

# Relazione progetto Qt

## 1 INTRODUZIONE

---

Lo scopo di questo progetto è la realizzazione di un contenitore per la gestione di un piccolo database in C++ e la conseguente realizzazione di una interfaccia grafica per la gestione del database. Sono state implementate le funzioni di inserimento, ricerca e modifica di elementi.

Il sistema operativo usato durante il progetto è Windows 8.1 su compilatore “Qt 4.7.4 for Desktop(MinGW)” ed utilizzo dell’IDE di sviluppo Qt Creator.

## 2 CONTAINER<T>

---

In “vettore.h” è stato implementato un contenitore che cerca di emulare il comportamento del *vector* della libreria standard STL per avere inserimento e rimozione in coda in tempo costante e accesso casuale agli elementi.

Si usano due parametri *size* e *capacity* per tenere conto rispettivamente del numero di elementi attuali presenti nel vettore e del numero totale di elementi che il vettore può contenere. Quando *size==capacity* il vettore viene ridimensionato al doppio della dimensione. Sono stati creati i metodi di inserimento *push\_back()* e rimozione *pop\_back()* in coda, oltre che ad inserimento e rimozione in una determinata posizione.

Sono stati resi disponibili anche degli iteratori per scorrere il vettore creato.

Non avendo una struttura a nodi con dei campi riferimento da aggiornare in modo preciso non è stata ritenuta indispensabile la realizzazione di Smart Puntatori.

### 3 GERARCHIA

---

E' stata realizzata una semplice gerarchia con una classe astratta *Bici* con un metodo virtuale puro *prezzo()* che calcola il prezzo della bici in base a diversi parametri. Da *Bici* derivano due classi concrete *MTB* e *Strada* che implementano il metodo puro della classe base ed hanno metodi e parametri propri.

E' stata inoltre definita la classe *Shop* che contiene un vettore di puntatori a biciclette e che implementa i metodi di inserimento, rimozione e ricerca delle biciclette nel vettore stesso.

La ricerca è stata effettuata in questa classe, prendendo una stringa restituita dai metodi delle classi grafiche Qt ed in base a quella cerca le bici corrispondenti a tali parametri di ricerca, ovvero gli attributi delle biciclette, restituendo un vettore di indici a tali biciclette che verranno poi visualizzate nella tabella dell'interfaccia grafica.

### 4 INTERFACCIA GRAFICA

---

L'interfaccia grafica presenta un pulsante di menù *File* che permette di attivare le finestre di dialogo per ricerca e inserimento di nuove biciclette. Sotto al pulsante è posizionata la tabella che mostra tutte le biciclette presenti nel vettore di *Shop*. Sotto la tabella è presente una status bar che contiene i pulsanti *Elimina*, che elimina la bicicletta relativa alla riga della tabella che deve essere precedentemente selezionata, *Aggiorna*, che aggiorna la tabella con le biciclette attualmente (al momento del click) presenti nel vettore, ed *Esci* che chiude l'applicazione.

Le finestre di dialogo permettono l'inserimento di una nuova bici (*nuovaBici.h*) con i parametri appropriati in base al tipo della bici, mentre la ricerca (*cercaBici.h*) permette di decidere in base a quale attributo statico delle biciclette effettuare la ricerca.

Le biciclette iniziali sono inizialmente caricate nel main dell'applicazione.