

oggetto Relazione Progetto Programmazione a Oggetti
studente Oliver Florin Stiglet
mat. 2044895
titolo QtSensorHub

1222 • 2022
800
ANNI



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Relazione Progetto Programmazione ad Oggetti

QtSensorHub

Introduzione

QtSensorHub è un'applicazione Qt sviluppata in C++ con l'obiettivo di assistere i medici nel monitoraggio dei pazienti diabetici attraverso l'utilizzo di dispositivi dedicati. Ogni paziente è dotato di un dispositivo che ospita tre sensori simulati per le misurazioni di glucosio, insulina e pressione. L'interfaccia principale offre diverse funzionalità utili, tra cui l'aggiunta di nuovi pazienti, il salvataggio e il caricamento dei dati, nonché la visualizzazione e la gestione dei sensori associati a ciascun paziente.

Una caratteristica distintiva dell'applicazione è la possibilità di filtrare la lista dei pazienti in base a criteri specifici, come la visualizzazione dei sensori preoccupanti o il nome del paziente.

Il fulcro dell'applicazione è rappresentato dalla scheda paziente, che consente ai medici di visualizzare e gestire i sensori associati a un paziente specifico. Attraverso questa interfaccia, è possibile aggiungere nuovi valori, simulare dati, eliminare o ripristinare sensori e pazienti. Inoltre, è disponibile la visualizzazione dei valori attraverso un grafico che offre una chiara rappresentazione dell'andamento dello stato del paziente, facilitando la comprensione se i valori sono nella norma.

Un'altra caratteristica cruciale dell'applicazione è la possibilità di salvare e caricare lo stato attuale dei dati in un file JSON.

In sintesi, questo programma si propone di semplificare e migliorare il processo di monitoraggio dei pazienti diabetici per i professionisti medici, fornendo un'interfaccia intuitiva e strumenti utili per gestire i dati dei sensori in modo efficace.

Descrizione del Modello

Il modello dell'applicazione è suddiviso in due componenti principali: una parte logica e una parte grafica (view)

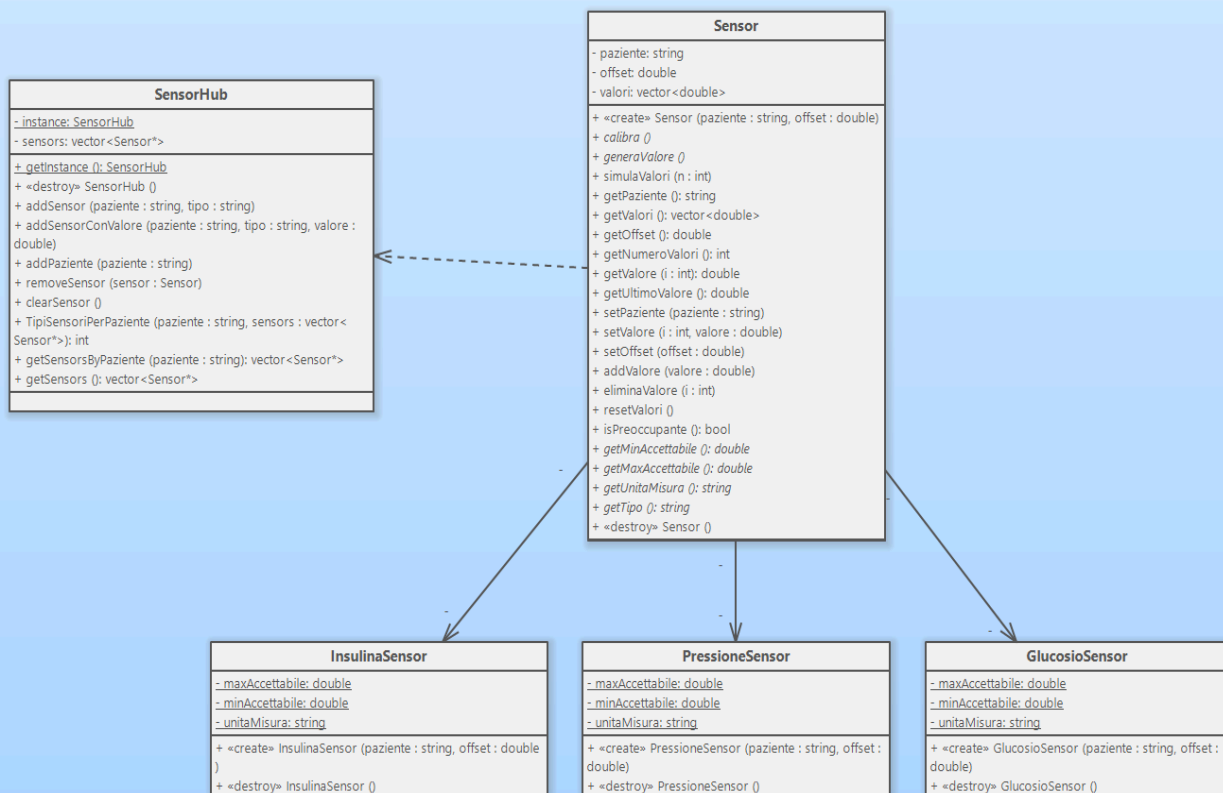
Parte Logica:

La componente logica dell'applicazione è strutturata attraverso due classi principali. La prima è la classe "Sensor," definita come astratta pura, che rappresenta genericamente un sensore con i suoi valori e l'offset associato. Da questa classe madre derivano le classi concrete, tra cui "GlucosioSensor," "InsulinaSensor," e "PressioneSensor." Ognuna di queste classi concrete implementa i metodi virtuali della classe base attraverso specifici calcoli, considerando valori massimi e minimi, per simulare il funzionamento del sensore correlato al tipo di misurazione.

