

PROGETTO PAO “FASTSALARIES”, AA 2019/2020

INTRODUZIONE

Lo scopo di questo progetto è fornire ad un concessionario un'applicazione, che fornisce la possibilità di calcolare gli stipendi dei dipendenti in base alle vendite da loro effettuate. Questo ipotetico concessionario ha in vendita delle normali automobili, delle moto, delle auto ibride e dei monopattini elettrici.

AMBIENTE DI SVILUPPO

La mia parte di progetto è stato sviluppata su Windows 10, con ide QT Creator 4.11.0, basato su Qt 5.14.0, compilatore Mingw64 build 4.3.5

SUDDIVISIONE DEL LAVORO PROGETTUALE

Il progetto è stato sviluppato in coppia, con Davide Pivato (mat: 1187595). Sono stati decisi assieme i componenti del modello, della GUI e tutte le funzionalità connesse, successivamente il progetto è stato sviluppato in contemporanea, con il supporto di Git, e brevi brief settimanali per il coordinamento del lavoro e del testing delle nuove funzionalità. Io mi sono occupato delle classi Azienda, Dipendente, Lista, e nella GUI di PageHandler, Home e InserisciVendita. Davide si è occupato di scrivere la gerarchia polimorfa e le due classi GUI CalcolaStipendi e MostraVendite.

Le ore impiegate per i miei compiti sono state:

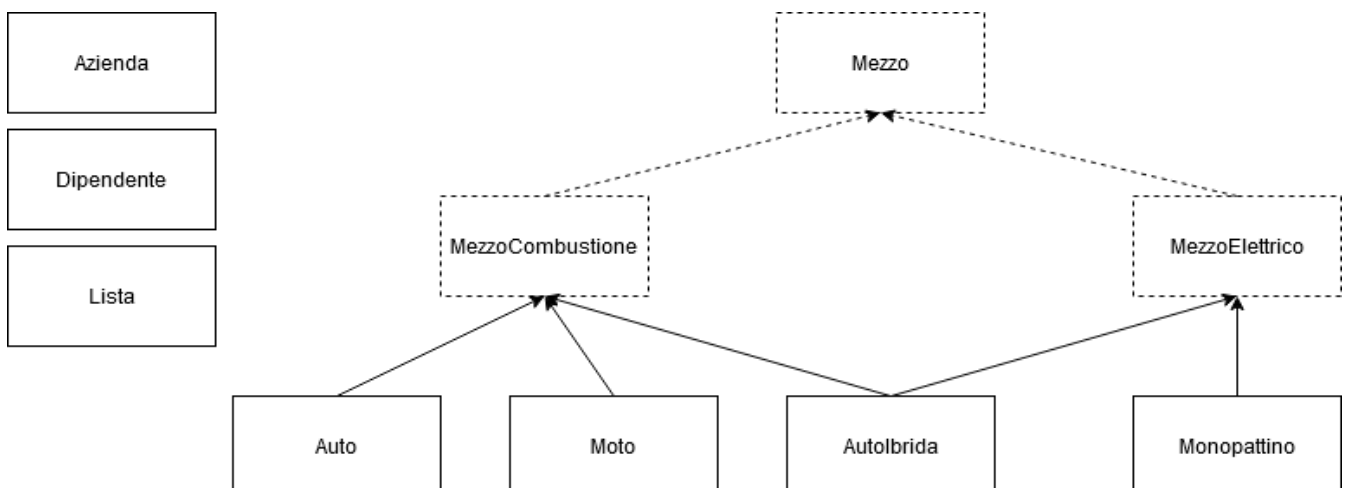
Concezione: 4, imparare QT: 9, programmazione: 35, debug: 9, testing: 3.

Il totale è 60 ore circa. Si è superata la soglia delle 50 ore probabilmente perché ho avuto problemi con molti errori nell'utilizzare correttamente i metodi e le proprietà della libreria Qt.

COMPILAZIONE

Viene fornito il file progetto.pro che deve essere usato per il comando qmake, seguito da make, per compilare con successo il progetto.

