



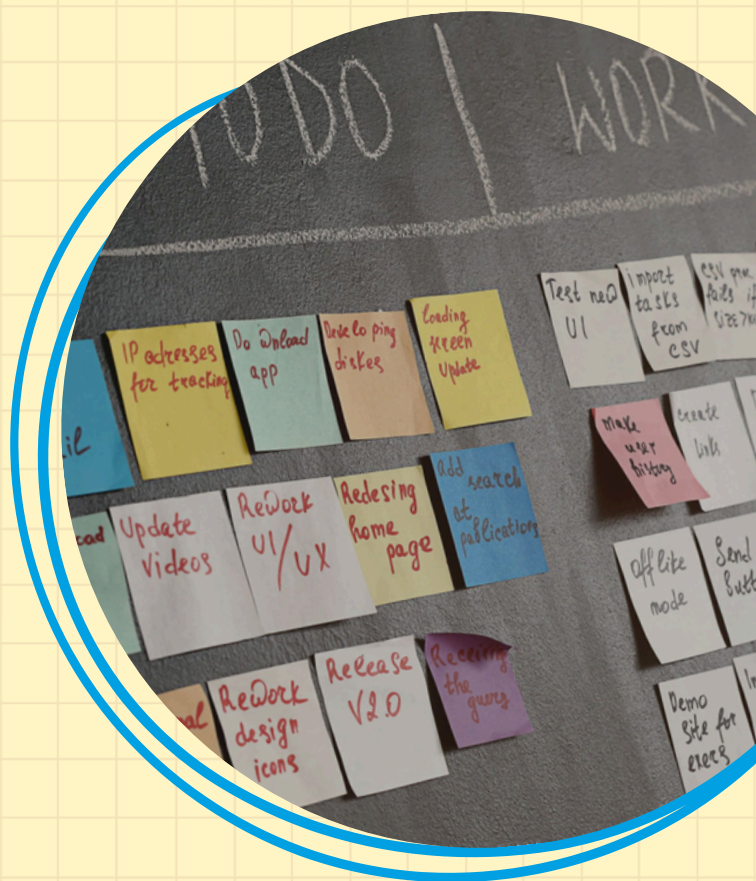
A GEOMETRIC DRAWING PROGRAM

A.A. 2024/2025

Class Diagram

Docente:

Pierluigi Ritrovato



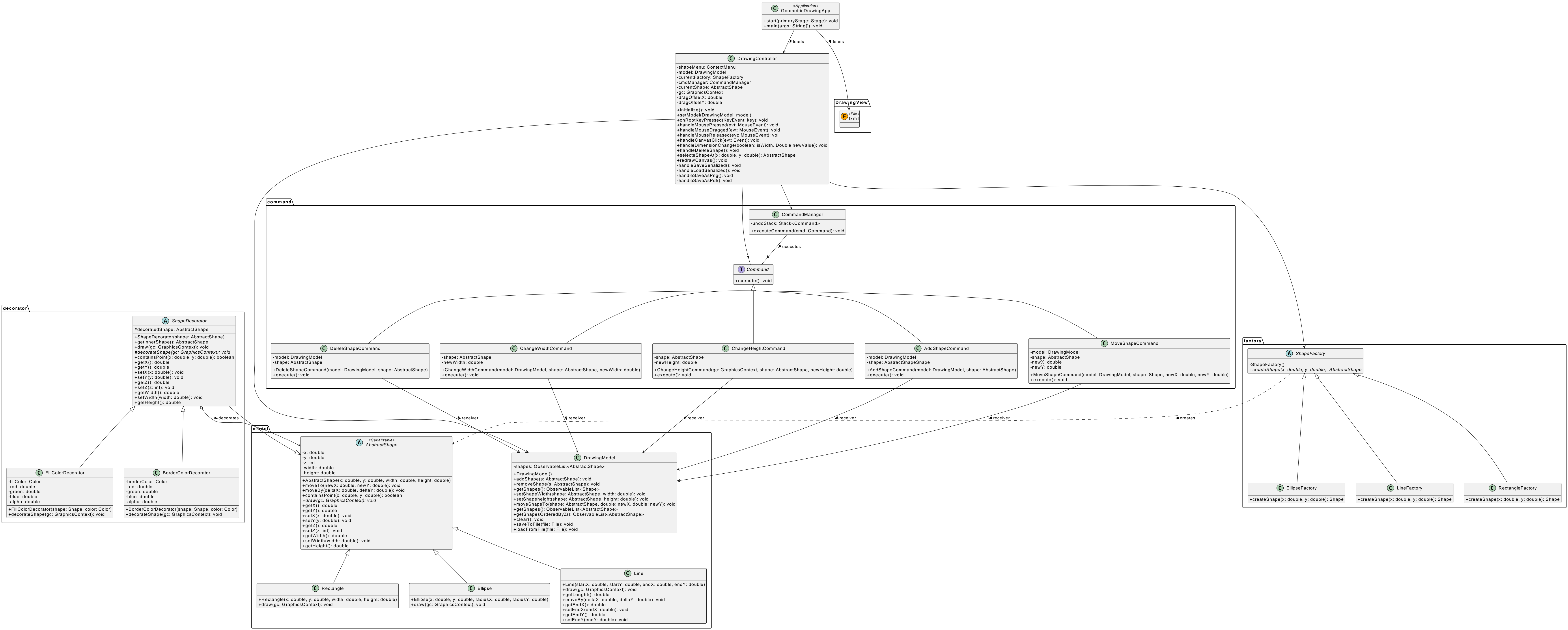
A cura del gruppo 5:

