

Laboratorio di Introduzione alla Programmazione

Progetto Finale - “Registro Frequenze”

di Filippo Fontanelli e Francesca Brogi

Breve spiegazione della struttura del progetto

Il programma è basato su una serie di menù e sottomenù, si parte dal menù principale con due principali scelte e la terza per l'uscita, ad ognuna delle due scelte corrisponde un sottomenù e così via.

In particolare ci sono menù per la creazione di un nuovo registro, per l'accesso a uno già esistente e quindi altri menù per la modifica, aggiunta o eliminazione di studenti, lezioni e presenze.

Inoltre si può effettuare l'accesso al file di ogni singolo studente dove sono presenti i suoi dati (Nome, Cognome, Matricola) e l'elenco delle lezioni a cui era presente; oppure si può accedere al registro del corso dove troviamo l'elenco delle lezioni con una breve descrizione del loro contenuto, la data e per ogni lezione i nomi degli studenti presenti. Ovviamente pur essendoci questa varietà di classi, quando l'utente aggiorna il registro delle presenze sia che lo faccia a partire dallo studente o che lo faccia a partire dalla lezione, il programma aggiorna automaticamente tutte le varie classi.

Scelte progettuali e motivazioni

Inizialmente abbiamo cercato di analizzare le varie opzioni per lo sviluppo del progetto, quindi prima di lavorare e di compilare il programma, abbiamo cercato di capire come doveva essere organizzato.

La soluzione più ottimale ci è sembrata quella di organizzare il Progetto con una struttura ad albero. Le classi che abbiamo implementato per prime, considerandole le basi da cui partire per procedere nella realizzazione del Progetto, sono state quella relativa al singolo Studente, che con molta fantasia abbiamo chiamato “Studente”, e quella relativa alla singola Lezione, chiamata “Lezione”.

In contemporanea abbiamo provveduto a creare due Classi che fungessero da contenitori di Studenti e di Lezioni, che abbiamo chiamato “RegStudenti” e “RegLezioni”; come si può vedere dalla documentazione “javadoc” le suddette classi contengono metodi essenziali per l'inserimento, eliminazione e modifica di Studenti e Lezioni.

A questo punto dopo aver definito le strutture basilari abbiamo incominciato a creare una serie di classi che fungessero da nucleo del Progetto.

Per far sì che ci fosse una chiara separazione tra la "logica del progetto" e i moduli che permettono di accedervi, abbiamo deciso di strutturare il nucleo su tre livelli e quindi tre relative classi:

- 1- RegistroFrequenzeInterfaccia
- 2- RegistroFrequenzeImp
- 3 -RegistroFrequenze.

Al termine della compilazione delle varie classi del nucleo, ci siamo concentrati sulla creazione del menù; tale menù è stato da noi strutturato su vari livelli:

menù organizzati ad “albero”, cioè un menù principale dalle cui opzioni si passa ad altri menù e così via fino ad arrivare all’operazione desiderata.

Vedere file DiagrammaMenu.pdf

Al fine di far sì che tutto il Progetto funzionasse regolarmente e senza errori è stato necessario testare le varie classi, ci siamo serviti principalmente della funzione di Debug e di alcune classi di test da noi create.

Infine abbiamo creato una piccola batteria di file di test da utilizzare tramite shell.

E come richiesto ci siamo occupati di definire le specifiche per le varie Classi e i relativi Metodi, creando la documentazione JavaDoc. All'interno del codice vi sono numerosi commenti che abbiamo usato durante l'implementazione, ho preferito non toglierli tutti, in quanto potrebbero tornarci utili qualora si presentasse la necessità di rileggere il codice a distanza di tempo e potrebbero tornare utili pure a lei nella comprensione delle logiche con qui abbiamo implementato certi metodi.

Spiegazione delle scelte fatte in corrispondenza di eventuali ambiguità nella specifica del progetto e di alcuni aspetti solo parzialmente specificati.

