

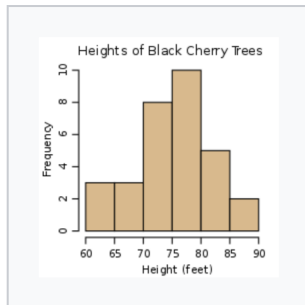
Progetto di Programmazione ad Oggetti, a.a. 2021/2022

prof. Francesco Ranzato

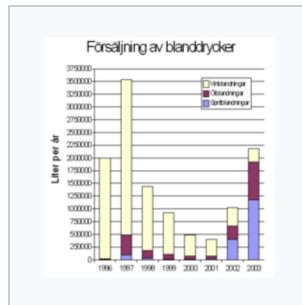
1 Scopo

Lo scopo del progetto è lo sviluppo in C++/Qt di un sistema per creare, archiviare, modificare e visualizzare **charts**, ovvero diagrammi di dati. Il progetto potrà essere sviluppato da **un singolo studente oppure da una coppia di studenti** e dovrà richiedere approssimativamente 50 ore di lavoro complessivo individuale.

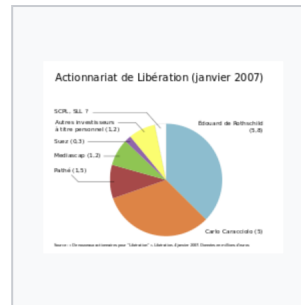
Un chart è una rappresentazione grafica di una collezione di dati con componenti tipicamente, ma non necessariamente esclusivamente, numeriche, in cui i dati sono rappresentati mediante simboli: ad esempio, come rettangoli in un diagramma a barre (detto anche istogramma), linee in un diagramma a linee, settori di un cerchio in un diagramma a torta. Si rimanda alla voce Chart di Wikipedia/English per una descrizione dettagliata dei charts e per un ampio campionario delle tipologie più comuni di charts.



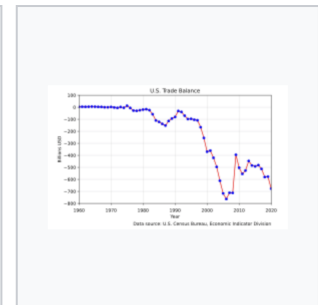
Histogram



Bar chart



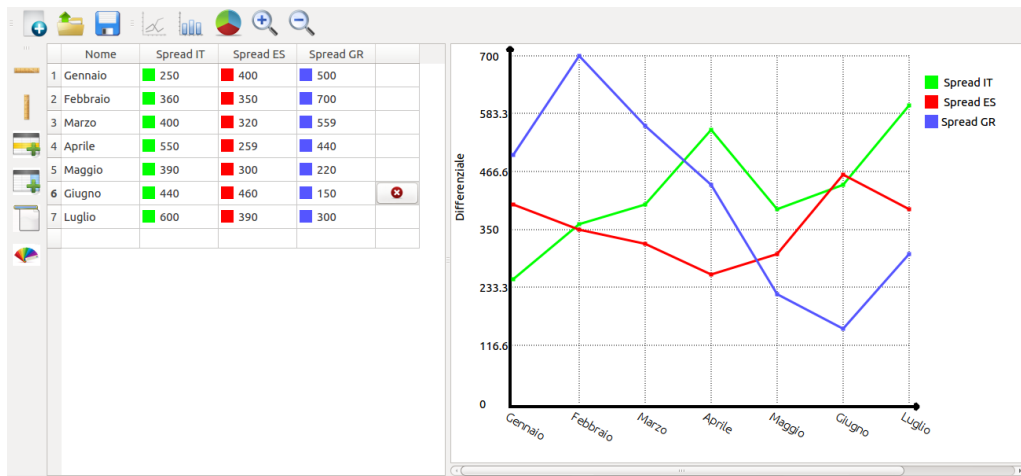
Pie chart



Line chart

Significative collezioni di dati (dataset) si possono trovare nel web, ad esempio i Chart-Infographics della conferenza ICPR.

La specifica **dettagliata** di quali funzionalità il progetto debba fornire è lasciata in massima parte a libera scelta dello studente, che può anche trarre ispirazione dal web, dagli innumerevoli charting software esistenti, dai propri interessi e dalla fantasia. Un esempio di GUI potrebbe essere come nella seguente immagine:



2 Vincoli Generali Obbligatori

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

1. Funzionalità di creazione e modifica dei dati di un chart.
2. Visualizzazione grafica di un chart con possibilità di scegliere tra almeno **tre tipologie** diverse di chart.
3. L'applicazione deve essere fruibile mediante una GUI sviluppata nel framework Qt.

4. Possibilità di archiviazione su file e recupero da file dei dati di un chart. Non è richiesto l'uso obbligatorio della libreria standard di I/O, è anche possibile valutare l'opportunità di usare le classi per l'I/O fornite da Qt. Per il formato dei file di I/O si possono anche considerare formati strutturati come XML, JSON (JavaScript Object Notation) o altri formati per lo scambio di dati.
5. Separazione tra modello logico e GUI.
6. Deve essere esplicitamente utilizzato del **codice polimorfo** che sfrutti in modo corretto, ragionevole ed estensibile una gerarchia di tipi.
7. **Obbligatorio per i progetti di coppia:**
 - (a) almeno **cinque** (invece che tre) tipologie di chart;
 - (b) utilizzo di un formato strutturato per l'I/O dei file, ad esempio JSON o XML.

