

Space Kalc

Silvio Meneguzzo

Introduzione

Cos'è Kalk?

Model

GUI

How to use

Caratteristiche  
kalc

# Qt Kalk

Silvio Meneguzzo

Università di Padova - Dipartimento di Matematica

*meneguzzosilvio@gmail.com*

23 Gennaio 2018

# Overview

Space Kalc

Silvio Meneguzzo

Introduzione

• Cos'è Kalk?

Model

GUI

How to use

Caratteristiche  
kalc

## 1 Introduzione

- Cos'è Kalk?

## 2 Model

## 3 GUI

- How to use

## 4 Caratteristiche kalc

# Cos'è Kalk?

Space Kalc

Silvio Meneguzzo

Introduzione

Cos'è Kalk?

Model

GUI

How to use

Caratteristiche  
kalc

Kalk è una calcolatrice che opera su tipi di dati non banali.

Questo progetto sviluppato durante il corso *Programmazione ad oggetti* ha lo scopo di creare un applicativo che possa essere un buon esempio di programmazione ad Oggetti.

Kalc mette a disposizione i classici operatori quali somma, sottrazione, moltiplicazione e divisione; la vera svolta sta nei tipi di dato che rappresentano tipi dimensionali colorati.

Ci sono 4 diversi tipi di dati tra cui è possibile svolgere operazioni; questi sono: oggetti ad una dimensione, oggetti a due dimensioni, oggetti a tre dimensioni e un oggetto RGBHex color che rappresenta un colore. Ognuno di essi ha un campo dpi che determina la definizione di stampa, inoltre la calcolatrice agisce anche da convertitore tra pixel, cm e inch.

# Model

Space Kalc

Silvio Meneguzzo

Introduzione

Cos'è Kalc?

Model

GUI

How to use

Caratteristiche

kalc

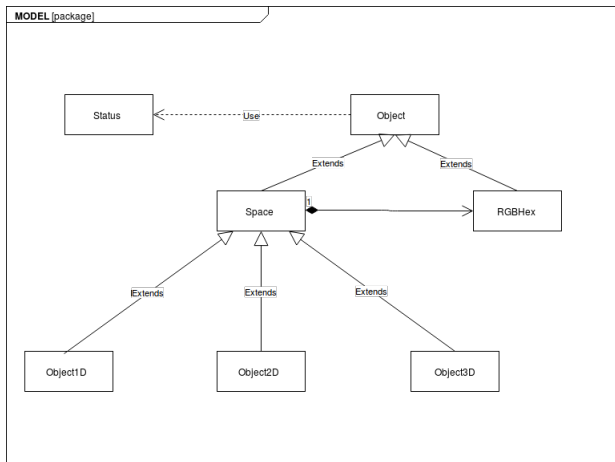


Figure: Gerarchia della parte Logica

# Classi Model

Space Kalc

Silvio Meneguzzo

Introduzione

Cos'è Kalc?

Model

GUI

How to use

Caratteristiche  
kalc

- **Object** rappresenta una classe da cui deriva tutta la gerarchia, cosichè quando la *BusinessLogic* andrà a fare operazioni sugli Object inseriti, a run time, eseguirà le operazioni esatte ed a aggiornerà lo *Status* in modo adatto in base al tipo dinamico dell'Object selezionato.
- **Space** classe Astratta rappresentante un Oggetto dimensionale avente un colore e una definizione di stampa (dpi), da cui poi ereditano *Object1D*, *Object2D* e *Object3D*.
- **Object1D** classe concreta che estende Space, possiede una lunghezza che determina se lo spazio dimensionale corrisponde a un punto unidimensionale (length=1) oppure un segmento.
- **Object2D** classe concreta che estende Space, possiede una lunghezza e una altezza e rappresenta un sottoinsieme del piano cartesiano (oggetto bidimensionale).
- **Object3D** classe concreta che estende Space, possiede una lunghezza, una altezza e una profondità e rappresenta un oggetto tridimensionale.

# Classi Model

Space Kalc

Silvio Meneguzzo

Introduzione

Cos'è Kalc?

Model

GUI

How to use

Caratteristiche  
kalc

- **RGBHex** classe concreta che deriva da `Object`, rappresenta un Colore RGB, costruibile tramite stringa (colore esadecimale), valori interi R G B. Questa classe copre semplicemente il ruolo di colorare *Object* e potenzialmente anche di interagire tramite operazioni su soli oggetti *RGBHex*.
- **Status** è una classe che svolge il ruolo di trasportatore; la classe `Status` è intrinseca in ogni classe derivante da `Object`, in quanto essa va ridefinita in base ai campi dati di quella classe.
- **BusinessLogic** è la classe che racchiude tutta la logica, tramite la funzione `esegui` che verrà richiamata dalla GUI con l'operatore "=" a run-time verrà scelto l'operazione adeguata ai tipi, inoltre fa da tramite con la GUI per tutto ciò che concerne la parte di `Model`.
- **Exceptions** è la classe che contiene le classi di eccezioni che possono essere sollevate.

