qDB

Progetto di Programmazione ad Oggetti, a.a. 2013/2014

prof. Francesco Ranzato

1 Scopo

Lo scopo del progetto qDB è lo sviluppo in C++/Qt di un sistema minimale per la gestione di un (piccolo) database tramite una interfaccia utente grafica.

2 Template di classe `Container<K>`

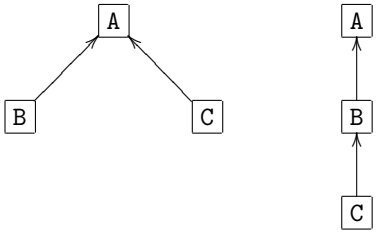
Definire un template di classe `Container<K>` i cui oggetti rappresentano un contenitore di oggetti di tipo `K`. Il template di classe `Container<K>` deve fornire almeno delle funzionalità minime di: (1) inserimento, (2) rimozione, (3) ricerca, (4) modifica.

Vi è un unico vincolo da rispettare nello sviluppo del template di classe `Container<K>`: **non** è permesso l'utilizzo dei contenitori della libreria STL.

Si scelga l'implementazione di `Container<K>` ritenuta più opportuna per usare il contenitore come struttura dati per il database.

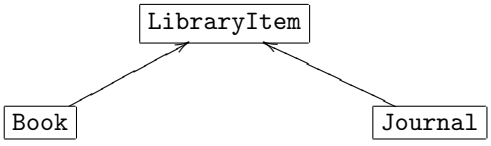
3 Gerarchia di classi

Definire una gerarchia di classi G composta almeno da tre classi `A`, `B` e `C` che modellano una realtà di oggetti da gestire tramite il database. Le gerarchie possibili per le tre classi `A`, `B` e `C` sono le seguenti:



La classe `A` è quindi superclasse di ogni altra classe della gerarchia G.

Ad esempio, nel campione di codice che accompagna il presente documento, la gerarchia di classi è la seguente:



Le classi della gerarchia dovranno essere dotate di opportune e realistiche interfacce pubbliche. Usare la fantasia ed ispirarsi ai propri interessi nello scegliere la realtà da modellare mediante questa gerarchia di classi. Naturalmente saranno accettate anche gerarchie più complesse che contengono piú di tre classi.

Al fine di popolare un contenitore `Container<K>` con oggetti della gerarchia di classi G, si valuti l'opportunità di definire una classe `SmartAptr` di puntatori smart alla classe base `A` di G — che avrà quindi come campo dati un puntatore polimorfo ad `A` — e conseguentemente di popolare il contenitore con oggetti di `SmartAptr`.

4 Interfaccia Grafica

Si richiede di sviluppare una GUI usando la libreria Qt che permetta all'utente di gestire agevolmente il database di oggetti delle classi della gerarchia G. Si tratta quindi di sviluppare una GUI per la gestione di un contenitore C di puntatori (smart) polimorfi alla superclasse `A` della gerarchia G, che come requisito minimale permetta:

(1) inserimenti

(2) rimozioni

(3) ricerche

(4) modifiche

di oggetti della gerarchia G nel database rappresentato dal contenitore C. In particolare, si valuti l'opportunità che le interrogazioni (ovvero le ricerche) del database possano offrire delle ricerche basate su alcune delle funzionalità pubbliche offerte dalle classi in G (ad esempio, ricerche per: anno di pubblicazione di un LibraryItem, titolo di un Book, numero di articoli in un Journal, etc). **Opzionalmente** la GUI dovrà permettere di eseguire:

(5) caricamento da file

(6) salvataggio su file

del database rappresentato dal contenitore C.

Si valuti l'opportunità di aderire al design pattern Model-View-Controller per la progettazione architetturale della GUI. Il campione di codice della GUI che accompagna e complementa il presente documento fornisce una intelaiatura minimale a meri fini esemplificativi per lo sviluppo della GUI. Come noto, la libreria Qt è dotata di una documentazione completa e precisa che sarà la principale guida di riferimento nello sviluppo della GUI, oltre ad offrire l'IDE QtCreator ed il tool QtDesigner. La libreria Qt offre una moltitudine di classi e metodi per lo sviluppo di GUI curate, dettagliate e user-friendly.

5 Valutazione del Progetto

Un buon progetto dovrà essere sviluppato seguendo i principi fondamentali della programmazione orientata agli oggetti, anche per quanto concerne lo sviluppo dell'interfaccia grafica. La valutazione del progetto prenderà in considerazione i seguenti criteri:

1. **Correttezza**: il progetto deve compilare e funzionare correttamente, e raggiungere correttamente e pienamente gli scopi previsti.
2. **Orientazione agli oggetti**: (A) progettazione ad oggetti, (B) modularità (in particolare, massima separazione tra il codice logico del progetto ed il codice della GUI del progetto), (C) estensibilità e (D) qualità del codice sviluppato.
3. **Quantità e qualità**: quante e quali funzionalità il progetto rende disponibili, e la loro qualità.
4. **GUI**: utilizzo corretto della libreria Qt, qualità ed usabilità della GUI.

