

# Specifica Progetto di Laboratorio

Programmazione ad Oggetti

a.a. 2023/24

## 1 Introduzione

L'obiettivo del progetto è sviluppare una simulazione di sensori, ad esempio sensori che rilevano temperatura, umidità, presenza di polveri sottili o altri fenomeni, utilizzando il linguaggio di programmazione C++ e il framework [Qt](#) per creare un'interfaccia grafica utente. In questa simulazione, gli utenti finali dell'applicazione devono essere in grado di definire nuovi sensori, modificare e cancellare quelli esistenti, oltre ad avviare una simulazione. Si incoraggia l'uso di design pattern appropriati, benché non sia obbligatorio. 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.

La GUI può trarre liberamente ispirazione sia da applicazioni per sistemi desktop che app per sistemi mobile. Si potrà aderire ai *design pattern Model-View-Controller* o *Model-View* per la progettazione architetturale della GUI. Qt include un insieme di classi *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. Il framework Qt è dotato 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++.

Il voto dell'esame scritto e la valutazione del progetto sono tra loro indipendenti ed entrambi concorrono a formare il voto finale dell'esame, il quale viene calcolato indicativamente come media pesata tra il voto della prova scritta, la quale ha un peso maggiore, e il corrispettivo numerico della valutazione del progetto. Vi è comunque un margine di flessibilità favorevole allo studente nella determinazione del voto finale, in cui rimarrà preponderante il voto dell'esame scritto. Sarà possibile riconsegnare il progetto mantenendo il voto dell'esame scritto e viceversa. Sarà possibile sostenere l'esame scritto senza aver consegnato il progetto e viceversa, tuttavia la valutazione del progetto avverrà solamente per studenti che abbiano superato l'esame scritto e si siano iscritti nella lista Uniweb per la registrazione del voto finale.

## 2 Vincoli

Il progetto deve obbligatoriamente soddisfare i seguenti vincoli:

1. essere un **lavoro originale** dello studente o della coppia di studenti
2. essere interamente **scritto in C++**

3. prevedere un'interfaccia grafica realizzata in Qt
4. **compilare senza errori** sulla macchina virtuale fornita (sono tollerati, sebbene generalmente penalizzati, i *warning* durante la compilazione)
5. realizzare i principi di **incapsulamento e *information hiding*** della buona programmazione ad oggetti: una classe deve astrarre un singolo concetto e includere tutti gli attributi e metodi di cui ha bisogno con opportuni livelli di visibilità
6. mantenere una **separazione netta** tra il modello logico e l'interfaccia grafica, ovvero il codice del modello deve poter essere riutilizzabile senza dipendere dall'interfaccia; nulla vieta che il codice del modello utilizzi strumenti di Qt non legati all'interfaccia, come le funzioni di I/O o le classi per la gestione di JSON
7. eseguire in maniera **efficiente e robusta**, senza errori a *runtime*
8. utilizzare il **polimorfismo in maniera non banale**; alcuni esempi di utilizzo *banale* sono i distruttori virtuali, metodi *getter* che restituiscono informazioni leggermente diverse a seconda dell'oggetto di invocazione, metodi *getType* che restituiscono una stringa contenente il tipo dell'oggetto; per contro un utilizzo *non banale* del polimorfismo si può ottenere con metodi che svolgono operazioni diverse a seconda del tipo dinamico della variabile polimorfa di invocazione, come costruire un *widget* grafico diverso a seconda dell'oggetto da mostrare; molti *design pattern* richiedono un utilizzo del polimorfismo non banale e possono pertanto fornire ottimi spunti
9. implementare una **gerarchia di classi** per i sensori con almeno **tre classi concrete** per i progetti svolti singolarmente, o almeno **cinque classi concrete** per i progetti svolti in coppia
10. consentire la **creazione, la ricerca, la modifica e la cancellazione** dei sensori, tali operazioni devono essere fruibili tramite interfaccia grafica
11. realizzare la **persistenza dei dati** dei sensori in una qualsiasi forma per i progetti svolti singolarmente, o in un formato strutturato (JSON, XML, etc.) per i progetti svolti in coppia
12. consentire, tramite l'interfaccia grafica, il **salvataggio e il caricamento di un file di sensori** (come descritto al punto precedente) attraverso opportune finestre di dialogo: non è consentito l'utilizzo di percorsi di file di sensori cablati nel codice sorgente
13. consentire, tramite interfaccia grafica, di **avviare una simulazione** per un sensore selezionato, simulando un certo numero di raccolte di dati; i dati dovranno essere quindi mostrati in maniera opportuna, ad esempio tramite QtCharts; non è necessario che la simulazione sia interattiva
14. essere corredato di una **relazione** in formato pdf, in lingua italiana o inglese, di massimo 8 pagine con testo a 10pt che riporti:

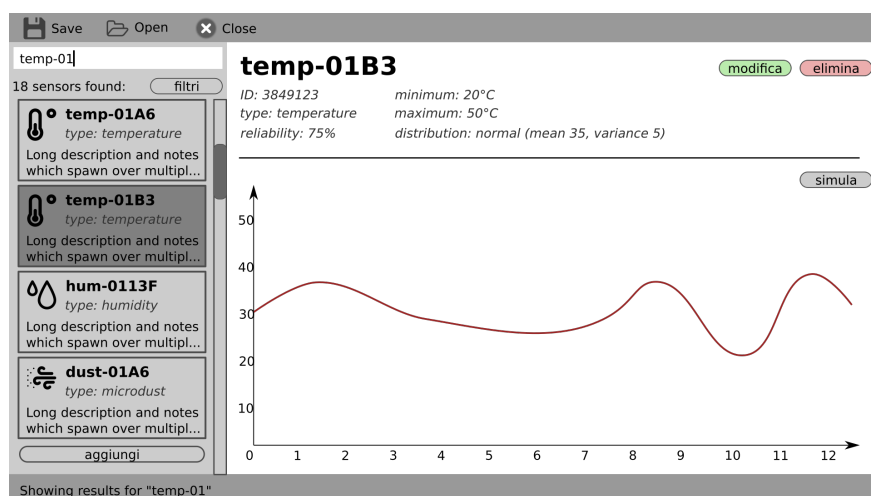


Figura 1: Esempio di scheletro dell'interfaccia della finestra principale

- (a) nome, cognome e numero di matricola di tutti i componenti del gruppo (o del singolo autore in caso di progetto svolto singolarmente)
- (b) una breve introduzione che descriva i sensori utilizzati
- (c) la descrizione del modello logico
- (d) la descrizione dell'utilizzo non banale del polimorfismo
- (e) la descrizione del metodo utilizzato per la persistenza dei dati
- (f) la descrizione delle funzionalità implementate
- (g) la rendicontazione delle ore previste e di quelle effettivamente svolte
- (h) solo per progetti svolti in coppia: la suddivisione delle attività progettuali
- (i) solo per progetti riconsegnati: una sezione che riporta le modifiche apportate rispetto all'ultima consegna; in caso di riconsegna senza modifiche è preferibile aggiungere comunque questa sezione indicando che non ci sono state modifiche

In caso di progetto di coppia, **le relazioni devono essere distinte**. Assieme al documento di specifiche viene fornito anche un modello commentato di relazione il cui utilizzo non è obbligatorio. Si suggerisce inoltre l'uso di [LaTeX](#) per la redazione della relazione, il sistema di scrittura universalmente usato dagli informatici in ambito accademico.

La Fig. 1 mostra un esempio di struttura dell'interfaccia grafica.

### 3 Consegna

La consegna avverrà tramite il Moodle del corso, all'interno del quale saranno attivate cinque sessioni di consegna del progetto per gli altrettanti appelli d'esame previsti durante l'anno accademico (due sessioni di consegna a gennaio/febbraio 2024, due sessioni di consegna a giugno/luglio 2024, una sessione di

consegna a settembre 2024). Si dovrà consegnare un unico archivio in formato zip della dimensione massima di 256MB attraverso l'apposita pagina Moodle. **Non saranno accettati tentativi di consegna con modalità diverse o formati che non siano zip.** La cartella compressa dovrà contenere:

- la relazione in formato PDF
- una sottocartella con i file sorgente del progetto (.h, .cpp, .pro, eventuali sottocartelle) ed eventuali file multimediali necessari (immagini, icone, ecc.)
- almeno un file d'esempio per la persistenza dei dati

La cartella *non dovrebbe* contenere codice oggetto compilato, eseguibili, file generati dall'IDE o in generale qualsiasi file non utile ai fini della valutazione. Si raccomanda di verificare il corretto caricamento dei file su Moodle e di comunicare tempestivamente eventuali malfunzionamenti.

Nei progetti di coppia ogni studente dovrà provvedere separatamente alla propria consegna includendo il codice (lo stesso per i due membri della coppia) e la propria relazione (diversa per ciascuno). Se nella coppia solamente uno studente è sufficiente all'esame scritto, questo studente può comunque procedere alla consegna del progetto con l'obiettivo di finalizzare l'esame. Si ricorda che la valutazione del progetto con relativo feedback avverrà solamente per gli studenti sufficienti all'esame scritto ed iscritti alla lista uniweb per il voto finale.

