Gestore Lezioni

AliceVittoria Sasso
Università degli studi di Padova
Matricola 1074588

1 Sviluppo

1.1 Sistema

Sistema operativo di sviluppo: Ubuntu 16.04 LTS
Compilatore: gcc version 5.4.0
Qt: version 5.6.0
QMake: version 3.0
QtCreator: version 3.6.1

1.2 Scopo

Per la realizzazione del progetto “Freedom” di Programmazione ad Oggetti a.a. 2015/2016 è stato scelto di programmare un Gestore di Lezioni per il conseguimento di Licenze di Volo.
Il progetto è stato nominato Gestore Lezioni e potrebbe essere utilizzato in una generica Avio Scuola per gestire le lezioni pratica impartite agli studenti e tenere traccia delle ore di volo maturate dagli stessi per il conseguimento delle ore minime curriculari per licenze offerte dall’Avio Scuola.

Gestore Lezioni contempla i seguenti tipi di lezione che riguardano i tipi diversi di licenze offerte dall’AvioScuola:

- Pilota Privato, nelle varianti Normale e Acrobatico
- Pilota di Aliante
- Pilota di Ultraleggeri

Gestore Lezioni permette di consultare tutte le lezione effettuate(in tutte le loro varianti) , visualizzarne tutti i dati ed il costo.
E’ possibile anche aggiungere nuove lezioni, eliminare lezioni, effettuare il pagamento delle singole lezioni, il tutto mantenedo la consistenza del DataBase.

E’ anche possibile consultare l’elenco studenti per i quali Gestore Lezioni specifica per ogni cliente le ore di lezione pratica eseguite per ogni tipologia e segnala il raggiungimento dei minimi previsti per ogni licenza, utile per individuare gli studenti preparati.
E’ inoltre visualizzato il debito corrente con l’AvioScuola e l’ammontare pagato per ogni studente, in base allo stato corrente *pagata* oppure *non pagata* delle lezioni per tale studente.

Gestore Lezioni utilizza un database in formato XML nel quale vengono memorizzate le lezioni con i loro dati. Vengono salvate le modifiche su tali lezioni, le lezioni aggiunte e quelle eliminate.

1.3 Note sulle avio-licenze

Le licenze prese in considerazione sono le seguenti:

- PPL-Private Pilot License, licenza di pilota privato (di aereo) a scopo privato, i requisiti minimi sono il raggiungimento di 12 ore di volo in doppio comando con istruttore, 45 ore di volo totale
- VDS-Volo Diporto o Sportivo, attestato per il volo con ultraleggeri, i requisiti minimi sono il raggiungimento di 16 ore di volo in doppio comando con istruttore
- GPL-Glider Pilot License, licenza di pilota di alianti, i requisiti mnimi sono il raggiungimento di 7 ore di volo in doppio comando con istruttore, 4 ore di volo in solitaria, 13 ore di volo totali
- Acrobatic PPL License, licenza di pilota privato acrobatico(di aereo) , i requisiti minimi sono il raggiungimento di 10 ore di volo, rientra nel tipo PPL è un abilitazione riguardante tale licenza

Per i costi e la poca diffusione sul territorio delle licenze acrobatiche, non è stata tenuta in considerazione tale abilitazione per GPL e VDS nelle tabelle o nella creazione di nuove lezioni di questi due tipi..

Gestore Lezioni specifica per ogni studente le ore di lezione pratica e segnala il raggiungimento dei minimi previsti per ogni licenza, utile per individuare gli studenti preparati.

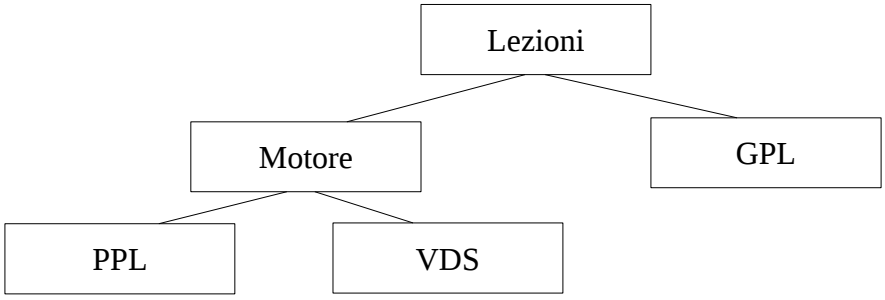
I minimi previsti coincidono con quelli di legge, è un aspetto importate del software in quanto per la legislazione europea ed italiana è possibile procedere con gli esami ed i corsi solo al superamento delle soglie minime.

Da notare che le ore maturate in altre Scuole possono essere accertate dagli organi competenti, non dalle Scuole, che quindi tenedono a tenere conto solo delle ore offerte nelle proprie strutture e offrire solo pacchetti di ore minime prestabilite in modo da accertarsi di mantenere i propri clienti per tutta la durata della preparazione.
Per questo il software segnala il raggiungimento delle soglie minime effettuando il conto solo in base al **database** fornito.

La vera preparazione è comunque accertata dagli istruttori, che spesso richiedono alla studente di effettuare alte lezioni. Per questo è possibile effetture più ore di quelle minime previste.

2 Classi

2.1 Gerarchia Lezioni



La gerarchia si divide in base alla tipologia di lezione effettuata.

