Programmazione ad Oggetti 2017/18

KALK

Matteo Depascale 1120162

Gianmarco Pettenuzzo 1097856

Relazione di Matteo Depascale

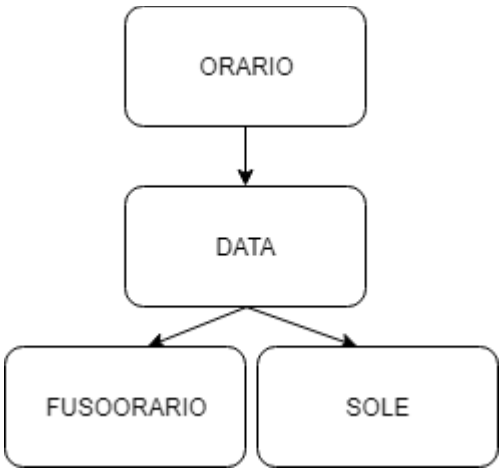
1) Introduzione

Abbiamo creato una calcolatrice in grado di fare operazioni con tipi orario, data, fuso orario e che permettesse, immettendo i dati correttamente, di calcolare tramite la classe sole, l'orario dell'alba, del tramonto e della massima elevazione del sole durante l'arco della giornata.

Per quanto riguarda la classe Sole, la scelta di impostare in input l'Equazione del Tempo e la Declinazione del Sole è stata puramente progettuale, in quanto, dopo svariate ricerche, ci siamo accorti che le 50 ore assegnate per il completamento del progetto, non sarebbero state sufficienti per studiare e progettare l'intera classe.

Viene inoltre inserito il file cpp.pro con il .pro del progetto, l'unica differenza sta nel comando qmake -project "QT+=widgets".

2) Descrizione Gerarchie



2.1 ORARIO

La classe base, composta dal campo dati privato *sec*. Sono presenti i metodi di classe come il calcolo della velocità, la divisione, il formato in 12h e i metodi virtuali per operazioni di somma e sottrazione, anche dei singoli elementi della classe, e per resettare o modificare il campo dati.

2.2 DATA

La sottoclasse di Orario, composta dai campi dati privati giorno, mese, anno. Sono presenti i metodi di classe come giorniMese, utile per i controlli, contaSettAnno e contaGiorniAnno, stagione, giornoSettimana, meseAnno, formatoData e durata che ritorna i giorni trascorsi o da trascorrere da una data. I metodi virtuali sono le ridefinizioni dei metodi virtuali di Orario e l'aggiunta delle apposite funzioni virtuali per sommare, sottrarre e modificare i relativi campi dati della classe.

2.3 FUSOORARIO

Sottoclasse di Data, composta dal campo dati privato *citta*. Sono presenti i metodi di classe Fuso, Latitudine, Longitudine, Emisfero, confrontaFusi che ritorna la differenza di fuso tra le due città e

orarioCitta che ritorna la data della città selezionata in base al proprio fuso orario. Per quanto riguarda i metodi virtuali invece vengono ridefiniti solamente quelli di somma e sottrazione, e clearMem poiché gli altri metodi vengono ereditati da Data.

2.4 SOLE

Sottoclasse di Data, composta dai campi dati privati lat, lon, tz, oraLegale, tempoEQ, soleDeclina. Sono presenti i metodi di classe giornoGiuliano, secoloGiuliano, angoloSolare e soleUTC utilizzare per calcolare le funzioni principali Tramonto, Sorgere e mezzogiornoSolare. Come metodi virtuali ridefinisce solamente il clearMem in quanto gli altri metodi vengono ereditati da Data. Bisogna specificare che la classe non fa uso di nessun altro metodo virtuale oltre a clearMem poiché il risultato delle funzioni potrebbe essere errato in quanto i campi dati tempoEQ e soleDeclina sono frutto a loro volta di calcoli e perciò, cambiando l'orario o la data sì avrà un calcolo diverso.

2.5 OVERFLOW_ERROR

Classe utilizzata per la gestione degli errori.

3) Descrizione Polimorfismo

Il polimorfismo viene principalmente utilizzato nelle operazioni di somma e sottrazione tra orari e tra un orario e una data. Abbiamo deciso di non sommare una data a un'altra data in quanto, in nostra opinione, questa somma non risulterebbe utile come sommare un orario ad una data.

Nella classe Orario viene reso virtuale il distruttore, in modo che, al momento della distruzione di un oggetto, venga invocato il distruttore della classe a cui appartiene.

E' presente un metodo virtuale clearMem che si occupa di resettare i campi dati allo stato di default.

Altri metodi virtuali sono aggiungiSecondi, aggiungiMinuti, aggiungiOre, sottraiSecondi, sottraiMinuti, sottraiOre per quanto riguarda Orario, mentre su Data abbiamo l'overriding di questi metodi e nuovi metodi virtuali come aggiungiGiorno, aggiungiMese, aggiungiAnno, togliGiorno, togliMese, togliAnno. Questi metodi verranno invocati nella classe dell'oggetto di appartenenza.

Nella classe Fusoorario non viene fatto l'overriding dei metodi sopra citati in quanto viene sfruttata l'ereditarietà della classe con Data.

Infine, nella classe Sole è possibile, per ereditarietà, usufruire dei metodi virtuali però i campi dati tempoEQ e soleDeclina dipendono dall'orario inserito in input alla creazione dell'oggetto Sole, quindi abbiamo preferito non ridefinire somma e sottrazione.