L'applicazione oltre a permettere la gestione di una collezione studenti con inserimento e rimozione, permette anche la modifica; inoltre oltre a permettere la gestione della collezione studenti, l'applicazione permette la gestione delle collezioni lezioni e presenze tramite vari menu'.

E' stata introdotta una piccola varianza riguardante l'inserimento sia di studenti che di lezioni, l'applicazione infatti permette l'inserimento sia manuale (da tastiera) sia da file testuale.

Tale file deve avere una forma ben definita:

Esempio: file Studenti.txt

```
Fontanelli - Filippo - 422385
Gennai - Sara - 544455
Palmiro - Cangini - 555556
Rossi - Nicole - 333322
Baresi - Franco - 121212
Fontanelli - Alessandro - 331222
Brogi - Francesca - 422555
De Luca - Lucia - 334455
**Fine**
```

Tradotto : il file deve contenere uno studente per riga, nella quale l'ordine delle informazioni deve essere il seguente :

Cognome – Nome – Matricola

e ogni singola informazione deve essere separata dalle successive tramite il seguente insieme di caratteri e spazi : “ - ” e come formula di chiusura del file nella riga successiva all'elenco degli studenti deve essere presente il metacarattere : “**Fine**”.

La stessa forma e le stesse regole sono utilizzate anche per il file relativo alle lezioni.

Esempio :

12/10/2008 - Introduzione - Breve presentazione del corso
13/10/2008 - Vettori - Introduzione dell'argomento e discussione con gli alunni
14/10/2008 - Eccezioni - Introduzione dell'argomento
15/10/2008 - Debug - Introduzione dell'argomento e discussione con gli alunni
Fine

Tradotto : il file deve contenere una lezione per riga, nella quale l'ordine delle informazioni deve essere il seguente :

Data – Nome – Argomento

e ogni singola informazione deve essere separata dalle successive tramite il seguente insieme di caratteri e spazi : “ - ” e come formula di chiusura del file nella riga successiva all'elenco degli studenti deve essere presente il metacarattere : “**Fine**”.

La visualizzazione delle statistiche consuntive avviene anch'essa tramite menu', inoltre è possibile visualizzare su video il registro e l'elenco degli studenti e delle lezioni.

Come richiesto dalla specifica del progetto, i vari registri di frequenza sono salvati sia in formato testo che binario, per quanto riguarda il formato testuale abbiamo previsto una forma specifica, se tale forma non viene rispettata il caricamento fallisce.

Il caricamento del file testuale puo' fallire solamente se le modifiche sono state apportate in modo inappropriato tramite un normale editor di testo, in quanto l'applicazione memorizza le informazioni all'interno del file in modo appropriato e quindi abbassa di molto o annulla l'eventualita' di un'errore. Tale forma prevede :

Esempio: file FormaSpecifica.txt

Nome del Corso con frequenza obbligatoria :FormaSpecifica

Numero minimo di presenze :10

Elenco delle lezioni e degli studenti :

****Start Lezioni****

12/10/2008 - Introduzione - Breve presentazione del corso
13/10/2008 - Vettori - Introduzione dell'argomento e discussione con gli alunni
14/10/2008 - Eccezioni - Introduzione dell'argomento e discussione con gli alunni
15/10/2008 - Debug - Introduzione dell'argomento e discussione con gli alunni

****End Lezioni****

****Start Studenti****

Baresi - Franco - 121212
Brogi - Francesca - 422555
De Luca - Lucia - 334455
De Martino - Francesco - 109088
Fontanelli - Alessandro - 331222

Fontanelli - Filippo - 422385
Gennai - Sara - 544455
Palmiro - Cangini - 555556
Rossi - Nicole - 333322

****End Studenti****

****Start Registro****

Registro delle Presenze :

---Lezione:---

12/10/2008 - Introduzione - Breve presentazione del corso

---Studenti:---

Baresi - Franco - 121212
Brogi - Francesca - 422555
De Martino - Francesco - 109088
Fontanelli - Filippo - 422385
Rossi - Nicole - 333322

