

LIBERTY : Progetto di Programmazione ad Oggetti, a.a. 2019/2020

prof. Francesco Ranzato

1 Scopo

Lo scopo del progetto è lo sviluppo in C++/Qt di una applicazione a libera scelta che soddisfi alcuni vincoli generali obbligatori. Il progetto dovrà essere sviluppato da un **gruppo di 2 o 3 studenti** e dovrà richiedere approssimativamente 50 ore di lavoro complessivo individuale. Per motivati casi eccezionali, il progetto **potrà essere sviluppato da un singolo studente**. Ogni progetto dovrà essere dotato di un nome, qui genericamente indicato con Liberty.

2 Gruppi progettuali

I gruppi progettuali di 2 o 3 studenti andranno preventivamente formati e dichiarati, **entro 2 settimane** dalla pubblicazione di questo documento. Analogamente, i singoli studenti che vorranno sviluppare individualmente il progetto dovranno fornire preventivamente la motivazione (1 linea) ed il nominativo. A tal fine, sarà **obbligatorio** inserire i nominativi dei gruppi progettuali e degli studenti individuali nel seguente documento Google Form: <https://bit.ly/3b0OJZI>

3 Vincoli Obbligatori

I **vincoli obbligatori** per il progetto Liberty sono i seguenti:

1. Definizione ed utilizzo di una gerarchia G di tipi che **usi multiple inheritance**, non necessariamente a diamante.
2. Definizione di un opportuno contenitore C , con relativi iteratori, che permetta inserimenti, rimozioni, modifiche.
3. Utilizzo del contenitore C per memorizzare oggetti polimorfi della gerarchia G .
4. Il front-end dell'applicazione deve essere una GUI sviluppata nel framework Qt.
5. Utilizzo di exception handling, dove appropriato.
6. **Obbligatorio per gruppi progettuali di 3 studenti:** Definizione di funzionalità di **input/output su file** per opportuni dati usati da Liberty: non è necessario usare la libreria standard di I/O, è anche possibile valutare l'opportunità di usare le classi per l'I/O fornite da Qt.

3.1 Gerarchia di tipi

La gerarchia G dovrà quindi includere almeno tre tipi istanziabili ed usare multiple inheritance. Usare la fantasia ed ispirarsi ai propri interessi nello scegliere la realtà da modellare mediante questa gerarchia di tipi. Naturalmente sono permesse anche gerarchie più complesse che contengono più di tre tipi.

3.2 GUI in Qt

La GUI può liberamente trarre ispirazione sia da applicazioni per sistemi desktop che app per sistemi mobile. Si potrà aderire al design pattern **Model-View-Controller** o **Model-View** per la progettazione architetturale della GUI. Qt include un insieme di classi di “view” che usano una architettura “model/view” per gestire la relazione tra i dati logici della GUI ed il modo in cui essi sono presentati all'utente della GUI (si veda <http://qt-project.org/doc/qt-5/model-view-programming.html>). 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.

3.3 Altri Requisiti Obbligatori

Il progetto LIBERTY deve soddisfare i seguenti requisiti obbligatori:

1. **Separazione tra modello logico e GUI:** in particolare, il modello logico della gerarchia G non ha alcuna relazione con il codice della GUI. La gerarchia G dovrebbe poter essere usata anche da una GUI scritta in un framework diverso da Qt.
2. Deve essere utilizzato del **codice polimorfo** che sfrutti l'organizzazione gerarchica dei tipi di G .

3.4 Tutorato

La libreria Qt offre una moltitudine di classi e metodi per lo sviluppo di GUI dettagliate e user-friendly. Il tutor Benedetto Cosentino (studente del I anno della Laurea Magistrale) renderà disponibili dei video-tutorato di laboratorio dedicati all'apprendimento delle caratteristiche di base del framework Qt per la progettazione di GUI. Verranno fatti degli annunci su Moodle quando saranno disponibili questi video-tutorato. Ulteriori informazioni al gruppo Facebook **TutorJuniorInformaticaUniPD**.

