Relazione progetto di Programmazione ad Oggetti

Francesco Bugno 1170991

Settembre 2021

Contents

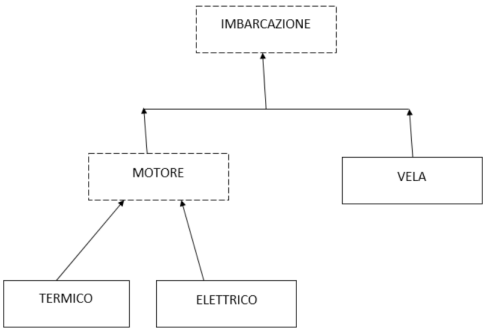
1 Descrizione

Il progetto consiste nel creare un gestionale per l'azienda MargheraBoat.
Questo programma consente l'inserimento di nuove imbarcazioni, la modifica delle caratteristiche, la rimozione, la visualizzazione delle caratteristiche, la ricerca secondo alcuni criteri e per finire il calcolo di alcune funzionalità utili.

2 Progettazione

2.1 Gerarchia

La classe base astratta Imbarcazione rappresenta la classe base della gerarchia e contiene al suo interno dei campi privati comuni a tutte le imbarcazioni. Come sottoclassi dirette della base astratta abbiamo rispettivamente Motore e Vela. Motore è una classe astratta che rappresenta tutte le imbarcazioni che come tipologia di propulsione utilizzano un motore. Derivate dalla classe Motore abbiamo le due classi concrete Termico ed Elettrico che implementano le caratteristiche diverse di categoria di motore presenti in azienda. Vela invece è una classe concreta che rappresenta tutte le imbarcazioni che come tipologia di propulsione utilizzano una vela per navigare.



Imbarcazione Classe base astratta della gerarchia:
fornisce le informazioni comuni a tutte le imbarcazioni, con i suoi campi dati:
nome, *cantiere*, *velocità* (nodi), *peso* e *lunghezza*.

Motore Classe astratta derivata da Imbarcazione:
fornisce in aggiunta le informazioni delle barche a motore, con i campi dati:
numero motori, *potenza singolo motore* (cv), *tipologia motore* (che può essere: entrobordo, fuoribordo, entrofuoribordo) .

Termico Classe concreta derivata da Motore:
gli oggetti rappresentano le imbarcazioni a motore termico, con l'aggiunta dei campi dati:
consumi (L/h), *capienza serbatoi* (L), *tipologia di carburante* (che può essere: diesel o benzina).

Elettrico Classe concreta derivata da Motore:
gli oggetti rappresentano un'imbarcazione a motore elettrico, con l'aggiunta dei campi dati:
consumi (Kw/h), *capienza batteria* (Kw), *tipologia di batteria* (che può essere: litio o piombo).

Vela Classe concreta derivata da Imbarcazione:
gli oggetti rappresentano un'imbarcazione a vela, con l'aggiunta dei campi dati:
numero vele, *motore ausiliario* (che puè essere: si o no), *potenza motore ausiliario* (cv).

2.2 Container

Il Container e' una classe contenitore templetizzata, che implementa un array dinamico, con iteratori, aggiunta, rimozione e altre funzionalità. E' stato implementato inoltre un template di classe DeepPtr che opera in modalità profonda. Il contenitore conterrà quindi oggetti di tipo DeepPtr che contengono un campo dati di tipo Imbarcazione* polimorfo.

2.3 GUI

Nella cartella Vista ci sono tutti i file relativi alla GUI, creati usando la libreria Qt.
Nella classe MainWindow, che si avvia all'inzio del programma, ci sono i vari widget di QT per la GUI. Come ad esempio:

MainLayout Layout principale che contiene l'accesso alle varie funzionalita', la lista delle imbarcazioni e i dettagli dell'imbarcazione selezionata.

MenuBar Barra di menu in alto che contiene alcune funzionalità tra cui: info generali su tutte le barche del sistema e informazione sull'autore.

Modify/Insert Layout Layout che appaiono quando si vuole aggiungere o modificare un'imbarcazione alla lista delle imbarcazioni presenti nel sistema.