---Lezione:---

13/10/2008 - Vettori - Introduzione dell'argomento e discussione con gli alunni

---Studenti:---

Brogi - Francesca - 422555
De Martino - Francesco - 109088
Fontanelli - Filippo - 422385

---Lezione:---

14/10/2008 - Eccezioni - Introduzione dell'argomento e discussione con gli alunni

---Studenti:---

---Lezione:---

15/10/2008 - Debug - Introduzione dell'argomento e discussione con gli alunni

---Studenti:---

Baresi - Franco - 121212
Brogi - Francesca - 422555
De Luca - Lucia - 334455
De Martino - Francesco - 109088
Fontanelli - Alessandro - 331222
Fontanelli - Filippo - 422385
Gennai - Sara - 544455
Palmiro - Cangini - 555556
Rossi - Nicole - 333322

****End Registro****

****Fine****

Tradotto :

Nome del Corso con frequenza obbligatoria : **"Nome del corso"**

Numero minimo di presenze :**10**

Elenco delle lezioni e degli studenti :

****Start Lezioni****

Elenco delle lezioni, le quali devono rispettare le medesime regole previste per il file Lezioni.txt, tranne la clausola “**Fine**”

****End Lezioni****

****Start Studenti****

Elenco degli studenti, i quali devono rispettare le medesime regole previste per il file Studenti.txt, tranne la clausola “**Fine**”

****End Studenti****

****Start Registro****

Registro delle Presenze :

---Lezione:---

Deve contenere le informazioni di una sola Lezione (Data – Nome – Argomento) .

