

QPhotoStudio

Progetto del corso "Programmazione ad Oggetti" - A.A. 2019-2020

alessio.barbiero.1@studenti.unipd.it

Relazione QPhotoStudio

Si riportano i nominativi dei componenti del gruppo che hanno partecipato alla realizzazione del progetto, con a seguito la rispettiva matricola:

Nominativo	Matricola
Alessio Barbiero	1136893
Ecaterina Bosinceanu	1169669
Matteo Lattanzio	1169566

1 Descrizione del progetto

Nello svolgimento del progetto didattico si è scelto di concentrare il lavoro di gruppo nella realizzazione di un software per la gestione di un listino di vendita. Nasce così *QPhotoStudio*, un programma basato sulla modellazione del database di un negozio dedicato alla vendita di articoli fotografici, che variano dai corpi macchina fino ai vari accessori utili.

Ogni prodotto memorizzato nel listino è caratterizzato dalla marca della casa di produzione, dal modello che lo identifica e dal prezzo di vendita al pubblico.

L'interfaccia di *QPhotoStudio* permette di interagire con il listino sopra descritto per simulare il reale utilizzo in un negozio dedicato; è infatti possibile aggiungere nuovi prodotti, specificandone gli attributi necessari, modificare i dati specifici di un prodotto o rimuoverlo dalla lista. Sono inoltre disponibili due funzionalità di ricerca, basate sulla marca di produzione o sul modello del prodotto, oltre alla possibilità di visualizzare il listino filtrandolo per la tipologia di prodotto desiderata.

Il contenuto della raccolta può essere esportato in due differenti formati, CSV ed XML, con lo scopo di fornire rispettivamente una versione "lite" e leggibile tramite i più comuni editor di file o una versione caricabile nel sistema e quindi utilizzabile come file di salvataggio.

2 Note tecniche

2.1 Realizzazione

Il software è stato sviluppato sfruttando i diversi dispositivi appartenenti ai membri del gruppo. Sono stati utilizzati i sistemi operativi Windows (versione 10 Home) e Linux Ubuntu (versione 20.04) e l'IDE QtCreator nella sua versione 5.14.2. Nel codice sono presenti alcune funzionalità peculiari di C++ 11 che ne richiedono una corretta compilazione.

2.2 Compilazione

Per la compilazione è necessario l'utilizzo del file `QPhotoStudio.pro` incluso nella directory di progetto. Per la compilazione e l'esecuzione del software è necessario eseguire da terminale i comandi seguenti:

```
qmake -o Makefile QPhotoStudio.pro
make
./QPhotoStudio
```

Nella directory secondaria *output* vengono inoltre forniti i file *QPhotoStudio.xml* e *QPhotoStudio.CSV*. Il primo può essere caricato nel sistema tramite l'apposita funzionalità per interagire con una listino già popolato; il secondo file fornisce l'esportazione più leggibile del listino sopracitato, ottenuta tramite l'apposita funzione del software.