4 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:
 - (a) compilare ed eseguire correttamente nella macchina `ssh.studenti.math.unipd.it`, di cui è stata fornita una immagine da usare come macchina virtuale in VirtualBox. **NB: si tratta di una condizione necessaria per la valutazione del progetto.** La compilazione dovrà essere possibile invocando la sequenza di comandi: `qmake -project ⇒ qmake ⇒ make`. Se la compilazione del progetto necessita di particolari flag di invocazione per `qmake` (ad esempio `QT += widgets`) oppure 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, e ciò dovrà essere **esplicitamente** documentato nella relazione.
 - (b) soddisfare pienamente e correttamente tutti i requisiti obbligatori
2. **Orientazione agli oggetti:** il progetto deve **obbligatoriamente** includere delle chiamate polimorfe. Saranno oggetto di valutazione le seguenti qualità del codice prodotto:
 - (a) incapsulamento
 - (b) modularità (in particolare, massima separazione tra parte logica e grafica (GUI) del codice)
 - (c) estensibilità ed evolvibilità, in particolare mediante polimorfismo
 - (d) efficienza e robustezza
3. **Funzionalità:** quante e quali funzionalità il progetto rende disponibili, e la loro qualità.
4. **GUI:** utilizzo corretto della libreria Qt; qualità, usabilità e robustezza della GUI.
5. **Relazione:** chiarezza e qualità della relazione, che verterà sui seguenti aspetti:
 - (a) descrizione **obbligatoria** di tutte le gerarchie di tipi usate
 - (b) descrizione **obbligatoria** dell'uso di chiamate polimorfe
 - (c) descrizione **obbligatoria** di eventuali formati dei file usati per l'input/output
 - (d) se l'applicazione lo richiede, manuale utente della GUI
 - (e) se necessario, indicazione di eventuali istruzioni di compilazione ed esecuzione
 - (f) indicazione delle **ore effettivamente richieste** dalle diverse fasi progettuali: analisi preliminare del problema, progettazione modello e GUI, apprendimento libreria Qt, codifica modello e GUI, debugging, testing. In caso di superamento del previsto monte di 50 ore di lavoro complessivo per persona, giustificazione per le ore in eccesso
 - (g) **Progetti di Gruppo:** dovranno essere prodotte **due o tre diverse e distinte relazioni individuali** per ogni progetto sviluppato da un gruppo di 2 o 3 studenti. Ogni studente del gruppo dovrà quindi scrivere la propria relazione sull'**intero progetto**. Non saranno accettate relazioni identiche per un gruppo di studenti. Inoltre, ogni relazione dovrà contenere una esplicita sezione "Suddivisione del lavoro progettuale" che descriva almeno approssimativamente le diverse responsabilità progettuali dei 2 o 3 studenti del gruppo. Naturalmente dovrà essere riportato il gruppo di studenti responsabile del progetto consegnato: nome, cognome, numero di matricola.
 - (h) **In caso di progetto riconsegnato:** la relazione dovrà **obbligatoriamente** descrivere in modo dettagliato le modifiche apportate al codice rispetto alla precedente versione consegnata, in particolare elencando in modo puntuale le modifiche al codice che risolvono i punti deboli riscontrati nella precedente valutazione del progetto.

Quindi, il progetto dovrà essere **obbligatoriamente** accompagnato da una relazione scritta di **massimo 8 pagine in formato 10pt**. 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.

5 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, che verrà sempre comunicata in Moodle.
3. **ogni studente del gruppo dovrà fare una consegna distinta del progetto su Moodle:** quindi il codice sarà in comune al gruppo, mentre ogni consegna sarà caratterizzata dalla propria relazione individuale
4. essersi iscritti alla lista Uniweb dell'esame orale

Il giorno dell'esame orale, nel luogo ed all'orario stabiliti, **eventualmente in modalità telematica**, verrà comunicato l'esito della valutazione del progetto assieme ad un sintetico **feedback** sui punti deboli riscontrati nella valutazione del progetto. Il feedback per progetti svolti in gruppo naturalmente sarà lo stesso, eventualmente diverso solo per la relazione individuale. 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 rifiutare il voto oppure richiedere l'esame orale (eventualmente in modalità telematica), che potrà portare a variazioni in positivo o negativo del voto proposto.
- (B) Valutazione del progetto da completarsi con un **esame orale obbligatorio** (eventualmente in modalità telematica). 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: in questo caso **il voto dell'esame scritto rimane valido**.

