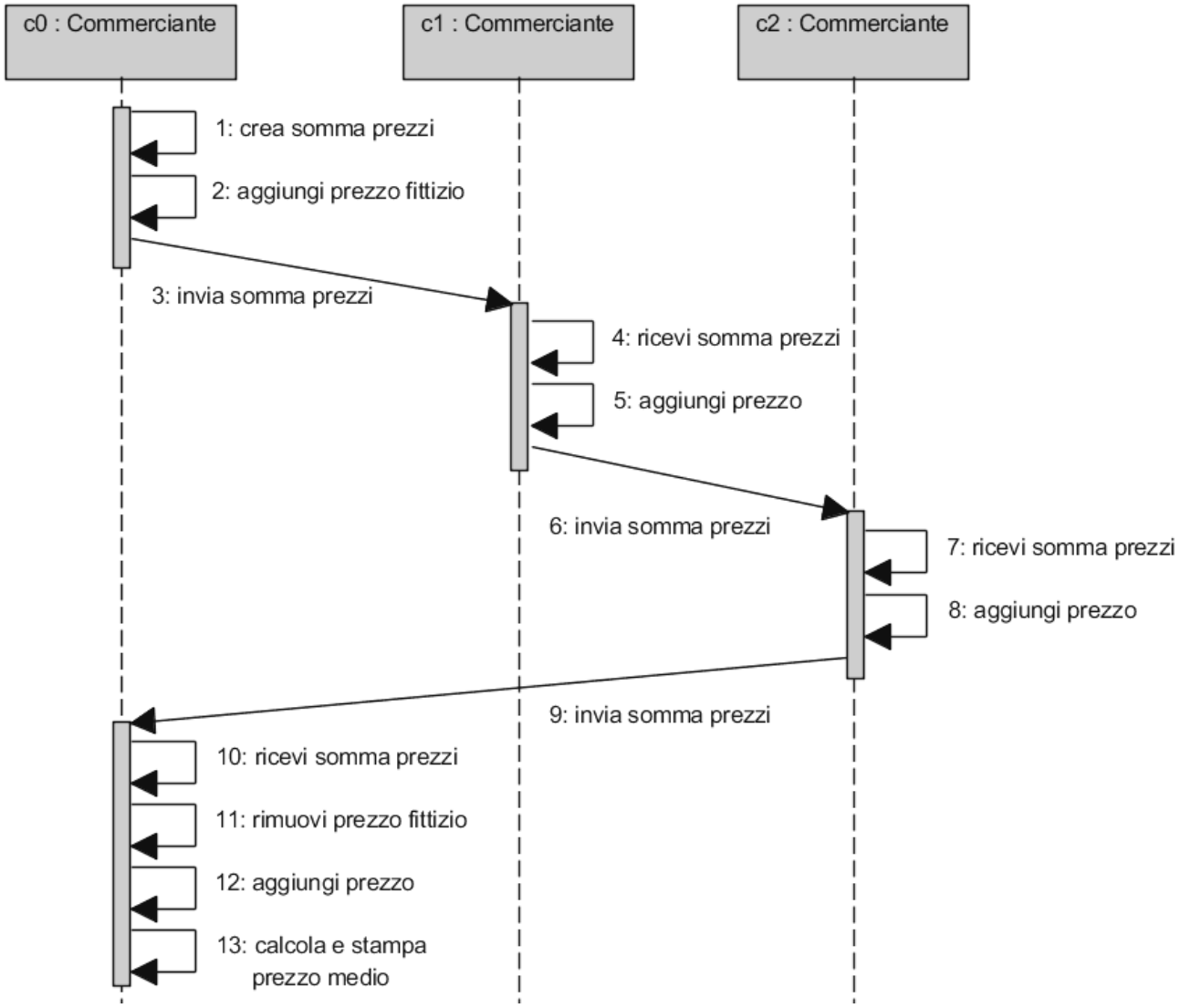


Fig.4 - Sequenza di passi che le istanze di *Commerciante* devono svolgere. Per brevità non viene rappresentata la classe *SommaPrezzi*.

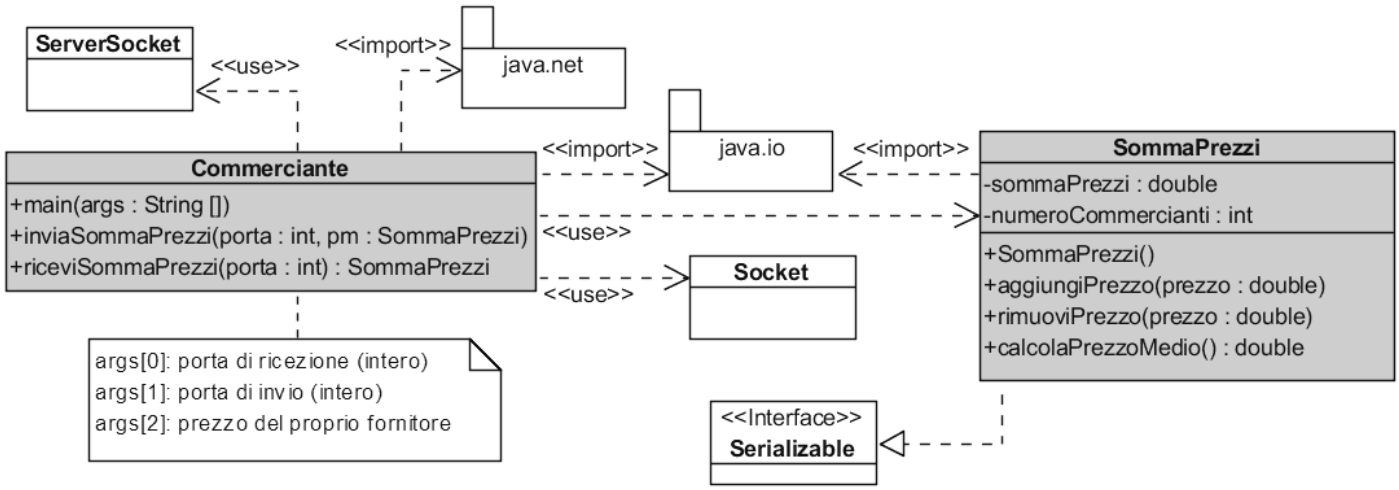


Fig.5 - Classi di cui deve essere composta l'applicazione (in grigio) e classi/package da usare (in bianco)