La GUI (Graphical User Interface) è strutturata:

- 1 MainWindow, da cui parte la creazione della BusinessLogic che a cascata verrà passata a tutti i sotto oggetti che comporranno il QGridLayout generale della calcolatrice, dopodichè viene costruito *Contenitore*, che rappresenta, a tutti gli effetti, il layout generale di *Space kalc*.
- 2 Contenitore, è la classe che crea effettivamente gli oggetti che permetteranno la creazione di Object (tramite CreateObject), la selezione degli Object su cui si vuole fare operazioni (tramite Table) e infine la parte di calcolo (Calculate) che consentirà dopo aver inserito due Object tramite i QPushButton "add here", di selezionare un operatore e fare i dovuti calcoli.

# How to use GUI

Space Kalc

Silvio Meneguzzo

Introduzione

Cos'è Kalc?

Model

GUI

How to use

Caratteristiche

kalc

Più in dettaglio vediamo le 3 componenti di Contenitore:

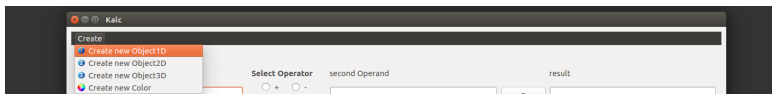


Figure: Create Object

Questa zona si trova nella parte superiore della calcolatrice dentro un QMenuBar, e attraverso Create (QMenu) si apre un menu a tendina (formato da QAction) che attraverso la pressione di quest'ultimi, vengono richiamati i costruttori di Oggetti e mostrati in una Finestra che si sovrapporrà a MainWindow.



Figure: Selection Object



# How to use GUI

Space Kalc

Silvio Meneguzzo

Introduzione

Cos'è Kalc?

Model

GUI

How to use

Caratteristiche  
kalc

Una volta Creati gli Object, verranno inseriti all'interno degli appositi QListWidget (tramite iterazioni sui vector corrispondenti, fatti nella BusinessLogic che è stata precedentemente modificata con la creazione); questi Oggetti sono indicati da un indice e una volta selezionato un Object viene fatto vedere nella sezione *Details* i rispettivi dettagli, inoltre è possibile vedere la corrispondente lunghezza in cm o inch.

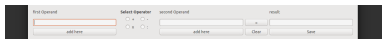


Figure: Calculate Object

Una volta selezionato un Object da un QListWidget è possibile aggiungerlo tramite uno dei QPushButton "add here", una volta aggiunti due Object, sarà possibile cliccare su un "QRadioButton" per selezionare l'operazione e avviare execute tramite il pulsante "=", se l'operazione sarà possibile tra i due tipi di Object verrà fatto vedere il risultato, altrimenti apparirà un "QMessageBox" rappresentante il tipo di errore.

- Polimorfismo è intrinseco nel Progetto
  - operatori virtuali puri in Space, a run time verranno chiamati quelli ridefiniti nelle classi concrete.
  - getStatus funzione virtuale ridefinita in tutte le classi, a run time verrà aggiornato in base al tipo dinamico dell'Object su cui è stata invocata la funzione.
  - Unico modo in cui faccio *RTTI* è tramite cast dinamico.
- Gli errori sono gestiti in due modi:
  - per la parte Logica è stata creata una classe di eccezioni che vengono sollevate qual'ora le operazioni di 2 Object, invocate tramite execute(), non fossero possibili, inoltre sono stati fatti controlli preventivi per prevenire chiamate incomplete.
  - per la parte Grafica sono stati inseriti controlli preventivi, e qual'ora la correttezza dell'inserimento dei dati o l'operazione selezionata non fossero possibili, verrà visualizzato un QMessageBox all'interno del quale sarà descritto il tipo di errore.

# Caratteristiche

Space Kalc

Silvio Meneguzzo

Introduzione

Cos'è Kalc?

Model

GUI

How to use

Caratteristiche  
kalc

- La gestione della memoria è minoritaria in questo caso, poichè l'applicazione non fa uso pesante della memoria, l'unica cosa che viene allocata e modificata è la BusinessLogic nella GUI, ma quest'ultima non viene ricreata, ma viene passata attraverso tutte le varie componenti e modificata. Mentre la Table che tiene visibili e aggiornati gli Object presenti nella BusinessLogic, viene ogni volta distrutta e ricreata.
- *Design Pattern* utilizzato è MVVM (Model View ViewModel) Questo pattern propone un ruolo più attivo della View rispetto a MVC: la View è in grado di gestire eventi, eseguire operazioni ed effettuare il data-binding. In questo contesto, quindi, alcune delle funzionalità del Controller vengono inglobate nella View, la quale si appoggia su un'estensione del Model.
- Le ore rendicontate per lo svolgimento del Progetto sono pari a **50**, suddivise tra analisi dei Requisiti, decisioni progettuali, codifica, stesura relazione, trascrizione in Java, verifica e accettazione complessiva.