Lo studente che decida di rifiutare il voto finale proposto, con o senza orale, **dovrà riconsegnare il progetto** per un successivo orale (tranne all'ultimo esame orale), cercando quindi di porre rimedio ai punti deboli segnalati nel feedback di valutazione e **descrivendo obbligatoriamente** le modifiche apportate al progetto nella relazione aggiornata. **Il voto sufficiente dell'esame scritto rimane comunque valido**. All'eventuale esame orale lo studente dovrà saper motivare **ogni** scelta progettuale e dovrà dimostrare una **generale conoscenza** del progetto ed una **piena conoscenza** delle parti di propria responsabilità nel progetto svolto in gruppo.

6 Regole

6.1 Compilatore e libreria Qt

Il progetto deve compilare ed eseguire correttamente sulla macchina **Linux** `ssh.studenti.math.unipd.it` (di cui è stata fornita una immagine da utilizzarsi come macchina virtuale per VirtualBox) con il compilatore GNU `g++ 5.x` (correntemente 5.4.0), la libreria Qt in versione 5.x (correntemente 5.9.5 su `ssh.studenti.math.unipd.it` e sulla macchina VirtualBox). È naturalmente possibile sviluppare il progetto su altri sistemi operativi come MacOS/Windows. In tal caso, prima di consegnare il progetto, ricordarsi di effettuare una prova di compilazione, esecuzione e funzionamento sulla macchina Linux `ssh.studenti.math.unipd.it`.

6.2 Cosa consegnare

Tutti i file sorgente `.h` e `.cpp` per il codice C++ assieme al file `relazione.pdf` contenente la relazione **individuale**, e ad eventuali file che memorizzano dati necessari per il corretto funzionamento del programma (ad esempio, dei file di input/output necessari al programma). Se la compilazione del progetto necessita di un project file (`.pro`) per `qmake` diverso da quello ottenibile tramite l'invocazione di `qmake -project` allora ciò dovrà essere dichiarato esplicitamente nella relazione e dovrà anche essere consegnato un file `progetto.pro` che permetta la generazione automatica tramite `qmake` del `Makefile`. **Ogni studente del gruppo** dovrà fare una consegna distinta del progetto: quindi il codice sarà in comune al gruppo, mentre ogni consegna sarà caratterizzata dalla propria relazione individuale.

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.x

6.3 Come consegnare

All'interno del Moodle del corso, saranno attivati 4 “compiti” con titolo **Consegna del progetto orale N**, con **N=1,2,3,4**, relativi alla consegna del progetto per i 4 esami orali previsti dal corso. Si dovrà consegnare un **archivio .zip** contenente **tutti e soli** i file da consegnare. **Attenzione:** La dimensione massima complessiva di tutti i file che verranno consegnati è 256MB (se la dimensione è maggiore il comando di consegna non funzionerà correttamente). **Non saranno accettate altre modalità di consegna** (ad esempio via email).

6.4 Scadenze di consegna

Il progetto dovrà essere consegnato rispettando **tassativamente** le scadenze **ufficiali** (data e ora) previste che verranno rese note tramite il Moodle del corso. Approssimativamente la scadenza sarà circa 12-14 giorni prima dell'esame orale.

Per i progetti ritenuti insufficienti, gli studenti del gruppo dovranno consegnare una nuova versione del progetto per un successivo appello orale.

Sessione estiva di esami orali: Le date degli esami orali della sessione regolare con relative scadenze tassative di consegna del progetto sono le seguenti:

Primo orale: lunedì 6 luglio 2020, scadenza di consegna: domenica 21 giugno 2020 ore 23:59

Secondo orale: giovedì 16 luglio 2020, scadenza di consegna: venerdì 3 luglio 2020 ore 23:59