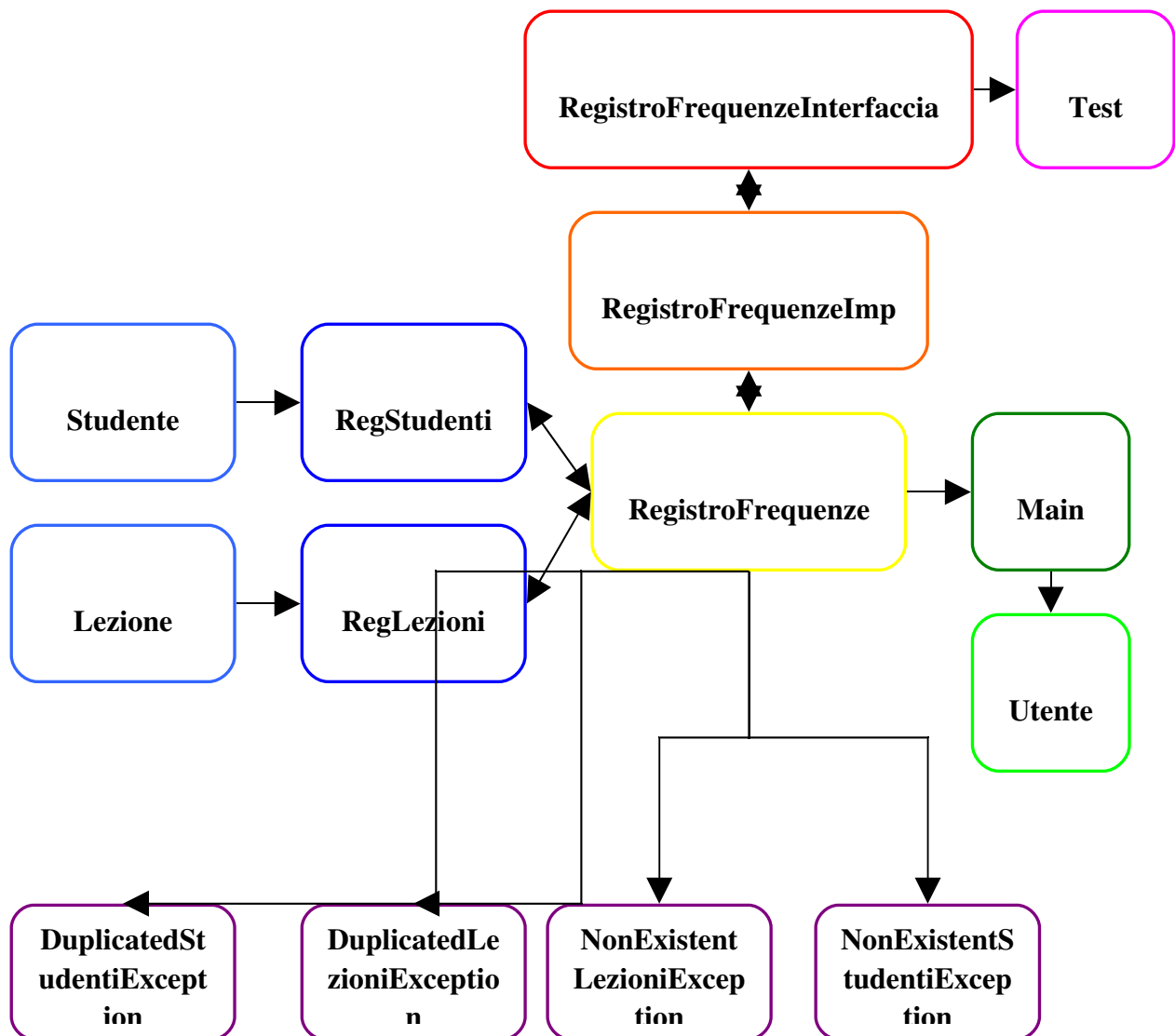
---Studenti:---

Deve contenere l'elenco degli Studenti (Cognome – Nome - Matricola) presenti alla lezione immessa precedentemente, sempre rispettando le regole previste Studenti.txt tranne la clausola *Fine**** .***

****End Registro****

****Fine****

Breve descrizione delle funzionalità e della struttura delle classi che compongono il progetto



Questo diagramma descrive in maniera schematica la struttura del progetto e il modo in cui sono organizzate le classi:

- **RegistroFrequenzeInterfaccia** e **RegistroFrequenzeImp**: sono rispettivamente l'interfaccia e la sua implementazione, contenenti tutti i metodi che realizzano le funzionalità per il registro delle frequenze.
- **RegistroFrequenze**: Classe principale del Progetto, la quale contiene l'attuale Registro. Questa classe serve per separare la parte centrale e fondamentale del progetto, cioè l'interfaccia e la sua implementazione, dalla parte di interazione con l'utente.
- **Main**: è la classe che contiene i metodi che gestiscono tutti i menù per l'interazione con l'utente ed è costituita soprattutto da cicli e *switch*.
- **RegLezioni**: è una lista di elementi omogenei di tipo *Lezione*. E' modificabile e ordinata secondo i criteri di ordinamento stabiliti per gli oggetti di tipo *Lezione*. Gli elementi della lista differiscono fra loro per data, nome e argomento. L'oggetto è serializzabile.

- **RegStudenti:** è una lista di elementi omogenei di tipo Studenti. E' modificabile e ordinata secondo i criteri di ordinamento stabiliti per gli oggetti di tipo Studenti. Gli elementi della lista differiscono fra loro per nome, cognome e matricola. L'oggetto e' serializzabile.
- **Studente:** contiene e gestisce tutti i dati di un singolo studente.
- **Lezione:** contiene e gestisce i dati di una singola lezione.
- **Test:** sono una serie classi che contengono i test per le classi: RegistroFrequenze, Lezione, RegLezioni, Studente e RegStudenti.
- **DuplicatedLezioniException:** classe di eccezione per indicare che una lezione che si vuole aggiungere è già presente.
- **DuplicatedStudentiException:** classe di eccezione per indicare che uno studente che si vuole aggiungere è già presente.
- **NotExistentLezioniException:** classe di eccezione per indicare che una lezione che tentiamo di rimuovere non e' presente nel registro.
- **NotExistentStudentiException:** classe di eccezione per indicare che uno studente che tentiamo di rimuovere non e' presente nel registro.