La seconda classe fondamentale è la "SensorHub," che rappresenta il cuore del programma. Questa classe contiene l'insieme di sensori collegati ai singoli pazienti e gestisce le operazioni di aggiunta e modifica di questi ultimi. La "SensorHub" svolge un ruolo centrale nel coordinamento e nell'interazione tra i sensori e i dati dei pazienti.

La classe "Sensor" fornisce una base comune, consentendo la creazione di nuovi sensori con caratteristiche specifiche. Le classi concrete implementano le logiche di misurazione specifiche per ciascun tipo di sensore.

La classe "SensorHub" facilita l'organizzazione e la gestione dei sensori, fornendo funzionalità chiave per il monitoraggio dei pazienti. L'aggiunta e la modifica dei pazienti tramite questa classe centrale agevola la coesistenza e la sincronizzazione dei dati tra i vari sensori e i relativi pazienti.



Poliformismo

Il polimorfismo riveste un ruolo fondamentale nella simulazione e rappresentazione dei valori, soprattutto quando si tratta di diversi tipi di sensori, ciascuno con limiti massimi e minimi specifici che richiedono particolare attenzione. Nell'ambito di questa simulazione, è essenziale generare valori plausibili per ciascun sensore, utilizzando funzioni e offset differenti in modo da riflettere realisticamente le caratteristiche di ciascun dispositivo.

Questo concetto di polimorfismo si estende anche alla creazione di grafici e alla rappresentazione tabellare dei dati. Ogni sensore, attraverso il polimorfismo, contribuisce a definire un comportamento unico nella generazione del grafico, garantendo una visualizzazione adatta alle specifiche del sensore considerato.

Persistenza dei dati

La gestione della persistenza dei dati in questo codice avviene attraverso la scrittura e la lettura di dati da un file JSON. Ecco una breve panoramica del processo:

Salvataggio dei dati: Quando l'utente preme il pulsante "Salva", il programma richiede all'utente di specificare un file JSON in cui memorizzare i dati dei sensori. Successivamente, il programma crea un oggetto JSON contenente tutte le informazioni dell'oggetto "SensorHub," includendo i sensori attivi con i loro nomi, tipi e valori associati. Questo oggetto JSON viene quindi salvato nel file specificato.

Caricamento dei dati: Quando l'utente preme il pulsante "Carica," il programma richiede all'utente di selezionare un file JSON da cui caricare i dati delle attività. Dopo la scelta del file, il programma lo apre correttamente e visualizza i sensori e i pazienti presenti nel file.

Funzionalità implementate

Queste sono le principali funzionalità implementate nel codice:

Pagina Principale:

- **Aggiunta di Paziente e Sensori:** Fornisce un'interfaccia intuitiva per l'aggiunta di nuovi pazienti e la connessione di sensori ad essi.
- **Lista di Widget:** Mostra una lista dettagliata di pazienti, sensori collegati e il loro stato corrente, facilitando la visione globale delle informazioni.
- **Pulsanti Salva/Carica:** Implementa la possibilità di salvare e caricare i dati attraverso file JSON, consentendo la persistenza dei dati tra diverse sessioni dell'applicazione.
- **Filtri Lista Pazienti:** Permette di filtrare la lista dei pazienti in base al nome e allo stato, semplificando la ricerca e la gestione dei dati.

Scheda del Paziente:

- **Aggiunta di Valori:** Permette ai medici di inserire nuovi valori associati a un paziente specifico.
- **Eliminazione di Sensori o Paziente:** Offre la possibilità di rimuovere sensori specifici o l'intero paziente, fornendo un controllo completo sull'archiviazione dei dati.
- **Reset Sensore:** Consente di reimpostare uno specifico sensore ai valori di default.
- **Simulazione di Nuovi Valori:** Permette di generare e visualizzare un numero simulato di nuovi valori per il sensore in esame.
- **Tabella dei Valori del Sensore:** Visualizza in modo ordinato e dettagliato i valori storici associati a un sensore.
- **Grafico del Sensore:** Offre una rappresentazione visiva chiara dell'andamento dei valori del sensore, fornendo un riassunto immediato delle informazioni più rilevanti.

Rendicontazione Ore

Attività	Ore Effettuate
Studio e progettazione	3
Sviluppo del codice del modello	12
Studio del framework Qt	10
Sviluppo del codice della GUI	15
Test e debug	10
Stesura della Relazione	5
Totale	55

Avvio Applicazione

Per avviare l'applicazione una volta andati all'interno della cartella QtSensorhub bisognerà eseguire i tre seguenti comandi:

- `qmake`
- `make`
- `./CampusConnect`