2.4 MVP

Si è scelto il pattern architetturale Model-View-Presenter in cui "MainWindow" funziona da presenter e collega il modello alla vista vera e propria.
La classe MainWindow interagisce con la classe Model e le varie classi della View (MenuBar, MainLayout, InsertLayout, ModifyLayout), gestendo la parte di signals e slot.

3 Chiamate Polimorfe

virtual std::string tipoPropulsione() const = 0 Questa funzione polimorfa ritorna una stringa contenente il nome del tipo di propulsione utilizzata dall'imbarcazione.

virtual std::string boat_toString() const Questa funzione polimorfa ritorna una stringa contentente i dettagli dell'imbarcazione oggetto di invocazione.

virtual float calcoloConsumi(float) const = 0 Questa funzione polimorfa ritorna il quantitativo di carburante (L) o il quantitativo di carica (Kw) necessario per navigare il *tempo* richiesto dall'utente.

virtual float calcoloAutonomia(unsigned int) const = 0 Questa funzione polimorfa ritorna il tempo di autonomia dell'imbarcazione dato il *carburante* inserito dall'utente.

virtual bool patenteNautica() const = 0 Questa funzione polimorfa ritorna true se per il tipo di imbarcazione selezionata è richiesta la patente nautica per navigare oppure ritorna false se non è richiesta.

virtual void print() const Stampa l'imbarcazione oggetto di invocazione in std::cout.

virtual Imbarcazione* clone() const = 0 Funzione polimorfa che restituisce un puntatore ad una copia dell'oggetto di invocazione.

Inoltre Imbarcazione ha un distruttore virtuale pubblico che permette l'invocazione del giusto distruttore da parte di qualsiasi oggetto polimorfo.

4 Eccezioni

Riguardo alle eccezioni ho creato una classe *BoatException*. Tutte le eccezioni sono state implementate con *QMessageBox*.

5 Tempistiche

- Analisi del problema: 3h;
- Progettazione modello e GUI: 10h
- Apprendimento libreria Qt: 4h
- Codifica modello e GUI: 30h
- Debugging: 4h
- Testing: 3h

6 Note finali

6.1 Note di Compilazione

Il sistema è una versione minimale del sistema Linux Ubuntu installato nelle macchine dei laboratori. Il progetto e' stato sviluppato con Qt 5.9.5, utilizzando l'IDE Qt Creator sulla macchina virtuale rilasciata dal professore. Il progetto prevede l'utilizzo di un file MargheraBoat.pro diverso da quello ottenibile con qmake -project, a causa della presenza di funzionalità di c++11. Bisognerà quindi aprire il terminale sulla cartella di progetto ed eseguire i seguenti comandi:

- qmake
- make

Eseguendo questo operazioni si potrà poi eseguire il file eseguibile con il comando: ./MargheraBoat

6.2 Manuale utente

L'applicazione si compone di :

- Un menu bar da cui si può vedere le info generali delle imbarcazioni nel sistema e vedere chi è l'autore dell'applicazione.
- Una barra di ricerca, per cercare le imbarcazioni nel sistema in base ad alcune caratteristiche, ad esempio nome, cantiere, velocità, peso e lunghezza.
- Vari pulsanti come Modifica, Elimina, Inserisci, per appunto modificare/eliminare o inserire una nuova imbarcazione nel sistema.
- Pulsanti come Svuota, che svuota del tutto la lista di imbarcazioni nel sistema; TipoImbarcazione che ci dice a quale categoria appartiene l'imbarcazione in base alla sua lunghezza; VerificaRichiestaPatente che ci dice se è richiesta o meno la patente nautica per utilizzare una data imbarcazione.
- Le varie funzionalità come calcoloAutonomia, calcoloConsumi, CalcoloMiglia, calcoloTempo che ci restituiscono le informazioni in base ai parametri da noi inseriti.
- Infine quando modifichiamo o aggiungiamo un'imbarcazione, si aprirà una finestra in cui bisogna specificare i vari parametri.