

Il progetto consiste nella realizzazione di una piccola applicazione per gestire un Sistema di Prenotazione degli esami di un corso di laurea.

Appello è usato come sinonimo di esame, ma NON di insegnamento/corso

Si devono costruire e gestire (per uso interno) due archivi, uno contenente tutti i dati necessari degli student(no. Matricola, cognome, nome, anno di corso, pagamento tasse SI/NO), ed uno con i dati (nome corso, A.A., anno di corso, docente) degli insegnamenti offerti. Inizialmente, gli archivi sono costruiti utilizzando dati che vengono letti da file esterni. Questi dati non cambiano.

Per ogni esame, sono disponibili 4 informazioni: il nome dell'insegnamento corrispondente, la data dell'esame, il docente titolare del corso associato all'esame, ed un valore booleano per indicare prenotazione ancora aperta o no (una quinta, no. massimo di prenotazioni, è nascosta).

Per semplicità, si presume che ciascun corso sia identificato univocamente dal suo nome (stringa, eventualmente con numerazione, come BasiDiDati2) e quindi non ci sono in questo archivio due esami con lo stesso nome.

Deve essere possibile aggiornare l'elenco degli esami (NON quello dei corsi/insegnamenti), con aggiunte di un nuovo appello, con data, docente e limite, e 'chiudere' le prenotazioni di un particolare esame. Ogni esame è rappresentato da un oggetto della classe Esame. Ad ogni esame è associato un insieme di studenti, quelli già prenotati per l'esame, che deve essere recuperabile dal docente.

Ogni studente è rappresentato da un oggetto della classe Studente ed è caratterizzato da un nome, un cognome, un numero di matricola (che identifica univocamente ogni studente), l'anno in corso ed un valore booleano che riflette la posizione pagamento tasse universitarie.

Non è possibile registrare per un esame due studenti con lo stesso numero di matricola. ~~Poichè il numero di prenotazioni per ogni studente è limitato (arbitrario, ma fissato per tutti gli studenti), deve~~ Deve essere possibile cancellare una prenotazione (se ancora aperta). Prima di accettare una prenotazione, il sistema deve verificare (tra le altre cose) che lo studente è in regola con il pagamento delle tasse universitarie per l'A.A. in corso.

All'inizio dell'accesso al sistema, la scelta è tra l'accesso alle "Prenotazioni" da parte dello studente e l'accesso alla "Gestione" degli esami in una zona riservata ai docenti

Con la scelta "Gestione", si deve optare per l'inserimento in archivio di un nuovo esame (con i 4 campi per i dati corrispondenti) o per esportare/salvare su file testo tutte le prenotazioni di un esame.

Se l'aggiunta è possibile (il docente risulta il titolare del corso per il corrente A.A.) viene mostrato un messaggio di conferma; altrimenti si presenta il messaggio d'errore appropriato. La creazione di un appello automaticamente 'apre' le prenotazioni. Deve essere possibile chiudere esplicitamente le prenotazioni (da parte del docente che ha creato l'appello)

L'Esportazione crea ~~un file testo cui possiamo assegnare un nome tramite una finestra di dialogo, con i due bottoni OK e Annulla. Il risultato ottenuto premendo il bottone OK è~~ un file di testo con le informazioni (in ordine alfabetico) nel seguente formato

Esame di

Docente:

Per l' Appello di 'data' si sono prenotati i seguenti studenti

1246835	Caio	Paolo
1357924	Sempronio	Elena
1234567	Tizio	Mario

Con la scelta "Prenotazioni", è possibile inserire Nome, Cognome e numero Matricola dello studente, e selezionare l'esame; a questo punto viene data l'opzione di mostrare i "Dettagli" dell'esame, quella per

“Prenotare” l’esame, e quella per rimuovere una prenotazione [dello studente stesso](#). Con “Dettagli”, vengono evidenziate le informazioni dell’esame (corso, docente e data) in formato testo. Con “Prenota”, si aggiungono i dati all’elenco prenotati. Se l’aggiunta è possibile (pagamento tasse in regola, no. Matricola non già presente, anno di studi appropriato, prenotazione non chiusa), viene mostrato un messaggio di conferma della registrazione; altrimenti si presenta il messaggio d’errore appropriato. Se la rimozione è di una prenotazione esistente, viene mostrato un messaggio di conferma, altrimenti si presenta il messaggio d’errore appropriato. Per questa applicazione, si ignorano aspetti di autenticazione dell’utente.

La versione “minima” di questa applicazione va sviluppata ipotizzando che tutti i dati/comandi dall’esterno siano ricevuti da tastiera o file.

[Per una votazione più alta, è possibile sviluppare una versione interattiva con finestre.](#)

La versione “deluxe” deve contenere, in aggiunta, un sistema di finestre (come oggetti di classi DocenteFrame, StudenteFrame, ecc.) in cui compaiono alternative visualizzate con pulsanti da “premere” (Aggiungi, Esporta, Dettagli, Prenota, Cancella) e campi dove “selezionare” (il nome del corso) o “inserire” (data, cognome, no. matricola, ecc.) le informazioni ricevute dall’esterno.

VANNO PRESENTATI

1. Diagramma delle classi in UML
2. Documentazione utilizzando Javadoc
3. Codice completo