Project Document

AUTHOR - Mani Ratna Sai Prabhu Teja, Anem

Objective: Purpose of this document is to give the details of the design patterns used in the project

Project Overview: This is Project where we will be building an “MS Paint”-like application in Java called JPaint

Features to be Created

Pick a shape

- o Ellipse
- o Triangle
- o Rectangle

- Pick a primary colour
- Pick a secondary colour
- Select shading type (outline only, filled-in, outline and filled-in)
- Click and drag to draw a shape
- Click and drag to select shapes
- Click and drag to move selected shapes
- Copy selected shapes
- Paste copied shapes
- Delete selected shapes
- Undo last action
- Redo last action
- Group selected shapes
- Ungroup selected shapes
- Selected shapes have dashed outline

Below are the design patterns implemented in the project so far

1. Factory Method Pattern

2. Command Pattern

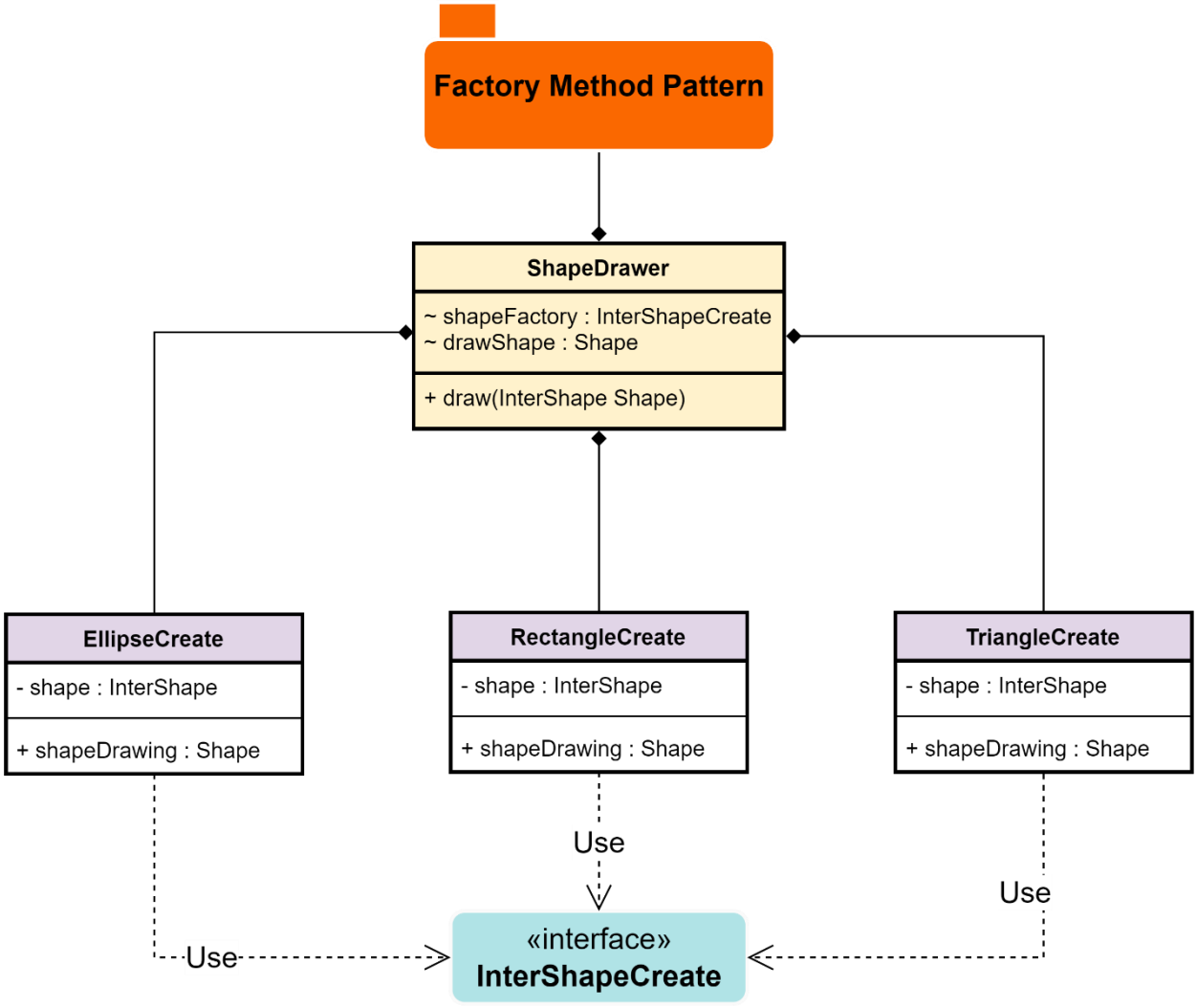
3. Strategy Pattern

4. Builder Pattern

5. Decorator Pattern

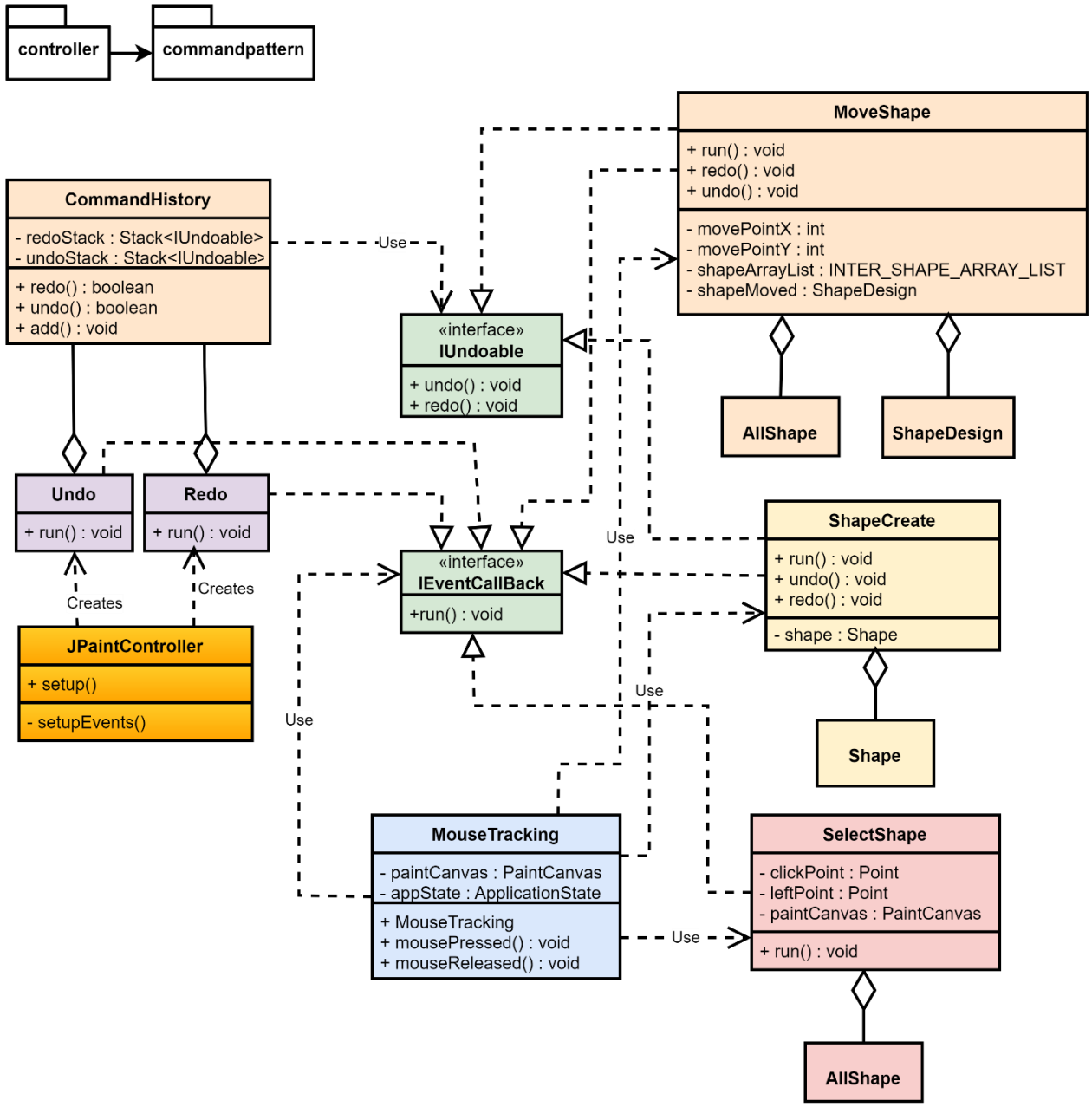
Factory Method

InterShapeCreate, RectangleCreate, EllipseCreate, TriangleCreate, and ShapeDrawer are the classes I used in the factory method pattern.



