

EnviroSense

Anno Accademico 2023/2024

Oggetto: Relazione sul progetto di Programmazione ad Oggetti

di

Bolognini Federica

federica.bolognini@studenti.unipd.it

Matricola: 2011881

Rendicontazione oraria:

Attività	Ore previste	Ore effettive
Studio e progettazione	8	8
Sviluppo del codice del modello	8	10
Studio del framework Qt	18	20
Sviluppo del codice della GUI	10	8
Test e debug	3	3
Stesura della relazione	3	3
Totale	50	52

1 Introduzione

Il programma implementa un sistema avanzato di gestione per diversi tipi di sensori ambientali usati all'interno di una serra.

L'architettura è basata su una gerarchia di classi che rappresentano vari tipi di sensori, ciascuno specializzato per misurare parametri specifici dell'ambiente.

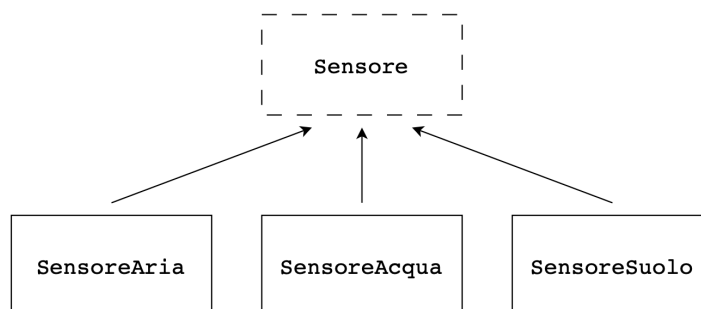
Il sistema è progettato per mantenere una collezione dinamica di sensori, consentendo operazioni di aggiunta, rimozione e modifica in modo flessibile e sicuro.

I sensori devono sottostare ad alcuni vincoli logici:

- Il nome non può essere vuoto.
- La precisione deve essere compresa tra 0 e 100%.
- Ogni misurazione deve avere un range valido definito tra un minimo ed un massimo.
Questo range operativo varia a seconda del tipo di sensore utilizzato e delle specifiche caratteristiche che il sensore è progettato per rilevare.
- L' ID deve essere univoco per ogni sensore.

2 Gerarchie dei tipi

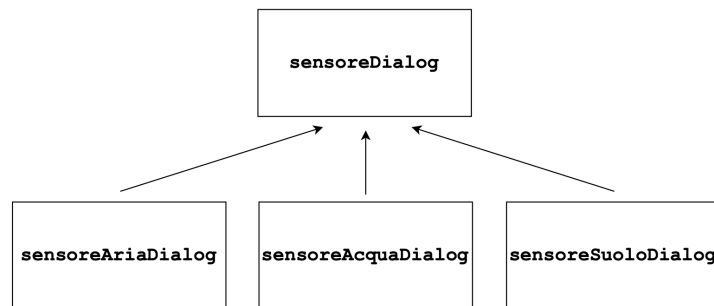
Il progetto EnviroSense prevede due gerarchie distinte: una gerarchia principale nel modello, composta dalle seguenti classi:



- La classe "Sensore" è la classe base astratta della gerarchia: essa gestisce le informazioni comuni a tutti i sensori, ovvero il nome, la precisione, l'ID, il range di temperatura valida e le temperature campionate.
- La classe "SensoreAria" è una classe concreta che eredita da "Sensore" e gestisce le informazioni specifiche per i sensori dell'aria, come i livelli di CO2 e ossigeno. Implementa i metodi virtuali puri della classe base:
 - Qualità(): Calcola un indice di qualità basato sui parametri specifici rilevati da ciascun tipo di sensore.

- Clone(): Definito anch'esso nella classe "Sensore", implementa il "clone pattern" (è un pattern per ottenere un "costruttore di copia polimorfo").
- La classe "SensoreAcqua" è una classe concreta che eredita da "Sensore" e si occupa dei dati relativi all'alcalinità e all'acidità dell'acqua. Anch'essa implementa i metodi virtuali puri della classe base.
- La classe "SensoreSuolo" è una classe concreta che eredita da "Sensore" e gestisce i parametri specifici del suolo come pH e umidità. Implementa i metodi virtuali puri come le altre classi concrete.

La seconda gerarchia appartiene alla vista:



- La classe base 'sensoreDialog' fornisce funzionalità comuni ed ha tre classi derivate specifiche per diversi tipi di sensori: 'sensoreAriaDialog', 'sensoreSuoloDialog', e 'sensoreAcquaDialog'.
- La classe 'sensoreAriaDialog' include campi per la configurazione dei livelli minimi e massimi di CO2 e ossigeno.
 - getValues(): Questo metodo crea un'istanza della finestra di dialogo e la esegue, raccoglie tutti i valori inseriti dall'utente (nome, precisione, limiti, ecc.), restituendo un oggetto dialogValues contenente tutti i dati raccolti e gestisce anche il caso di annullamento dell'operazione, lanciando un'eccezione
- La classe 'sensoreAcquaDialog' gestisce impostazioni per alcalinità e acidità. Anche in questa classe viene implementato il metodo getValues().
- La classe 'sensoreSuoloDialog' si concentra su parametri come umidità e pH. Anche in questa classe viene implementato il metodo getValues().