La classe base astratta **Lezione** contiene i dati utili per tutte le lezioni della gerarchia: id della lezione, codice dello studente, codice immatricolazione veicolo, presenza dell'istruttore a bordo, la durata in minuti, lo stato del pagamento, il tipo della lezione, se si tratta della variante acrobatica per quel tipo, il numero dei traini utilizzati per arrivare in quota. Comprende il metodo virtuale puro per il calcolo del costo della lezione, e un metodo virtuale puro per la scrittura nel DataBase.

La classe **Motore** eredita da Lezione, è base per le classi che richiedono il calcolo del costo della benzina, ereditano da Motore le classi PPL e VDS che riguardano veicoli dotati di mototore e implementano quindi il metodo per il caolcolo della benzina che in Mototre è virtuale puro.

La classe **PPL** eredita da Motore e ne implementa il metodo di calcolo della benzina, utilizzato nel metodo di calcolo del costo della lezione, anch'esso implementato. Implementa saveLezione per la scrittura nel DataBase.

La classe **VDS** eredita da Motore e ne implementa il metodo di calcolo della benzina, utilizzato nel metodo di calcolo del costo della lezione, anch'esso implementato. Implementa saveLezione per lascrittura nel DataBase.

La classe **GPL** eredita dalla base Lezione in quanto riguarda veicoli sprovvisti di motore, e ne implementa il metodo di calcolo del costo della lezione. Implementa saveLezione per lascrittura nel DataBase.

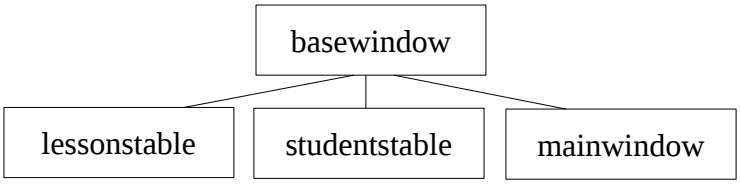
2.2 Contenitore "Complete"

Gestore Lezioni si serve della classe contenitore nominata Complete per gestire e memorizzare le Lezioni.
Tramite questa classe sarà possibile l'aggiunta, l'eliminazione e la modifica delle Lezioni.
Esegue anche i conti relativi alle ore.

2.3 Classe DataBase

Si occupa di interagire con il DataBase XML fornito al programma. Carica le lezioni valide nel contenitore e le modifiche avvenute tramite il contenitore saranno apportate dalla classe DataBase nel file xml.
Contiene il campo dati db, di tipo Complete, sulla quale lavora.
In assenza di databse viene costruito un file XML pronto ad accogliere nuovi dati(stato vuoto). Può trovarsi anche nello stato Corrotto in caso di lettura di Lezioni con tag scorretti. Entrambe le eventualità vengono segnalate all'utente all'avvio del programma.

2.4 View



Si è deciso di aderire al design pattern Model-View-Controller.
Le classi relative alla GUI si servono di classi controller in modo da separare la logica dalla presentazione.

La Classe **basewindow** è la classe base per tutte le finestre e tabelle della gui e deriva da QWidget. Contiene il DataBase e il metodo getDB() per essere utilizzato dalle classi derivate; contiene il metodo updateView() virtuale puro.

La Classe **lessonstable** deriva dalla basewindow, ne sfrutta il database ed implementa updateView() utilizzato per ricaricare la tabella ad ogni modifica, aggiunta ed eliminazione.
Crea una tabella QtableWidget che permette di visualizzare per ogni lezione il suo ID, il codice dello studente, il codice immatricolazione veicolo, presenza dell'istruttore, la durata, lo stato del pagamento, il tipo della lezione, il costo della lezione. Se si tratta della variante acrobatica per il tipo PPL, il numero dei traini utilizzati per arrivare in quota per il tipo GPL.

La Classe **studentstable** deriva dalla basewindow, ne sfrutta il DataBase ed implementa updateView() utilizzato per ricaricare la tabella ad ogni modifica del database. Crea una tabella QtableWidget che visualizza per ogni studente il suo codice, i minimi raggiunti, il debito totate e il pagamento totale effettuato (in Euro).

La Classe **mainwindow** deriva da basewindow e contiene tre differenti tab. La prima per visualizzare le lezioni, si serve della Classe lessonstable per visualizzare le lezioni, da qui si possono eliminare o pagare le lezioni. La seconda per l'elenco studenti con le ore maturate e l'importo dovuto, o pagato, per tali ore.
La terza permette di inserire le tre tipologie di lezione, ogni una con particolari campi richiesti, a sinistra PPL, al centro VDS, a destra GPL.

2.5 Controller

La Classe **c_mainwindow** serve a realizzare l'implementazione MVC, si occupa di apportare le modifiche al database quando avvengono aggiunte, eliminazioni o modifiche di Lezioni.

2.6 Polimorfismo

All'avvio Gestore Lezioni procede con la load() di DataBase che memorizza in db, che è un contenitore Complete, le Lezioni lette nel file DataBase.xml sfruttando il polimorfismo.
Durante tale load() vengono create le lezioni nelle loro varianti PPL VDS GPL, puntatori a tali lezioni vengono passati alla add() di Complete, che utilizza un vettore di puntatori a Lezioni, permette il polimorfismo.
L'aggiunta di lezioni passa per il controller c_mainwindow dove un puntatore a qualsiasi tipo di lezione creata nella mainwindow viene passato alla addDB() della classe DataBase che sfrutta la add() di Complete.
Nella close() grazie al polimorfismo verranno chiamati i metodi saveLezione disponibili in ogni tipo di lezione.