

Raccomandazioni e Chiarimenti sul progetto di Laboratorio di Programmazione 2004/2005

Luca Tesei

15 Feb 2005

1 Raccomandazioni e Chiarimenti

- La suddivisione in classi suggerita nel testo è largamente consigliata, almeno come base da estendere.
- Prestate attenzione agli specificatori di accesso. Ricordo che tutte le variabili istanza dovrebbero essere dichiarate private, a meno di giustificati motivi.
- Evitate assolutamente variabili o metodi statici.
- Evitate i metodi con effetti collaterali. Non mettete **mai** una stampa System.out che contiene un messaggio o un risultato all'interno di un metodo. I risultati vanno restituiti come tipo di ritorno (potete creare classi interne apposite i cui oggetti servono esclusivamente a contenere i risultati di un certo metodo). Mentre i messaggi di errore vanno gestiti con le eccezioni.
- Fate due soli metodi principali di GestoreRicerche (uno per la prima ricerca richiesta e l'altro per quella della settimana). Utilizzate i parametri di questi metodi per specificare i vari criteri di ricerca. In particolare prestate attenzione al suggerimento che do nel testo riguardo la classe CaratteristicheAula. Il risultato di questi metodi dovrebbe essere un oggetto di tipo Aula. Questo riferimento e altri parametri andranno poi passati ad un opportuno metodo di DatabasePrenotazioni per visualizzare il risultato della ricerca. Questo opportuno metodo potrà ritornare una stringa con i risultati o un ArrayList di oggetti opportuni...
- Pensate a metodi "furbi" di organizzare gli oggetti della classe Prenotazione nel DatabasePrenotazioni. Per esempio potreste pensare di in-

serire in maniera ordinata le prenotazioni, per aule e/o per data, in modo da ottimizzare poi le ricerche.

- Non “dimenticatevi” di gestire la scelta di ore consecutive e/o non consecutive.
- Per la ricerca della settimana potete individuare la settimana tramite un giorno della stessa, o il lunedì della stessa.
- La parte più difficile del progetto è fornire la possibilità di prenotare una stessa aula nello stesso giorno, nelle stesse ore, con descrizione diversa, da parte di uno stesso utente **dopo che questi ha dato conferma**. Ci sono due problemi da gestire:
 1. Oggetti Prenotazione multipli per la stessa ora (i metodi per capire se ci sono le ore libere richieste che si basano sul conteggio delle ore prenotate devono tenere conto dei “doppioni” dovuti a prenotazioni multiple sulla stessa ora)
 2. La conferma non andrebbe richiesta nel metodo come effetto collaterale. Il mio consiglio è di creare una eccezione apposita che viene sollevata in caso di tentativo di prenotazione su un’aula e un’ora già impegnate. All’esterno del metodo si dovrà gestire l’eccezione verificando se ci sono gli estremi per richiedere la conferma. Nel caso poi in cui la conferma sia data si potrebbe richiamare di nuovo il metodo settando però un parametro apposito che dice di effettuare la prenotazione senza sollevare l’eccezione (oppure, ancora meglio, si potrebbe creare un metodo apposito per questo caso specifico, tipo `prenotaPrenotazioneMultipla(.....)`).
- L’applicazione richiesta **non** richiede che le prenotazioni, aule, utenti inseriti vengano in qualsiasi modo memorizzati su file. Il progetto ha come scopo quello di evidenziare la vostra capacità di progettazione orientata agli oggetti e il vostro apprendimento dei concetti di classe, oggetto, metodo (oltre che le varie caratteristiche di Java) e quindi una sua effettiva operatività non è importante. Semplicemente alla chiusura dell’applicazione verranno perse tutte le informazioni memorizzate negli oggetti in memoria.
- **NON** perdetevi tempo ed energie a programmare orribili interfacce testuali a menu (premere 1 per inserire aula, premere 2 per... cose del genere). Oltre che orribili risultano del tutto superate. Se proprio volete dotare di una interfaccia la vostra applicazione provate a fare un’interfaccia grafica (è relativamente facile). Ribadisco che l’interfaccia **NON** è richiesta dal

progetto. La cosa fondamentale, ripeto, è la struttura e l'organizzazione delle classi e dei metodi del **nucleo** dell'applicazione.

- Per la presentazione all'orale semplicemente scrivete una classe di test che contiene un main con, in sequenza, una serie di creazioni di oggetti Aula, Prenotazione etc. seguite da messaggi su System.out di quello che state facendo. Tipo "Adesso ho creato le aule e faccio un inserimento" "Inserimento di ... eseguito". "Adesso faccio una ricerca" Chiamata del metodo con gli opportuni parametri. "Visualizzo il risultato della ricerca: ...". Ciò è perfettamente sufficiente per far vedere che quello che avete scritto funziona. Con questo approccio, inoltre, non bisogna nemmeno caricare dati da file di testo come suggerito, e male interpretato, nel testo del progetto. Lasciate perdere i file di testo.
- Non cercate di svicolare. Vi vengono richieste poche cose e precise: l'implementazione delle varie classi e l'implementazione di certe ricerche. Non fate altre cose se quelle richieste sono incomplete.