2.1 GUI e Design Patterns

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.

L'applicazione potrà anche applicare design pattern comunemente utilizzati in C++, ad esempio alcuni pattern utili potrebbero essere (con link alle corrispondenti descrizioni wikipedia):

- Visitor
- Proxy
- Bridge
- Decorator

3 Video Tutorato su Qt

La libreria Qt offre una moltitudine di classi e metodi per lo sviluppo di GUI dettagliate e user-friendly. I video-tutorati dell'a.a. 2020/21 del tutor Benedetto Cosentino dedicati all'apprendimento delle caratteristiche di base del framework Qt per la progettazione di GUI sono disponibili su YouTube.

4 Iscrizione al Progetto

I singoli e le coppie andranno preventivamente dichiarati **entro il 23 gennaio 2022**. A tal fine, sarà **obbligatorio** inserire i nominativi dei singoli studenti o delle coppie di studenti nel seguente documento Google Form: <https://forms.gle/CaU11NHro87ya6NfA>

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:
 - (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 vincoli e 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 coppia:** dovranno essere prodotte **due distinte relazioni individuali** per ogni progetto sviluppato da una coppia di studenti. Ogni studente della coppia dovrà quindi scrivere la propria relazione sull'**intero progetto**. Non saranno accettate relazioni identiche per una coppia di studenti. Inoltre, ogni relazione dovrà contenere una esplicita sezione "Suddivisione del lavoro progettuale" che descriva almeno approssimativamente le diverse responsabilità progettuali della coppia di studenti. Naturalmente la relazione dovrà indicare la coppia 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.

6 Valutazione Progetto e Registrazione Voto

La registrazione del voto finale 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 per ogni appello d'esame, che verrà sempre comunicata in Moodle.
3. **ogni studente di un progetto di coppia dovrà fare una consegna distinta del progetto su Moodle:** quindi il codice sarà in comune alla coppia, mentre ogni consegna sarà caratterizzata dalla propria relazione individuale
4. essersi iscritti alla lista Uniweb della sessione d'esame per la registrazione del voto

Il giorno della registrazione del voto, verrà inviato all'email istituzionale degli studenti iscritti alla lista uniweb un sintetico **feedback** sui punti deboli riscontrati nella valutazione del progetto e verrà comunicato via la lista uniweb il voto finale complessivo dell'esame. In caso di valutazione negativa del progetto, l'esame non sarà superato: sarà quindi necessaria la **riconsegna del progetto** per una successiva scadenza di consegna; in questo caso **il voto dell'esame scritto rimane comunque valido**. Lo studente che rifiuterà il voto finale proposto via uniweb, **dovrà riconsegnare il progetto** per una successiva scadenza di consegna (tranne all'ultimo appello d'esame), cercando quindi di porre rimedio ai punti deboli segnalati nel feedback di valutazione e **descrivendo obbligatoriamente** le modifiche apportate al progetto nella relazione aggiornata; anche in questo caso **il voto sufficiente dell'esame scritto rimane comunque valido**.

7 Regole

7.1 Compilatore, libreria Qt, macchina virtuale

Il progetto deve compilare ed eseguire correttamente sulla **macchina virtuale Linux** per VirtualBox di cui è stata fornita una immagine. La macchina virtuale è dotata di compilatore GNU `g++` 7.3 e di libreria Qt nella versione 5 (5.9.5). È naturalmente possibile sviluppare il progetto su altri sistemi operativi come MacOS/Windows o su altre versioni di Linux/`g++`/Qt.

WARNING: Prima di consegnare il progetto, è **fortemente consigliata** una verifica finale di compilazione/esecuzione/funzionamento del progetto sulla macchina virtuale Linux. Si suggerisce fortemente di fare la verifica con un sufficiente anticipo in modo da avere un congruo tempo per eventuali modifiche correttive.

7.2 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 del comando `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`. Nei progetti di coppia, **ogni studente della coppia** dovrà fare una consegna distinta del progetto: quindi il codice sarà in comune alla coppia, 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.

7.3 Come consegnare

All'interno del Moodle del corso, saranno attivati 5 sessioni di **consegna del progetto per l'appello d'esame N**, con $N=1,2,3,4,5$ relativi alla consegna del progetto per i 5 esami previsti dal corso. Si dovrà consegnare **un unico 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).

7.4 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 due settimane prima della data di registrazione dell'esame (dipende dal numero di consegne previste per una specifica sessione di consegna).

Per i progetti ritenuti insufficienti, gli studenti della coppia o lo studente singolo dovranno consegnare una nuova versione del progetto per un successivo appello d'esame.

Prima sessione di esami: Le date di registrazione d'esame della sessione regolare con relative scadenze tassative di consegna del progetto sono le seguenti:

Prima registrazione: sabato 12 febbraio 2022, scadenza di consegna: domenica 30 gennaio 2022 ore 23:59

Seconda registrazione: sabato 26 febbraio 2022, scadenza di consegna: domenica 13 febbraio 2022 ore 23:59