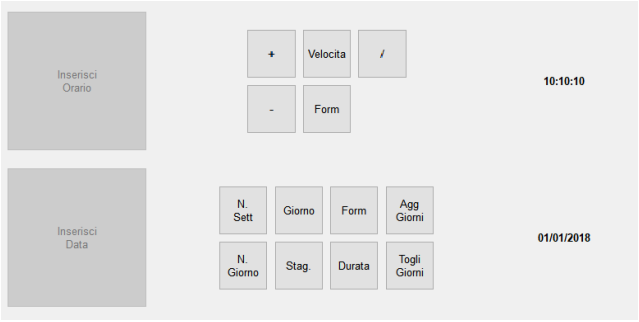
4) Manuale utente GUI

- + e – per fare operazioni di somma e sottrazione

- Velocità per calcolare la velocità media immettendo come input una distanza

- Form ritorna l'orario in formato da 12 ore

- / per dividere l'orario per un dato input



- E' possibile utilizzare le funzioni della classe Orario

- N. Sett ritorna il numero delle settimane da inizio anno fino alla data in corso

- N. Giorno ritorna il numero di giorni da inizio anno fino alla data in corso

- Giorno ritorna il nome del giorno della settimana

-Stagione ritorna il nome della stagione in corso

- Form ritorna una data scritta con nome giorno, numero, nome mese e anno

- Durata ritorna la differenza in giorni da una data chiesta in input

- Agg. Giorni e Togli Giorni, rispettivamente, aggiungono tot giorni e tolgono tot giorni

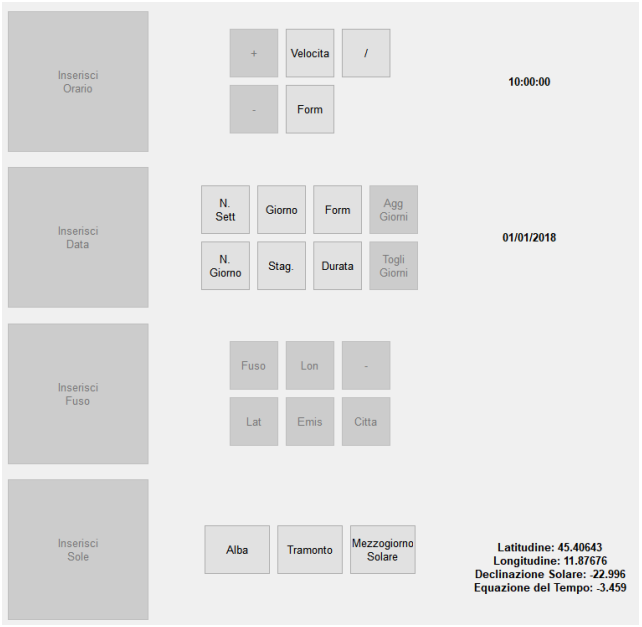
- E' possibile utilizzare le funzioni della classe Orario e della classe Data

- Fuso e Emis ritornano rispettivamente il fuso orario della città di appartenenza e l'emisfero

- Lat e Lon ritornano Latitudine e Longitudine

- - ritorna la differenza tra fusi orari con una città scelta in input

- Citta, dopo aver scelto una città, ritorna l'orario corrente di quella città con l'aggiunta del proprio fuso orario



- E' possibile utilizzare alcune funzioni della classe Orario e della classe Data, altre funzioni non disponibili per i motivi spiegati precedentemente nella relazione

- Alba ritorna l'orario in cui avverrà l'alba al giorno e posizione indicati

- Tramonto ritorna l'orario in cui avverrà il tramonto al giorno e posizione indicati

- Mezzogiorno Solare ritorna l'orario della massima elevazione del sole al giorno e posizione indicati

- Si vuole far presente che, per motivi di praticità, i calcoli di variabili importanti come Equazione Solare e Declinazione del sole verranno inseriti direttamente da input, per calcolarli abbiamo visitato questo sito: https://it.calcuworld.com/calendari/calcolare-sorgere-tramonto-sole/

- un esempio di utilizzo può essere per calcolare Padova del 1/01/2018 alle 10:00:00:

- Latitudine: 45.40643
- Longitudine: 11.87676
- Fuso Orario: 1
- Giorno: 1
- Mese: 1
- Anno: 2018
- Ora (attuali): 10
- Minuti (attuali): 0
- Secondi (attuali): 0
- Ora Legale (1 o 0): 0
- Equazione del Tempo: -3.459
- Declinazione Solare: -22.996

5) Suddivisione del lavoro

Abbiamo impiegato 51 ore per la realizzazione del progetto KALK. Così distribuite:

- Progettazione: 8 ore (di cui 5 ore per capire come implementare Sole)
- Implementazione classi: 10 ore
- GUI & MVC: 23 ore (di cui 3 ore per studio della libreria QT)
- Implementazione classi JAVA: 4 ore
- Debug: 3 ore
- Relazione: 3 ore

Ambiente di Sviluppo:

- Qt Creator 4.6.0 Based on Qt 5.10.1 (MSVC 2015, 32 bit), usato per gran parte del tempo
- Qt Creator 3.5.1 Based on Qt 5.5.1 (GCC 5.2.1 20151129, 64 bit) presente nella macchina virtuale
- Eclipse IDE for Java Developers, Version: Oxygen.3a Release (4.7.3a) per la parte JAVA