

qCharts

Progetto di Programmazione ad Oggetti, a.a. 2012/2013



Sistema operativo di sviluppo: Linux Mint 14 Nadia

Versione compilatore: Qt Creator 2.5.2

Versione Qt: 4.8.2

Introduzione

Il progetto è stato sviluppato con Qt Creator e librerie Qt 4.8.2 in ambiente Linux Mint Nadia 14.

Scopo del Progetto

Lo scopo del progetto qCharts è lo sviluppo in C++/Qt di un sistema per creare, archiviare, modificare e visualizzare grafici.

Le funzionalità principali del progetto si compongono di:

- Creazione e modifica dei dati di un chart.
- Salvataggio (e apertura) su file Xml dei dati.
- Visualizzazione dei dati su un grafico a scelta tra: BarChart, LineChart e PointChart.

Viene modellato un campionato immaginario tra squadre (eventualmente di calcio), ognuna delle quali ha uno specifico punteggio. Per testare il programma viene fornito il file *Prova.xml*.

Scelte progettuali di maggior rilevanza

- Identificazione dei tipi a run-time

Effettuata esclusivamente utilizzando il costrutto *dynamic_cast<>()*.

- Gestione della memoria

Per quanto riguarda la parte grafica, la maggior parte degli oggetti creati viene distrutta dalla libreria Qt, la quale alla distruzione del widget padre distrugge automaticamente i figli. Fa eccezione la classe Table le cui righe sono deallocate mediante il distruttore ridefinito che dealloca l'oggetto RowTable corrispondente. Infine sono stati ridefiniti i distruttori di Valori e Canvas per deallocare gli oggetti allocati nello heap (Squadre e Chart).

- Overloading di operatori

Dove necessario è stato effettuato l'overloading degli operatori interessati (ad esempio gli operatori di confronto per la classe Squadra).

Parte Logica

- Classe Squadra

Questa classe rappresenta una squadra ed è caratterizzata da un campo nome e un punteggio.

- Classe Valori

Classe collezione di squadre realizzata mediante l'utilizzo del contenitore vector. Contiene i puntatori alle squadre che si trovano attualmente in memoria.

- Classe Dati

Rappresenta i dati numerici appartenenti agli oggetti contenuti in Valori, sostanzialmente è un vector di punteggi.

- Classe DatiN

Contiene i dati normalizzati rispetto al massimo di essi, in modo da poterli rappresentare graficamente in modo proporzionato.

- Classe Chart

Classe base astratta che rappresenta un chart, contiene un vector di punti in coordinate Qt nella parte protetta in modo che siano accessibili dalle sottoclassi concrete.

- Classi BarChart LineChart PointChart

Classi derivate pubblicamente da Chart che implementano il metodo virtuale puro *Draw()* di Chart per disegnare il rispettivo grafico: a barre, linee o punti.

Parte Grafica

- Classe MainWindow

La finestra principale del programma contenente menu che permettono all'utente di sfruttarne le caratteristiche, tra cui l'apertura e il salvataggio di file Xml (sfruttando le classi QDomStreamReader e QDomStreamWriter fornite da Qt).

- Classe Mdi

E' il widget principale di MainWindow. E' sottoclasse di QMdiArea e si occupa di visualizzare in due distinte schede i dati e il grafico corrispondente.

- Classe Table

Classe designata per la visualizzazione dei dati (eredita da QTableWidgetItem) che vengono presentati all'utente in forma tabellare. La tabella può essere modificata mediante la modifica dei dati stessi o l'aggiunta di nuove righe fino ad un massimo di 99.

- Classe RowTable

Sottoclasse di QObject i cui oggetti rappresentano una riga della classe Table, formati da un campo contenente il nome della squadra e da uno contenente il punteggio.

- Classe Canvas

Widget su cui vengono dipinti i dati (sottoclasse di QWidget), cioè disegnato il Chart con i dati in memoria utilizzando il metodo virtuale che produce il grafico corretto tra i 3 possibili (RTTI). Nel caso di modifica ai dati da parte dell'utente il grafico viene aggiornato.

- Classe Aiuto

Eredita da QDialog ed è una finestra di dialogo con l'utente definita allo scopo di informarlo brevemente sull'utilizzo del programma, una sorta di guida rapida.