Design Pattern Utilizzati

Command Pattern

Per implementare i comandi dell’interprete (creazione, eliminazione, spostamento, ridimensionamento, etc.) e supportare l'operazione di undo.

Factory Pattern

Per la creazione degli oggetti grafici (cerchi, rettangoli, immagini) in modo centralizzato e flessibile.

Composite Pattern

Per gestire gruppi di oggetti come un'unica entità (ad esempio, un gruppo di cerchi e rettangoli che possono essere manipolati insieme).

Singleton Pattern

Per la gestione centralizzata dello stato dell’applicazione, come il gestore delle forme e il registro dei comandi eseguiti.

Observer Pattern

Per aggiornare dinamicamente l’interfaccia utente o altre componenti quando un oggetto grafico viene modificato.

Memento Pattern

Per salvare lo stato degli oggetti e supportare l’operazione di undo senza violare l'incapsulamento.

Punti chiave per eseguire al meglio il progetto

Analisi dei Requisiti

Definire in dettaglio i comandi richiesti dall’interprete e i comportamenti attesi per ciascun oggetto grafico.

Identificare i requisiti funzionali e non funzionali.

Progettazione Software (UML & Design Pattern)

Creare i diagrammi UML per rappresentare l’architettura software.

Scegliere i pattern più appropriati per la gestione dei comandi, della creazione degli oggetti e della gestione dello stato.

Implementazione dell’Interprete di Comandi

Usare StreamTokenizer per l’analisi lessicale dei comandi.

Implementare il Command Pattern per ciascuna operazione supportata.

Gestione degli Oggetti Grafici

Creare una gerarchia di classi per rappresentare gli oggetti (Circle, Rectangle, Image).

Usare il Factory Pattern per la loro creazione dinamica.

Gestione dei Gruppi di Oggetti

Implementare il Composite Pattern per consentire operazioni su gruppi di oggetti.

Supporto per Undo/Redo

Salvare lo stato precedente di ogni oggetto usando il Memento Pattern.

Mantenere una pila di comandi per consentire operazioni di undo e redo.

Test e Validazione

Scrivere test unitari con JUnit per validare il corretto funzionamento dell’interprete e delle operazioni sugli oggetti.