6 Esame Orale e Registrazione Voto

La partecipazione all'esame orale è possibile solo dopo:

1. avere superato con successo (cioè, con voto $\geq$ 18/30) l'esame scritto
2. avere consegnato il progetto entro la scadenza stabilita
3. essersi iscritti alla lista Uniweb dell'esame orale

Il giorno dell'esame orale (nel luogo ed all'orario stabiliti) verrà comunicato l'esito della valutazione del progetto. Tre esiti saranno possibili:

(A) Valutazione positiva del progetto con registrazione del voto complessivo proposto **con esenzione dell'esame orale**. Nel caso in cui il voto proposto non sia ritenuto soddisfacente dallo studente, sarà possibile richiedere l'esame orale, che potrà portare a variazioni in positivo o negativo del voto proposto.

(B) Valutazione del progetto da completarsi con un **esame orale obbligatorio**. Al termine dell'esame orale, o verrà proposto un voto complessivo sufficiente oppure si dovrà riconsegnare il progetto per un successivo esame orale.

(C) Valutazione negativa del progetto che comporta quindi la **riconsegna del progetto** per un successivo esame orale (il voto dell'esame scritto rimane valido).

Si ricorda inoltre che all'eventuale esame orale lo studente dovrà saper motivare **ogni** scelta progettuale e dovrà dimostrare la **piena conoscenza** di ogni parte del progetto.

7 Regole

Il presente documento va inteso come una "specifica minimale" di progetto, ossia tutto ciò che non è espressamente richiesto è lasciato a libera scelta. Il progetto dovrà essere realizzato da ogni singolo studente in modo **indipendente** da terze persone.

7.1 Relazione

Il progetto dovrà essere obbligatoriamente accompagnato da una **breve** (massimo 6 pagine in formato 10pt) relazione scritta che descriva **sinteticamente** le scelte progettuali ritenute più significative. La relazione deve essere presentata come un file PDF di nome (preciso) `relazione.pdf`. La relazione deve anche specificare il sistema operativo di sviluppo e le versioni precise del compilatore e della libreria Qt.

7.2 Compilatore e libreria Qt

Il progetto deve compilare ed eseguire correttamente sulle macchine **Linux** del laboratorio informatico del plesso Paolotti o della torre Archimede con il compilatore GNU `g++` 4.x. Nelle macchine Linux del laboratorio è installata la libreria Qt nella versione 4.8 (5.2 è invece l'ultima versione di Qt), si veda `http://www.studenti.math.unipd.it/index.php?id=corsi#c544`. È naturalmente possibile sviluppare il progetto su altri sistemi operativi come MacOS/Windows. In tal caso, prima di consegnare il progetto, ricordarsi di effettuare (anche remotamente tramite `ssh`) una prova di compilazione, esecuzione e funzionamento sulle macchine Linux del laboratorio.

7.3 Cosa consegnare

Tutti i file sorgente `.h` e `.cpp`, il file `relazione.pdf` contenente la relazione, eventuali file che memorizzano dati necessari per il corretto funzionamento del programma (ad esempio, un file contenente un database di prova). Se la compilazione del progetto necessita di un project file (.pro) per `qmake` diverso da quello ottenibile tramite l'invocazione di `qmake -project` allora deve anche essere consegnato un file `progetto.pro` che permetta la generazione automatica tramite `qmake` del `Makefile`.

Cosa non consegnare: codice oggetto, eseguibile, file di back-up generati automaticamente da editor o IDE e tutto quanto non necessario per la corretta compilazione ed esecuzione del programma.

7.4 Come consegnare

Dalle macchine del laboratorio invocando il comando

```
consegna progetto-pao-2014
```

dalla directory contenente **tutti e soli** i file da consegnare. **Non saranno accettate altre modalità di consegna** (ad esempio via email). Naturalmente è possibile consegnare remotamente il progetto tramite il server

```
ssh.studenti.math.unipd.it
```

e opportuni comandi/programmi come `ssh`, `sftp`, `scp`, etc.

7.5 Scadenze di consegna

Il progetto dovrà essere consegnato rispettando **tassativamente** le scadenze **ufficiali** (data e ora) previste che verranno rese note tramite le liste Uniweb di iscrizione agli esami scritti ed orali e tramite il gruppo Facebook del corso `https://www.facebook.com/groups/programmazione.oggetti`. Approssimativamente la scadenza sarà circa 7-10 giorni prima dell'esame orale.

Per i progetti ritenuti insufficienti, lo studente dovrà consegnare una nuova versione del progetto per un successivo appello orale.

Prima sessione regolare di esami orali: Come già annunciato, la prima sessione di esami prevederà un ulteriore esame orale di recupero per compensare l'inizio ritardato del corso. Le date degli esami orali della sessione regolare con relative scadenze tassative di consegna del progetto sono le seguenti:

> **Primo orale:** mercoledì 26 marzo 2014 ore 14:00 aula 1BC50, scadenza di consegna: giovedì 20 marzo 2014 ore 23:59
>
> **Secondo orale:** giovedì 3 aprile 2014 ore 13:30 aula LuF1, scadenza di consegna: giovedì 27 marzo 2014 ore 23:59
>
> **Orale di recupero:** giovedì 17 aprile 2014 ore 12:30 aula 1BC50, scadenza di consegna: giovedì 10 aprile 2014 ore 23:59