## 4 Criteri di valutazione

Il progetto viene valutato sulla base dei vincoli obbligatori e delle funzionalità implementate. Più precisamente, **se uno o più vincoli obbligatori non risultano soddisfatti il progetto verrà considerato insufficiente** e sarà necessaria una riconsegna in uno degli appelli successivi. Viceversa, **se tutti i vincoli obbligatori sono soddisfatti il progetto è considerato (almeno) sufficiente** e la valutazione aumenterà in base alla qualità delle funzionalità sviluppate e, in misura minore, in base alla qualità della relazione.

Una funzionalità viene valutata positivamente in base alla sua pertinenza al tema, all'utilità, all'usabilità, alla complessità e alla qualità del codice attraverso cui è implementata. Funzionalità più semplici o generiche migliorano la valutazione, sebbene non tanto quanto idee più complesse o articolate. Le scorciatoie da tastiera, per esempio, sono migliorie generiche semplici da ottenere con poche righe di codice in Qt, così come la gestione del ridimensionamento delle finestre o l'uso di icone. Per contro l'integrazione con un sistema di API o l'uso di basi di dati come SQL o MongoDB per la persistenza sono significativamente più complessi e richiedono la scrittura di classi apposite.

La qualità della relazione, pur avendo un'incidenza minore, verrà valutata sulla base della completezza, della chiarezza e della coesione. **Errori linguistici evidenti** come sistematica mancanza di punteggiatura o errori di battitura frequenti **verranno penalizzati**. È possibile redigere la relazione in Italiano o Inglese, a propria discrezione. La scelta della lingua non avrà effetti sulla valutazione.

Poiché il corso non tratta di usabilità o resa estetica della GUI la loro mancanza **non verrà penalizzata**, purché questo non pregiudichi il corretto funzionamento del programma. Tuttavia, se il progetto viene sviluppato ponendo particolare attenzione a queste caratteristiche, verrà riconosciuto un bonus come se si trattasse di una funzionalità aggiuntiva.

La valutazione del progetto è **valida per l'intero anno accademico** e fino a che non viene (ri)consegnato un progetto.

La valutazione è accompagnata da un feedback testuale che motiva la valutazione ed evidenzia i punti di forza e debolezza del progetto. È possibile riconsegnare un progetto per ottenere una nuova valutazione, che potrebbe anche essere peggiorativa, tuttavia se si seguono le indicazioni fornite dal feedback sarà generalmente migliorativa. La valutazione è *idempotente*: riconsegnare lo stesso progetto senza modifiche produrrà esattamente lo stesso voto.

Consegnare un progetto svolto anche parzialmente da altri (con o senza il loro consenso) oppure generato da sistemi automatici comporta automaticamente l'insufficienza.

La registrazione del voto finale è possibile solo dopo:

- avere superato con valutazione maggiore o uguale a 18 la prova scritta
- essersi iscritti alla lista Uniweb della per la registrazione del voto finale
- avere consegnato il progetto entro la scadenza prevista per la sessione in cui si intende verbalizzare il voto e aver ottenuto una valutazione positiva

Si ricorda che le liste Uniweb verificano il soddisfacimento delle propedeuticità e hanno una data di chiusura anticipata di almeno 5 giorni. Non saranno ammesse iscrizioni manuali in ritardo, per nessun motivo.

Il giorno della registrazione del voto finale previsto dalla corrispondente lista uniweb verrà inviato all'email istituzionale dei soli studenti iscritti alla lista uniweb il feedback relativo alla valutazione del progetto e verrà registrato su 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, per cui ovviamente non esiste una successiva scadenza di consegna), 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.

## 5 Note

La parte di laboratorio dell'insegnamento di Programmazione a Oggetti ha una propria pagina ufficiale su GitHub all'indirizzo [github.com/Unipd-Object-Oriented-Programming](https://github.com/Unipd-Object-Oriented-Programming). Questo spazio contiene il materiale relativo a questo modulo dell'insegnamento, inclusi i lucidi delle lezioni e gli esempi del codice, suddivisi per anno accademico.

I video-tutorati dell'anno accademico 2020/2021 del tutor Benedetto Cosen-  
tino dedicati all'apprendimento delle caratteristiche di base del framework Qt  
per la progettazione di GUI sono disponibili su [YouTube](#).