3 Gerarchia delle classi

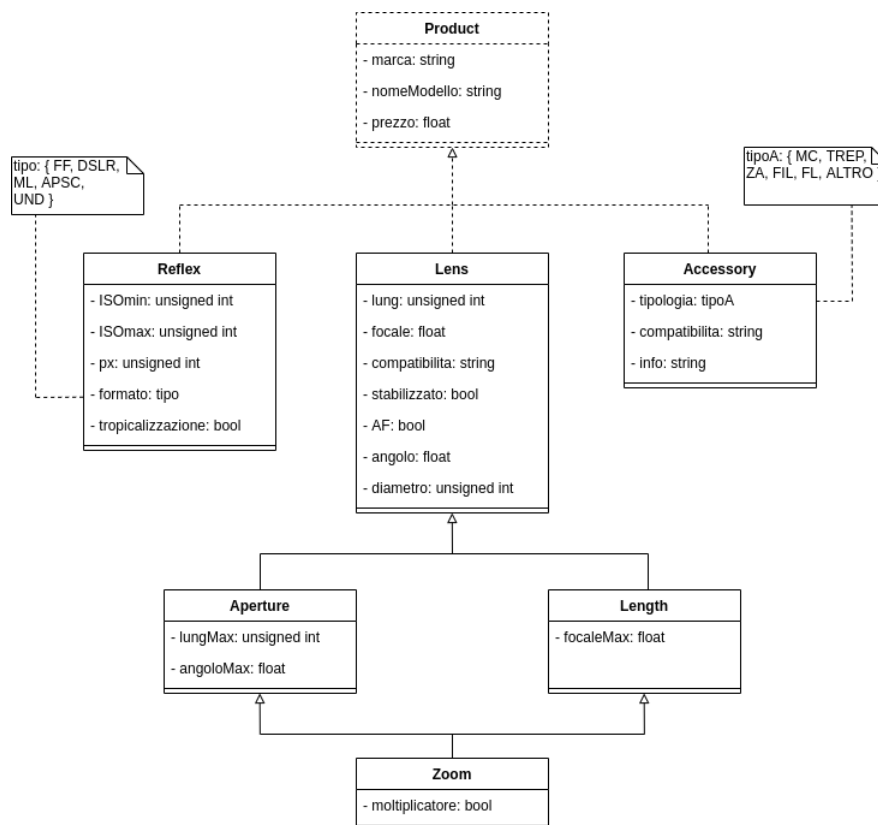


Diagramma delle classi

La gerarchia delle classi si sviluppa su quattro livelli. La classe **Product** costituisce la base della gerarchia e rappresenta un generico prodotto. Si tratta di una classe astratta, in quanto comprende metodi virtuali puri, ed essa non viene mai istanziata. Ogni prodotto è caratterizzato dalla marca della casa di produzione, dal nome del modello e dal prezzo di vendita al pubblico. Il seguente livello gerarchico contiene le tre classi che rappresentano le macro-categorie di prodotti a listino: **Reflex**, **Lens** e **Accessory**. Queste estendono la classe base e ne concretizzano i metodi virtuali. Nel dettaglio sono così strutturate:

- **Reflex** (corpo macchina): rappresenta una macchina fotografica vendibile dal negozio. Ognuna di esse è caratterizzata da:
 - ISOmin e ISOmax, cioè i valori minimi e massimi di sensibilità alla luce;
 - tropicalizzazione, che indica la presenza o meno sul corpo macchina del trattamento per resistere agli agenti atmosferici;
 - px, ovvero i megapixels del sensore della macchina;
 - formato, un attributo di tipo che indica la tipologia di sensore della macchina.
- **Lens** (obiettivo fisso): rappresenta un obiettivo ottico, anch'esso vendibile dal negozio. Ogni obiettivo possiede i seguenti attributi:
 - lung, ovvero la lunghezza focale della lente;
 - focale, indicante il valore del diametro di apertura focale dell'obiettivo ed esprime la sua capacità di catturare la luce in ingresso;
 - compatibilità, cioè un'informazione sulle case produttrici dei corpi macchina compatibili con la lente;
 - stabilizzazione, che indica la presenza o meno del sistema meccanico di riduzione delle oscillazioni;
 - AF (Automatic Focus), che indica la presenza o meno del sistema automatico di messa a fuoco;
 - angolo, ovvero l'angolo di inquadratura fornito dall'obiettivo;
 - diametro, rappresentante il diametro della lente misurato al suo estremo.

Un obiettivo base viene indicato anche come **Fisso** in quanto possiede un unico valore di apertura focale e lunghezza focale (e di conseguenza di diametro dello stesso).

- **Accessory** (accessorio): cioè un prodotto generico che rappresenta diversi oggetti utili al fotografo o all'amatore. Presenta alcuni attributi:
 - tipologia, attributo di tipo che indica se l'accessorio è un flash, un treppiede, una scheda di memoria o altri tipi definiti;
 - compatibilità, cioè un'informazione sulle case produttrici dei corpi macchina compatibili con l'accessorio;
 - info, eventuali note aggiuntive interessanti riguardo il prodotto.

Il secondo livello gerarchico contiene le specializzazioni della classe **Lens**. Un obiettivo infatti non è solamente "Fisso" ma può presentare valori variabili di apertura focale o di lunghezza focale. Nascono così le due classi figlie dirette della classe obiettivo:

- **Aperture** (obiettivo a focale fissa): essi presentano un valore costante solamente nel campo dell'apertura focale, mentre la loro lunghezza è variabile. Questo implica l'utilizzo di due nuovi attributi:
 - lungMax, ovvero la lunghezza focale massima della lente. In questo caso, l'attributo *lung* della classe base indica la lunghezza minima dell'obiettivo;
 - angoloMax, ovvero l'angolo di inquadratura più ampio fornito dall'obiettivo. In questo caso, l'attributo *angolo* della classe base indica l'angolo di inquadratura minore.
- **Length** (obiettivo a lunghezza fissa): essi possiedono un valore variabile di apertura focale, mentre la loro lunghezza è fissa. Possiede un solo attributo aggiuntivo:
 - focaleMax, ovvero il valore massimo del diametro di apertura focale dell'obiettivo. In questo caso, l'attributo *focale* della classe base indica il diametro di apertura minore.