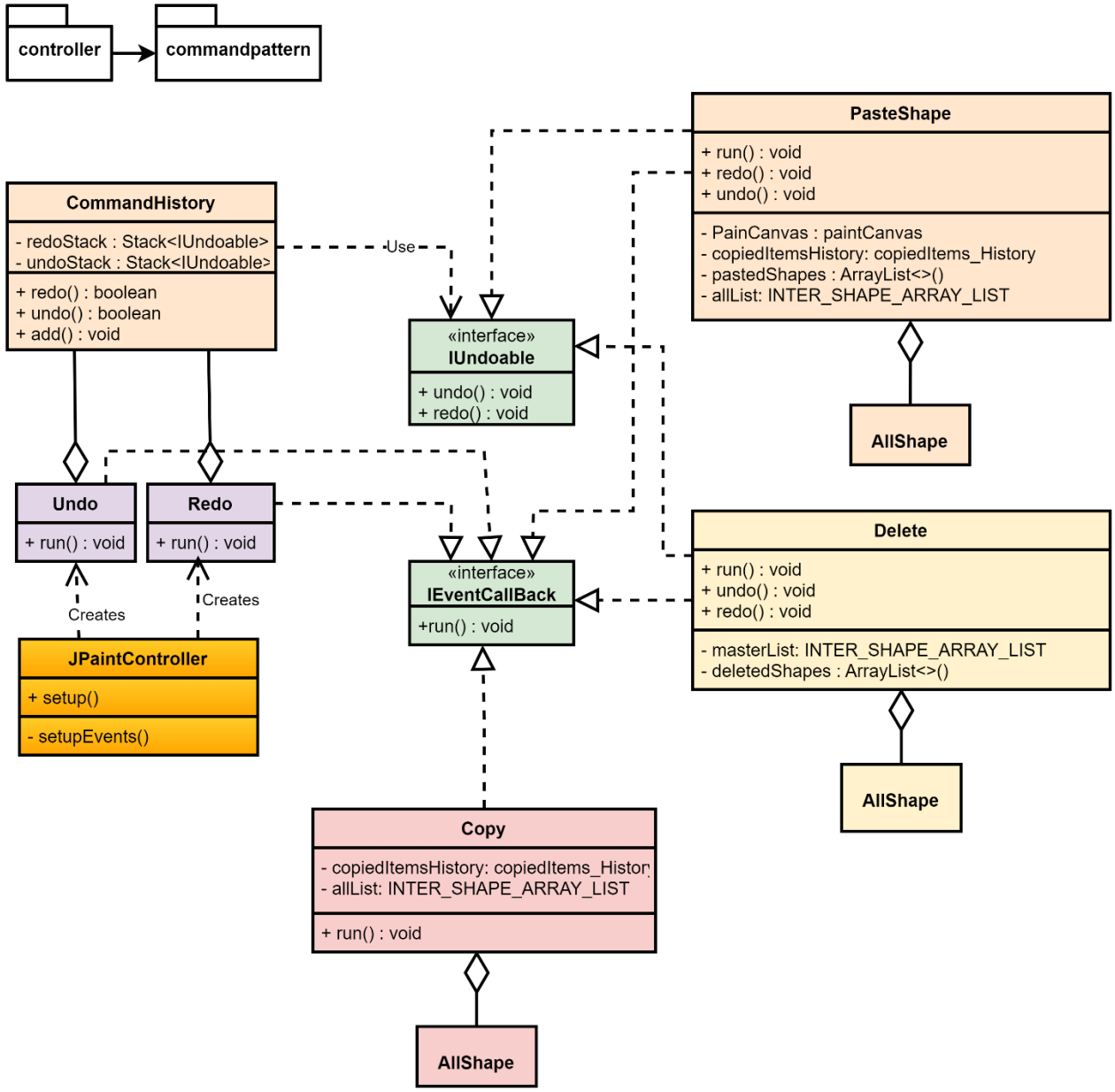
2. Command Pattern

Implemented classes “Command History”, “ShapeCreate”, “MoveShape”, “SelectShape”, “MouseTracking”, “Undo”, “Redo”, “JPaintController”, “Shape”, “AllShape”, “ShapeDesign” and interfaces “IEventCallBack” and “IUndoable” in command pattern



