

# Ingegneria del software - Esame del 10 febbraio 2004

Tempo a disposizione: 2 ore

Regolamento: 1) libri e quaderni chiusi; 2) vietato scambiare informazioni con altri; 3) indicare su tutti i fogli, con chiarezza, nome e numero di matricola; 4) è ammesso l'uso della matita.

## Esercizio 1

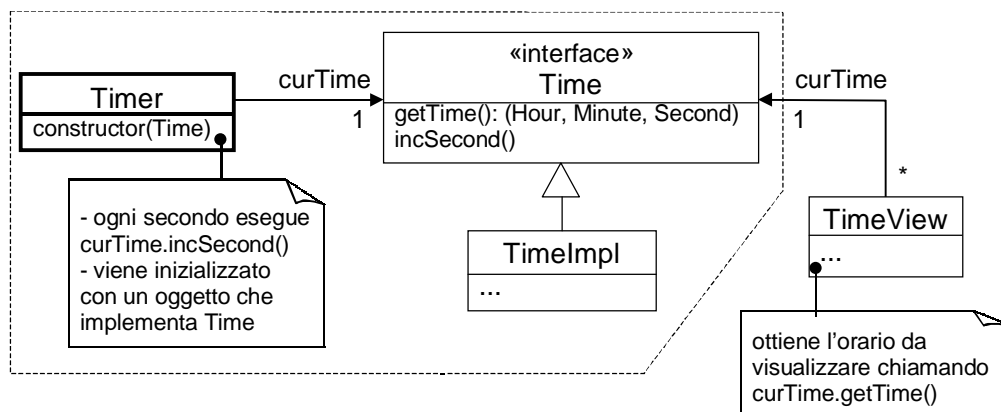
Considera la seguente interfaccia (una versione semplificata di `java.util.List`):

```
public interface List {  
  
    /** appende l'elemento specificato alla fine della lista */  
    boolean add(Object o);  
  
    /** rimuove tutti gli elementi della lista */  
    void clear();  
  
    /** restituisce l'elemento nella posizione specificata */  
    Object get(int index);  
  
    /** restituisce vero se la lista non contiene alcun elemento */  
    boolean isEmpty();  
  
    /** rimuove l'elemento nella posizione specificata */  
    Object remove(int index);  
  
    /** rimpiazza l'elemento nella posizione specificata con l'elemento specificato */  
    Object set(int index, Object element);  
  
    /** restituisce il numero di elementi nella lista */  
    int size();  
}
```

Si è realizzata una classe `LinkedList` che implementa tale interfaccia utilizzando una rappresentazione a lista doppiamente collegata. Si decide di scrivere i test di unità della nuova classe. Si realizzi una classe `LinkedListTest` che realizzi i test di unità per almeno quattro dei metodi della classe `LinkedList`.

## Esercizio 2

Considera il seguente design class diagram.



*Time* è una interfaccia per oggetti che rappresentano un orario. Lo stato di un *Time* è accessibile mediante il metodo *getTime()* e modificabile solo per incrementi di un secondo mediante il metodo *incSecond()*. La classe *Timer* ogni secondo invoca il metodo *incSecond()* su un oggetto di tipo *Time* che gli viene fornito in fase di inizializzazione. La classe *TimeView* permette di visualizzare oggetti di tipo *Time*. Secondo il progetto mostrato, un *TimeView* non può sapere quando il *Time* che deve visualizzare cambia stato.

Devi modificare il progetto in modo che oggetti *TimeView* possano accorgersi dei cambiamenti di stato del loro oggetto *Time*. Purtroppo la parte racchiusa dalla linea tratteggiata non è modificabile. Inoltre la soluzione che prevede che *TimeView* faccia polling su *Time* è considerata da scartare in quanto inefficiente. Sei invece libero di aggiungere classi che hanno associazioni e derivano da classi e interfacce nella parte tratteggiata e di modificare la classe *TimeView*.

Mostra la tua soluzione fornendo un design class diagram con sufficienti dettagli e annotazioni dal quale si evinca la realizzabilità della soluzione proposta. Se applichi design pattern noti specifica il ruolo delle varie classi nell'ambito del pattern con dei commenti.

### **Esercizio 3**

Si deve realizzare un sistema informatico per automatizzare alcuni processi nel contesto produttivo descritto di seguito. Dovrai dimostrare, con un elaborato scritto, di essere in grado di lavorare sul progetto secondo la metodologia unified process. Puoi sviluppare l'elaborato come credi ma una possibile traccia può essere la seguente

- un breve sviluppo del caso d'uso che ritieni più importante indicando attori, trigger, stakeholders, precondizioni, postcondizioni, lo scenario principale e qualche accenno a eventuali scenari secondari
- la descrizione di una possibile architettura software per risolvere il problema proposto al livello di dettaglio che ritieni opportuno
- alcuni dei principali rischi che possono compromettere il successo del progetto ed alcune idee per mitigarli
- se hai tempo puoi menzionare ulteriori casi d'uso, iniziare a fare una programmazione del tempo (lunghezza della iterazione, fasi, milestones), una stima delle risorse (in termini di tempo/persona) che ritieni necessarie o entrare in qualche dettaglio dell'architettura.

Qui di seguito troverai i dettagli per svolgere l'esercizio. Per tutto ciò che non è esplicitamente detto puoi fare tu delle ipotesi.

#### ***Contesto produttivo***

La ditta DistribuLibri acquista libri dagli editori e li promuove nelle *librerie* mediante i suoi *agenti*. Durante la settimana gli agenti visitano le librerie e prendono le *ordinazioni*, cioè i quantitativi che ciascuna libreria intende acquistare di ciascun libro. Poiché le ordinazioni devono essere tempestivamente evase dal magazzino di DistribuLibri gli agenti si devono affrettare a comunicare le ordinazioni raccolte in sede.

#### ***L'idea per l'informatizzazione***

Gli agenti verranno dotati di calcolatori portatili. Si deve realizzare un sistema software, distribuito sui portatili degli agenti e su uno o più calcolatori in sede, che supporti DistribuLibri nei processi di business sopra descritti. Gli agenti tutte le sere comunicheranno gli ordini effettuati per mezzo di rete e/o collegamento telefonico e allo stesso modo riceveranno le informazioni sulle novità editoriali da promuovere che verranno memorizzate sul portatile.

### ***Aspetti critici***

- E' fondamentale che le informazioni sugli ordini non vadano perse e/o duplicate a causa di errori e/o ritrasmissioni.
- La ditta mette a disposizione del team di sviluppo il personale del magazzino e della direzione ma **nessuno degli agenti**
- A fine progetto è previsto un collaudo di una settimana alla fine del quale verrà espresso un parere della direzione, del magazzino e degli agenti.
- Il progetto si intende completato con successo se dopo il collaudo avrà ottenuto parere positivo di tutte le figure professionali coinvolte.