Apicella Antonio 0622702531

Celano Benedetta Pia 0622702558

Cuomo Carmine 0622702688

Guerra Simone 0622702675



Detailed Class Diagram

Grazie all'identificazione dei pattern già effettuata dal team nella fase di PreGame, la realizzazione di un class diagram dettagliato è stata facilitata. Di seguito si riporta una spiegazione delle classi principali:

Main

In cima al diagramma vi è la classe *GeometricDrawingApp* che ospita l'entry point dell'applicazione e che estende *Application* facendo l'override del metodo `start`.

Pattern Factory Method

Di seguito si illustra come si è proceduto all'applicazione del pattern:

ShapeFactory

La classe astratta che funge da Creator. Ogni factory concreta (`EllipseFactory` - `RectangleFactory` - `LineFactory`) implementa il metodo `createShape(x, y)` per cui x ed y rappresentano le coordinate del vertice in alto a sinistra della figura (estremo sinistro se una linea).

AbstractShape

La classe astratta che funge da Product e che unifica le proprietà comuni a tutte le figure come le coordinate (x, y) di partenza, le dimensioni, e la terza dimensione z per rappresentare l'"altezza della figura" nel disegno. Possiede il metodo `draw(GraphicsContext gc)` esteso dalle classi figlie concrete (`Ellipse` - `Rectangle` - `Line`) per il rendering della figura su Canvas.

Pattern Decorator

ShapeDecorator

Una classe astratta che estende *AbstractShape* (quest'ultima funge da Component), mantiene un riferimento ad un'istanza di questa classe affinché le classi concrete figlie (`FillColorDecorator` - `BorderColorDecorator`) possano aggiungerne responsabilità tramite la specifica rispettivamente del colore di riempimento e di bordo.

Pattern Command

Command Manager

Si tratta della classe concreta che mantiene lo stack di tutti i comandi che sono stati effettuati

Command

Rappresenta l'interfaccia comune col metodo `execute()` di cui tutte le classi concrete - `ChangeWidthCommand`, `ChangeHeightCommand`, `MoveShapeCommand`, `AddShapeCommand`, `DeleteShapeCommand` - ne effettuano l'override.

DrawingController

Rappresenta il controller nel pattern MVC. Nello specifico:

- Mantiene un riferimento ad un'istanza di `ShapeFactory` che incapsula la logica per la creazione di una nuova figura,
- Mantiene un riferimento all' `AbstractShape` per la figura attualmente selezionata o inserita;
- Mantiene un riferimento al `CommandManager` per coordinare l'esecuzione dei vari comandi.

Si occupa inoltre della coordinazione tra il model (`DrawingModel`) e la View - quest'ultimo mediante la conservazione del riferimento al `GraphicsContext` ottenuto dal Canvas, aggiornando l'interfaccia ridisegnando il Canvas ad ogni cambiamento nel model.

Drawing Model

Funge da receiver per le concrete command e dunque ha come compito principale quello di mantenere la lista ordinata di `AbstractShape` e gestire l'aggiunta, la rimozione e gli aggiornamenti.