Esistono però obiettivi che presentano valori variabili in entrambi gli attributi e perciò si ha la necessità di sfruttare l'ereditarietà multipla: la classe **Zoom** nasce per rappresentare gli obiettivi con valore variabile sia in lunghezza che in apertura focale ed è figlia di entrambe le classi descritte al livello sovrastante. Inoltre, viene utilizzato un ulteriore attributo chiamato moltiplicatore, indicante la compatibilità dell'obiettivo con un moltiplicatore di focale esterno in grado di estenderne la portata.

4 Polimorfismo

La classe Product presenta i seguenti metodi virtuali (puri e non):

- **virtual ~Product() = default:** distruttore di default della classe Product;
- **virtual Product* clone() const = 0:** restituisce un puntatore ad una copia dell'oggetto su cui viene effettuata la chiamata della funzione;
- **virtual string print() const:** ritorna una stringa contenente i dettagli dell'oggetto per un'eventuale stampa;
- **virtual string getType() const = 0:** restituisce un oggetto di tipo stringa indicante la sottoclasse di appartenenza dell'oggetto su cui viene invocato il metodo;
- **virtual string CSV() const:** metodo per esportare l'oggetto in formato CSV.
- **virtual string XML() const:** metodo per esportare l'oggetto in formato XML.

5 Container

Il progetto comprende una classe contenitore template, ovvero Container. Essa è costituita da una Doubly Linked List di oggetti di tipo DeepPtr; questa struttura è stata scelta con lo scopo di permettere la rimozione e l'aggiunta di elementi in punti casuali della lista. DeepPtr è invece una classe di puntatori smart ad oggetti di tipo Product, consentendo operazioni di gestione profonda.

6 Esportazione ed importazione

6.1 XML

Il formato XML viene utilizzato per il salvataggio e il caricamento di tutte le informazioni sui prodotti memorizzati. Nella sezione File del menù è possibile selezionare la voce Salva per esportare il listino in un file XML, specificandone il nome ed il percorso di salvataggio. La voce Carica consente invece di caricare il contenuto di un file XML precedentemente esportato dal sistema, ripristinando il listino.

Queste funzionalità sono implementate dai metodi *save* e *load* nella classe MyXml. Di seguito viene fornito un esempio di oggetto esportato in XML.

```
<Product>
  <Accessorio>
    <Marca>SanDisk</Marca>
    <Modello>SDXC 128GB</Modello>
    <Prezzo>55</Prezzo>
    <Tipo>MemoryCard</Tipo>
    <Compatibilita>Tutte</Compatibilita>
    <Note>Velocità di scrittura: 120Mb/s</Note>
  </Accessorio>
</Product>
```

6.2 CSV

Il formato CSV consente di esportare le informazioni del listino in un formato facilmente leggibile dall'utente e consultabile tramite software esterni (ad esempio Microsoft Excel). Questo metodo di esportazione non include tutte le informazioni sui prodotti ma solo alcuni dettagli basilari e **non può essere utilizzato per importare i dati nel sistema**. Il file .csv viene salvato all'interno della directory *output* con il nome *QPhotoStudio.csv*.

7 Interfaccia utente

L'interfaccia grafica è implementata tramite una serie di file specifici per la gestione delle singole finestre del programma o di alcune particolari funzionalità. Troviamo quindi:

- *mainwindow.cpp*: contiene la classe utilizzata per la gestione di tutte le componenti dell'interfaccia sia dal punto di vista grafico che da quello funzionale;
- *Add.cpp*: contiene la classe dedicata all'aggiunta di nuovi prodotti al listino. Presenta un form che permette di inserire tutti i parametri dell'oggetto e i cui campi variano in base alla tipologia di oggetto che dev'essere inserito. Viene inoltre mostrata un'immagine differente per ogni tipo di prodotto che si vuole inserire;
- *Details.cpp*: contiene la classe che permette la visualizzazione di tutti i dettagli del prodotto selezionato. Consente inoltre la modifica dei campi dell'oggetto o la sua totale rimozione dal listino;
- *Index.cpp*: contiene la homepage del programma. Essa è caratterizzata da tre pulsanti ("Inserisci", "Ricerca" e "Listino") e da una barra menù, contenente le principali azioni eseguibili;
- *Listino.cpp*: contiene la classe utilizzata per mostrare in forma tabulare tutti i prodotti contenuti nel listino. Consente, tramite appositi pulsanti, di filtrare i prodotti per le categorie di appartenenza (Reflex, Obiettivi e Accessori);
- *Search.cpp*: contiene la classe utilizzata per gestire la ricerca di prodotti a listino. Questa può essere effettuata ricercando una parola chiave nel modello o nella marca del prodotto, tramite l'apposito filtro.
- *Model.cpp*: contiene la classe utilizzata per la gestione dell'oggetto Container. Essa rappresenta il reale punto d'interazione tra la struttura interna della gerarchia e l'interfaccia grafica implementata;
- *MyXml.cpp*: contiene la classe incaricata di gestire l'IO da file XML, sfruttando l'opportuna libreria adottata;
- *MyCSV.cpp*: contiene la classe incaricata di gestire l'output su file CSV tramite gli opportuni metodi di IO.

Tutte le impostazioni grafiche sono state invece gestite mediante un foglio di stile esterno (*style.qss*).

8 Suddivisione del lavoro progettuale

Tutti i membri del gruppo hanno partecipato attivamente alla realizzazione di ogni classe, incaricandosi delle attività di codifica, testing e debug. La tabella serve a specificare le sezioni in cui ogni membro ha concentrato maggiormente il proprio lavoro personale.

Tabella dei ruoli assegnati

Attività	Assegnazione
Analisi dei requisiti	Matteo Lattanzio
Modellazione della gerarchia	Alessio Barbiero
Exception handling	Matteo Lattanzio
Realizzazione delle classi container	Ecaterina Bosinceanu
Realizzazione del modello	Alessio Barbiero
Realizzazione della homepage	Ecaterina Bosinceanu
Realizzazione delle sezioni di listino, ricerca e dettaglio	Alessio Barbiero
Realizzazione della sezione di inserimento	Matteo Lattanzio
Debug e correzione degli errori, controlli di input	Ecaterina Bosinceanu
Icone, immagini e diagrammi	Ecaterina Bosinceanu
Realizzazione del foglio di stile	Matteo Lattanzio

Tabella del lavoro individuale di Alessio Barbiero

Attività	Ore
Brainstorming sulle possibili realtà da modellare	2h
Analisi dei requisiti	3h
Modellazione gerarchie	6h
Debug gerarchie	1h
Studio delle classi utili al contenitore	1h
Studio individuale delle librerie Qt	4h
Ricerca delle librerie esterne da utilizzare	1h
Progettazione GUI	3h
Realizzazione componenti GUI	11h
Debug GUI	4h
Testing generale delle funzionalità	3h
Stesura relazione individuale	4h
Totale ore individuali impiegate	43h

La quantità di ore individuali impiegata nella realizzazione del software è fortemente influenzata dall'aver sostenuto la consegna del progetto didattico anche nell'A.A. 2018/2019. Per questo

