Programmazione ad Oggetti

Nome: Follador Saverio
Matricola: 1096984

Indice

1 Introduzione

1.1 Ambiente di sviluppo

- **Sistema Operativo:** Manjaro Linux 16.10
- **Compilatore:** GCC 6.2.1
- **Versione Qt Creator:** 4.2.0
- **Versione Qt:** 5.7.1

1.2 Compilazione

Per compilare il progetto posizionarsi tramite terminale all'interno della cartella dello stesso. A questo punto, eseguire il comando **qmake** per generare il makefile. Eseguire quindi il comando **make** per avviare la compilazione. A compilazione terminata all'interno della cartella si potrà trovare un file eseguibile **ProgettoP2**.

1.3 Descrizione

Come da consegna si è sviluppato un software che permettesse l'accesso da parte di alcune tipologie di utenti ad un database. Il database che si è scelto di modellare è un database di prodotti di una qualsiasi azienda. Al database si può accedere come utente o come amministratore. Esistono tre tipologie di utenti (casuale, utilizzatore e rivenditore), ciascuna delle quali può ricercare prodotti conoscendone il nome e visualizzarne alcune informazioni. Le informazioni visualizzate saranno più o meno complete a seconda della tipologia di utente. L'amministratore può inserire, eliminare o modificare prodotti e utenti.

2 Struttura

2.1 Informazioni generali

Per modellare i database di Utenti e Prodotti si è utilizzata la struttura dati **vector<T>** contenuta nella libreria STL. Entrambi i database sono formati da un vector di puntatori a oggetti del tipo rispettivo. Non sono stati utilizzati puntatori smart in quanto, data la tipologia di database modellati, non vengono mai effettuate copie di oggetti. La gestione del garbage è affidata alle singole funzioni di eliminazione.

2.2 Classi modellate

2.2.1 Utente

Gli utenti a cui viene garantito l'accesso al database sono gestiti da una gerarchia di classi. Dalla classe base astratta **Utente** derivano le classi **UtenteCasuale**, **UtenteUtilizzatore** e **UtenteRivenditore**. UtenteCasuale rappresenta un utente che si presuppone acceda saltuariamente al database dell'azienda. Pertanto ha accesso solo ad alcune informazioni dei prodotti con le sue ricerche (nome e uso del prodotto). UtenteUtilizzatore rappresenta un utente che utilizza i prodotti dell'azienda. In quanto utilizzatore ha accesso anche alla durata dei prodotti oltre che a nome e uso. UtenteRivenditore rappresenta un rivenditore dei prodotti dell'azienda. Ha accesso a nome, uso, durata e prezzo dei prodotti.
La classe base Utente contiene le informazioni base di ciascun utente; le immagazzina attraverso oggetti di due classi create ad hoc: **LoginPw** e **Info**.
I diversi privilegi di ricerca per i vari tipi di utenti sono implementati tramite funtori (oggetti della classe **Funtore**).

2.2.2 Database

I database sono modellati tramite due classi: **DatabaseProdotti** e **DatabaseUtenti**. Salvo alcune piccole variazioni sono due classi speculari, contententi nella parte privata il contenitore vector di puntatori e nella parte pubblica i metodi per la gestione del database. La classe DatabaseUtenti contiene inoltre un metodo per l'autenticazione dell'utente.

2.2.3 Salvataggio su file

2.2.4 Controller

Il controller si occupa di collegare la parte logica e la parte grafica. Sono presenti due classi a questo scopo: **ControllerAdmin** e **ControllerUtente**. Un oggetto controller viene creato ogni qualvolta si effettua l'accesso. Il ControllerAdmin possiede metodi per inserire, modificare ed eliminare oggetti di entrambi i database; il ControllerUtente possiede metodi solo per interrogare il database dei prodotti.

2.2.5 Divisione Grafica/Logica

Si è cercato di seguire l'architettura **MVC** (Model-View-Controller). La parte grafica è separata dalla parte logica; entrambe vengono messe in comunicazione fra loro tramite il controller.
Ciascuna schermata del progetto è identificata da una specifica classe