Non è consentito usare libri o appunti.

Esercizio 1 (15 punti) – *Libro* è una classe che descrive un oggetto libro. Esso è definito dal *titolo*, dall'*autore*, dal *numero* di pagine, dalla *lista* dei capitoli e da un identificatore unico *id* che inizia da 1 e viene incrementato di uno ogni volta che viene creato un libro che abbia almeno il titolo specificato.

Ogni libro è caratterizzato da

- 3 costruttori: uno senza parametri che imposta il titolo e l'autore alla stringa "Sconosciuto" e il numero di pagine e l'id a 0 e a *null* la lista dei capitoli, uno che specifica il titolo, il numero di pagine e la lista dei capitoli, imposta a "Sconosciuto" l'autore e imposta l'id al prossimo numero in sequenza, e uno che specifica titolo, autore, numero di pagine e lista dei capitoli e imposta l'id come nel secondo.
- Il metodo equals, che verifica se due libri hanno stesso autore, stesso titolo, stesso numero di pagine e stessa lista di capitoli
- Il metodo toString
- Il metodo getInitials che restituisce le iniziali del nome e cognome dell'autore, se questo è noto (cioè se non è uguale a "Sconosciuto" o a null). Si supponga che al più vi possono essere due nomi e un cognome.

Un libro può essere

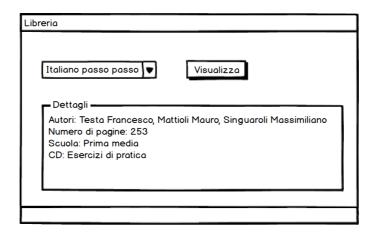
- un *Libro Scolastico*, caratterizzato dall'ordine e grado scolastico (es. terza media), dalla materia e dal *CD* (si assuma un campo di tipo String col titolo)
- un *Volume* di un'enciclopedia, caratterizzato dal *titolo* dell'enciclopedia e dal nome del *curatore*;

Definire un'opportuna gerarchia di classi e ridefinire per ciascuna tipologia di libro i metodi equals e toString

Esercizio 2 (7 punti) Scrivere una classe *Libreria* che consente di

- aggiungere nuovi libri alla libreria;
- ricercare un libro in base al titolo. Tale metodo lancia una eccezione controllata se il titolo non è presente nella libreria;
- ottenere l'elenco dei libri come oggetto stringa.

Esercizio 3 (8 punti) - Implementare un'interfaccia grafica che, data una lista di libri memorizzati in un file testuale, visualizzi una combo-box con i titoli dei libri e un pulsante *visualizza* che visualizza in un'area di testo i dettagli del libro selezionato.



Ogni violazione delle regole enunciate ai punti sotto elencati comporta l'annullamento della prova (l'elaborato viene valutato 0).

- 1. Prima di eseguire eclipse assicurarsi che non ci siano file Java (sorgenti, bytecode, workspace,progetti, pacchetti) sul desktop.
- 2. Eseguire eclipse specificando un workspace sul desktop.
- 3. Durante la prova d'esame è vietato usare:
 - a. libri e appunti sia in forma cartacea che in forma digitale
 - b. supporti di memoria esterni
 - c. un font di dimensione maggiore di 10 punti.
- 4. Non è consentito modificare i file allegati alla traccia.
- 5. Il nome del progetto consegnato deve cominciare con COGNOME seguito dal carattere underscore e quindi dal NOME (tutto in maiuscole). Ad esempio, il nome del progetto di Marco Rossi può essere ROSSI_MARCO, ROSSI_MARCO_P2, ROSSI_MARCO_ESERCIZIO, ROSSI_MARCO_549449384, etc.
- 6. Il file da consegnare deve essere creato da eclipse seguendo i passi:
 - a. Seleziona "export..." nel menu file
 - b. Seleziona "Archive File" in "General"
 - c. Pressa "Next"
 - d. Seleziona progetto da esportare
 - e. Controllare il percorso del file (nell'area di testo con etichetta "To archive file:")
 - f. Assicurarsi che i pulsanti radio nel pannello Options siano selezionati su "Save in zip format" e "Create directory structure for files"
 - g. Pressa "Finish"

Assicurarsi che i progetti consegnati possono essere importati in eclipse come: General → Existing Projects into Workspace

- 7. Dopo aver effettuato la consegna, assicurarsi che il file sia stato ricevuto dal server docente (chiedere al docente) e quindi procedere alla cancellazione dei file prima di spegnere il PC
- 8. Per avere una copia del progetto consegnato rivolgersi al docente prima di lasciare l'aula (munirsi di supporto di memoria USB)