DESCRIZIONE DELLE CLASSI



La gerarchia rappresenta i mezzi venduti dal concessionario. Ogni mezzo istanziabile ha disponibile il metodo `getReward()`, dichiarato virtuale puro su `Mezzo`, che restituisce quanto (in €) aggiungere allo stipendio del dipendente che ha venduto quel mezzo, calcolato in base alle caratteristiche del mezzo stesso. Anche il metodo `getStar()` è definito in `Mezzo` come virtuale puro, e ritorna il valore (da 1 a 5) per definire la qualità del mezzo in oggetto. Le caratteristiche base che ogni mezzo ha (definite in `Mezzo`) sono un ID (che può non essere univoco nella concessionaria), la potenza e il prezzo; ogni derivazione aggiunge poi caratteristiche esclusive, e può modificare i parametri e il modo con cui vengono calcolate le stelle e il reward per quella specifica classe.

`AutoIbrida` chiude il diamante con `Mezzo-MezzoCombustione-MezzoElettrico`, è stato perciò necessario derivare da `Mezzo` `MezzoCombustione` e `MezzoElettrico` come virtual.

La gerarchia permette di istanziare solamente i mezzi veri e propri, ognuno con le sue caratteristiche. La si potrà estendere con nuovi mezzi derivando dalle classi virtuali `MezzoCombustione` e `MezzoElettrico`.

La classe `Lista` è il contenitore templatizzato, molto simile a quello visto a lezione, che viene usato per contenere i `Dipendenti*` e i `Mezzi*`.

La classe `Azienda` rappresenta il concessionario, essa contiene una lista di `Dipendenti*` e metodi utili al salvataggio/aggiornamento delle vendite e degli stipendi. Nota: Ai fini didattici si è deciso di popolare già la classe azienda con dei dipendenti, i quali hanno già delle vendite e degli stipendi.

La classe `Dipendente` rappresenta il dipendente del concessionario e contiene una lista di `Mezzo*` e metodi utili alla modifica dello stipendio, della valutazione e dei mezzi venduti. I dipendenti possono avere ID uguale in questa gerarchia: sarà cura di una possibile classe che implementa la lettura/scrittura da file dei dipendenti verificare se ci siano dipendenti con ID uguale.

DESCRIZIONE CHIAMATE POLIMORFE

Nella classe `InserisciVendita` nel metodo `aggiornaStelline()` viene creato un `Mezzo*`, su cui poi si esegue `mezzo->getStar()`, che in base al tipo di mezzo creato sopra ritorna la giusta quantità di stelline da visualizzare poi nella GUI.

Quando un utente clicca sulla toolbar per salvare le vendite viene chiamato

`MostraVendite::savePressedSlot` dove viene chiesto all'azienda di salvare tutte le vendite. L'azienda chiama per ogni dipendente `assegnaReward` che (indirettamente) usa `getReward()` per ottenere da tutti i mezzi il bonus vendita e `assegnaRating` che usa `getStar()` per ottenere da tutti i mezzi le stelline. `getReward` e `getStar` sono decisi in base al tipo dinamico e hanno comportamenti diversi dipendentemente dal tipo di mezzo.

DESCRIZIONE GUI

Per gestire il cambio di schermate si è pensato di creare una classe `PageHandler`: essa crea l'azienda e le varie schermate da mostrare, mette poi le schermate in uno `QStackedWidget`. Con il `QStackedWidget` si riesce a cambiare la schermata visualizzata tramite segnali e slot che collegano i pulsanti di cambio schermata con slot in `PageHandler`, dove viene cambiato l'indice del `QStackedWidget`. Le classi `InserisciVendita`, `CalcolaStipendi` e `MostraVendite` sono costruite con l'azienda creata nel `PageHandler`, in modo che possano accedere ad essa tramite i metodi proposti da `Azienda`.

Ogni classe interagisce con le altre tramite segnali e slot connessi da `PageHandler`. In particolare sono stati necessari per fare i refresh quando vengono modificati i dati in `Azienda`.

MANUALE UTENTE

L'applicazione nella schermata principale presenta tre pulsanti, tramite i quali si possono raggiungere le tre seguenti schermate:

Aggiungi vendita: questa schermata serve per aggiungere una vendita fatta dal concessionario, inserendo i vari parametri richiesti. Prima di cliccare su salva si può avere una preview del rating che si verrà ad assegnare al venditore per aver venduto quel mezzo. Il rating è deciso in base alle caratteristiche del mezzo stesso.

Calcola stipendi: questa pagina mostra gli stipendi del mese corrente dei dipendenti, con a fianco il loro rating di vendita. Il pulsante presente permette di azzerare definitivamente tutti gli stipendi e i rating.

Mostra vendite: Qua sono presenti le vendite non ancora salvate. Tramite il pulsante possono essere confermate definitivamente e quindi si va a modificare lo stipendio e il rating del dipendente oggetto della vendita.