

Freedom

Progetto di Programmazione ad Oggetti, a.a. 2015/2016

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 e che richieda approssimativamente 50-60 ore di lavoro complessivo. Ogni progetto dovrà essere dotato di un nome, qui genericamente indicato con Freedom.

2 Vincoli Obbligatori

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

1. Definizione ed utilizzo di una gerarchia G di tipi di altezza ≥ 1 e larghezza ≥ 1 .
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.

2.1 Interfaccia Grafica

Si valuti l'opportunità di aderire al design pattern Model-View-Controller per la progettazione architeturale 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. La libreria Qt offre una moltitudine di classi e metodi per lo sviluppo di GUI curate, dettagliate e user-friendly.

3 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 Freedom prenderà in considerazione i seguenti criteri:

1. **Correttezza** (peso 10%): il progetto deve:
 - (a) compilare ed eseguire correttamente (**NB: condizione necessaria** per la valutazione del progetto)
 - (b) soddisfare pienamente i vincoli obbligatori
 - (c) raggiungere correttamente gli scopi che si prefigge
2. **Orientazione agli oggetti** (peso 30%): qualità desiderabili 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** (peso 25%): quante e quali funzionalità il progetto rende disponibili, e la loro qualità.
4. **GUI** (peso 20%): utilizzo corretto della libreria Qt; qualità, usabilità e robustezza della GUI.
5. **Relazione** (peso 15%): chiarezza e qualità della relazione sui seguenti aspetti:

- (a) scopo del progetto
- (b) descrizione della gerarchia G : utilità e ruolo dei tipi di G
- (c) descrizione dell'uso di codice polimorfo
- (d) manuale utente della GUI, se l'applicazione lo richiede

Il progetto dovrà quindi 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.

4 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 Freedom entro la scadenza stabilita, che verrà sempre comunicata nel gruppo Facebook del corso
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 (non vi saranno altre modalità di comunicazione della valutazione del progetto) assieme ad un sintetico **feedback** sui punti deboli riscontrati nella 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 rifiutare il voto oppure 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).

Lo studente che decida di rifiutare il voto finale proposto, con o senza orale, dovrà riconsegnare il progetto per un successivo orale (tranne al quinto orale), cercando quindi di porre rimedio ai punti deboli segnalati nel feedback di valutazione. Il voto sufficiente dell'esame scritto rimane comunque 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.

5 Regole

5.1 Singleness

Il progetto dovrà essere realizzato da ogni singolo studente in modo **indipendente** da terze persone.

5.2 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 e la libreria Qt in versione 5.x. Nelle macchine Linux del laboratorio è installata la libreria Qt nella versione 5.3.2 (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.

5.3 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.

5.4 Come consegnare

Dalle macchine del laboratorio invocando il comando

```
consegna progetto-pao-2016
```

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.

5.5 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/pao15.16>. Approssimativamente la scadenza sarà circa 8-10 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: Le date degli esami orali della sessione regolare con relative scadenze tassative di consegna del progetto sono le seguenti:

Primo orale: martedì 16 febbraio 2016, scadenza di consegna: domenica 7 febbraio 2016 ore 23:59

Secondo orale: venerdì 26 febbraio 2016, scadenza di consegna: mercoledì 17 febbraio 2016 ore 23:59