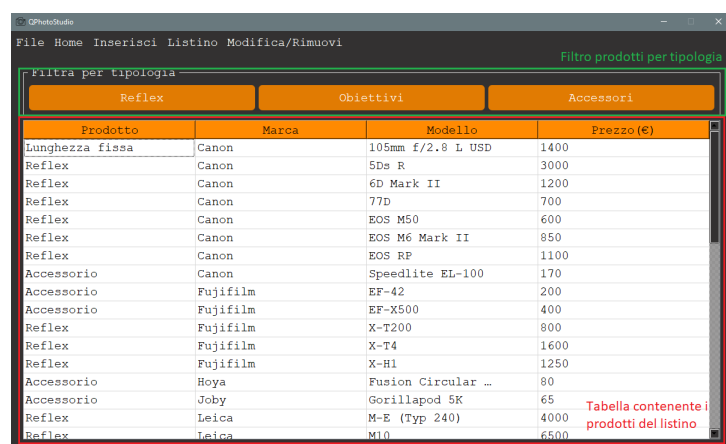
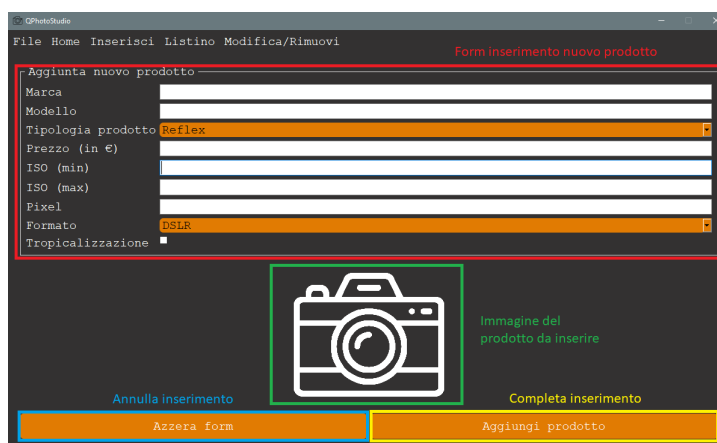
motivo, la gestione già nota di alcune criticità e lo studio pregresso delle librerie Qt hanno consentito di impiegare un monte ore inferiore a quello dei colleghi, lasciando spazio alla rifinitura di alcuni dettagli secondari.

9 Manuale d'utilizzo



Homepage di QPhotoStudio. I pulsanti e il menù di navigazione consentono di accedere alle funzioni principali del software. La sezione "File" permette di gestire il salvataggio ed il caricamento del listino. In ogni momento è possibile terminare l'esecuzione del software cliccando sul pulsante di chiusura della finestra; prima di confermare l'uscita, un messaggio chiederà se si intende salvare i dati prima di uscire.

Sezione di inserimento di un nuovo prodotto. Compilando i campi correttamente è possibile inserire un nuovo prodotto a listino, cliccando il pulsante di conferma. Se i dati inseriti non sono corretti, l'utente visualizza un messaggio di errore. Il pulsante "Azzera form" svuota i campi inseriti.



Listino di QPhotoStudio. Qui vengono visualizzati, ordinati alfabeticamente per marca e modello, i prodotti memorizzati nel listino. I tre pulsanti consentono di filtrare i prodotti in base alla loro categoria e di visualizzarne i dettagli specifici. Per tornare al listino generico è sufficiente cliccare nuovamente sull'apposita voce del menù di navigazione.

Pagina di modifica/rimozione di un prodotto dal listino. Questa sezione consente inoltre la ricerca un prodotto per marca o modello. In entrambi i casi vengono mostrati tutti i prodotti che, nella sezione specificata, includono la keyword richiesta; il tasto "Ripristina" azzerà i risultati di ricerca. Facendo doppio click su un prodotto è possibile accedere alla sezione di modifica descritta di seguito.

Prodotto	Marca	Modello	Prezzo (€)
Lunghezza fissa	Canon	105mm f/2.8 L USD	1400
Reflex	Canon	5Ds R	3000
Reflex	Canon	6D Mark II	1200
Reflex	Canon	77D	700
Reflex	Canon	EOS M50	600
Reflex	Canon	EOS M6 Mark II	850
Reflex	Canon	EOS RP	1100
Accessorio	Canon	Speedlite EL-100	170
Accessorio	Fujifilm	EF-42	200
Accessorio	Fujifilm	EF-X500	400
Reflex	Fujifilm	X-T200	800
Reflex	Fujifilm	X-T4	1600
Reflex	Fujifilm	X-H1	1250
Accessorio	Hoya	Fusion Circular ...	80
Accessorio	Joby	Gorillapod 5K	65
Reflex	Leica	M-E (Typ 240)	4000
Reflex	Leica	M10	6500

Modifica elemento selezionato

Rimuovi elemento selezionato

Ripristina

Sezione di dettaglio che consente la modifica delle informazioni di un prodotto o la rimozione dello stesso dal listino. Questa finestra viene aperta facendo doppio click sul corrispondente elemento visualizzato nella pagina sopracitata. Per annullare le operazioni è sufficiente chiudere la finestra; anche in questo caso il tentativo di inserire valori inconsistenti nei campi porta a visualizzare un messaggio di errore.