3 Polimorfismo

Il concetto di polimorfismo è stato implementato attraverso:

- Il metodo virtuale puro "Qualità()" nella classe base Sensore.
Questo metodo viene implementato in ciascuna classe derivata (SensoreAcqua, SensoreAria, SensoreSuolo) per calcolare la qualità specifica del sensore.
Per ogni tipo di sensore, il metodo Qualità() calcola una percentuale basata sui valori misurati e sui limiti validi impostati. Ad esempio, per SensoreAria, viene calcolata la percentuale di misurazioni valide per temperatura, CO2 e ossigeno.
Il metodo restituisce un valore intero che rappresenta la qualità complessiva del sensore, ottenuta come media delle percentuali calcolate per ciascun parametro.
Questo approccio permette di trattare uniformemente i diversi tipi di sensori attraverso l'interfaccia comune della classe base, mentre ogni classe derivata implementa la logica specifica per valutare la propria qualità.
- Il metodo virtuale puro "clone()" nella classe base Sensore.
Questo metodo viene implementato in ciascuna classe derivata (SensoreAcqua, SensoreAria, SensoreSuolo) per creare una copia esatta dell'oggetto specifico. Per ogni tipo di sensore, il metodo clone() restituisce un puntatore ad un nuovo oggetto dello stesso tipo, contenente una copia di tutti i dati dell'oggetto originale.

4 Persistenza dei dati

Il programma consente di salvare i dati in file strutturati XML, la struttura del file consta dei seguenti tag:

- <collezione>:indica l'insieme dei sensori nel suo complesso.
- <sensore tipo>:indica il tipo del sensore.
- <nome>:indica il nome del sensore.
- <precisione>:indica la precisione del sensore.
- <id>:indica l'id del sensore.
- <minValidTemperatura>:indica il valore minimo della temperatura possibile.
- <maxValidTemperatura>:indica il valore massimo della temperatura possibile.
- <temperature>:indica una raccolta di temperature rilevate.
- <minValidFirstVector>:indica il valore minimo del primo vettore del sensore concreto.

Ad esempio, per il sensore acqua il primo vettore è l'alcalinità, per il sensore aria è CO2 e per il sensore suolo è il PH.

- `<maxValidFirstVector>`:indica il valore massimo del primo vettore del sensore concreto.
- `<FirstVector>`:raccolta di misurazioni del primo vettore.
- `<minValidSecondVector>`:indica il valore minimo del secondo vettore del sensore concreto.
Ad esempio, per il sensore acqua il secondo vettore è l'acidità, per il sensore aria è ossigeno e per il sensore suolo umidità.
- `<maxValidSecondVector>`:indica il valore massimo del secondo vettore del sensore concreto.
- `<secondVector>`:raccolta di misurazioni del secondo vettore.

NOTA BENE: Durante la fase di testing sull'IO si è notato come la traduzione da "double" a "QString" usi il simbolo "." per separare le unità dalla parte decimale su ambiente Windows, a differenza dei sistemi operativi basati su kernel Linux in cui si usa il simbolo ",".

Per questo motivo si è deciso di sostituire il simbolo "," con "." su ambiente Windows, durante la lettura dati (viceversa per tutti gli altri OS).

5 Funzionalità implementate

5.1 Funzionalità C++:

- Gestione di una collezione di sensori (operazioni CRUD).
- Gestione della simulazione di campionamento dati per la collezione di sensori.
- Salvataggio dei dati in formato XML: ogni collezione di sensori può essere salvata in un file strutturato XML tramite due voci nel menù "File": "Salva" e "Salva col nome"; la voce "Salva", oltre a salvare i dati, consente di poter memorizzare il percorso del file utilizzato in modo da poter, usando lo stesso comando, salvare sullo stesso file una versione aggiornata dei dati senza aprire una seconda volta il dialog di sistema per il salvataggio.
- Ripristino dei dati da file XML: è possibile ripristinare, se il file è strutturato in maniera coerente ed i dati sono logicamente corretti, una collezione di sensori precedentemente salvata.

5.2 Funzionalità grafiche:

- Ricerca di un sensore tramite campo di ricerca filtrando la collezione di sensori tramite il nome o il tipo.
In caso di ricerca vuota vengono mostrati tutti i sensori salvati della collezione.



- Visualizzazione del sensore tramite tabella, se l'utente esegue un doppio click su una cella della riga del sensore cercato quest'ultimo diventerà il sensore attivo.
- Operazioni CRUD del sensore selezionato tramite appositi bottoni.
- Visualizzazione dei dettagli del sensore attivo, in cui vengono mostrati il nome, tipo, ID, precisione, temperatura minima, temperatura massima e qualità.
- Visualizzazione dei dati campionati del sensore attivo tramite grafico a linee.
- La simulazione del sensore attivo viene avviata tramite il click dell'apposito bottone "Simula". Ogni volta che si esegue la simulazione vengono aggiornati la tabella, i dettagli e il grafico per il sensore attivo.