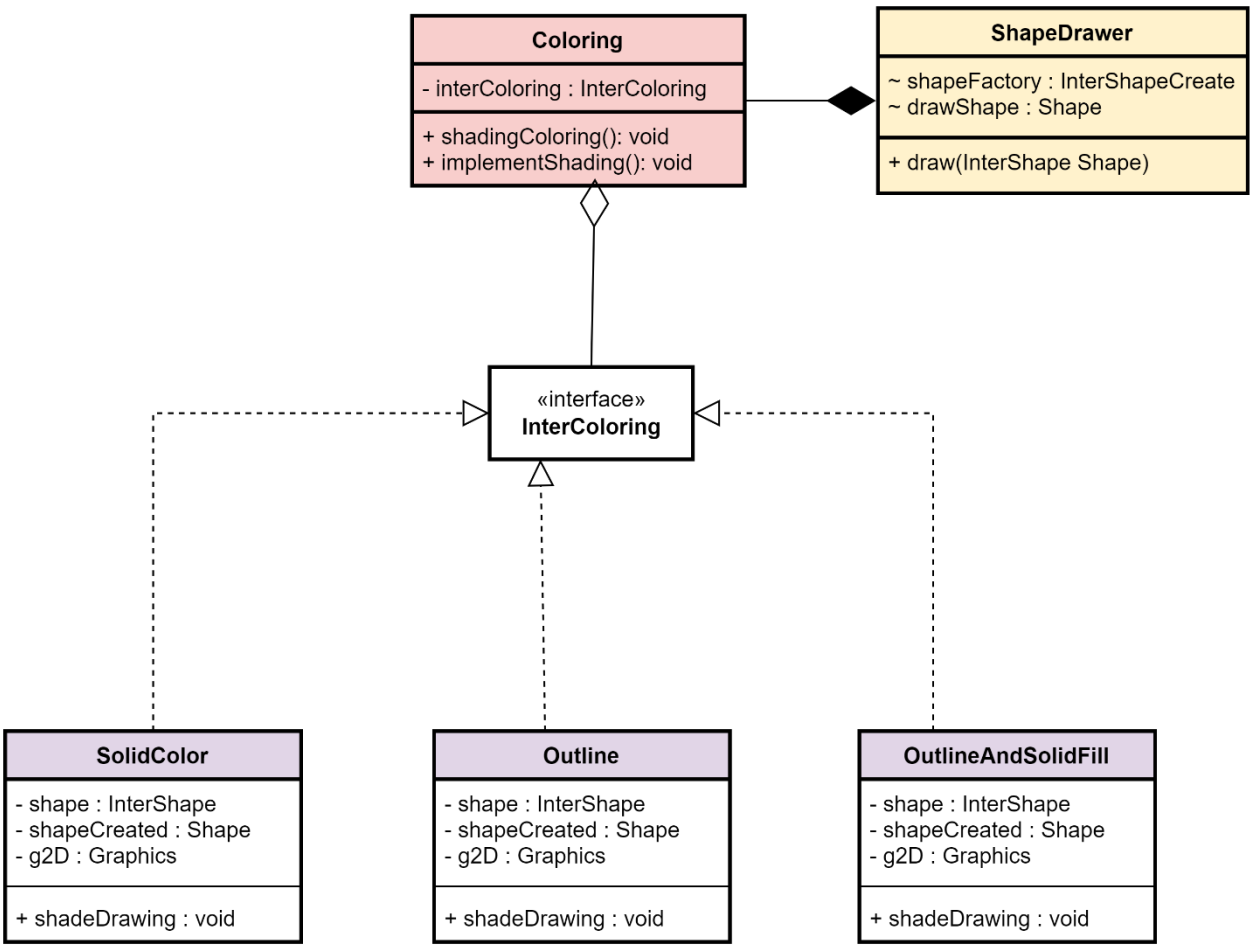
Command Pattern

Implemented the functionalities “Copy”, PasteShape” and “Delete” in command pattern



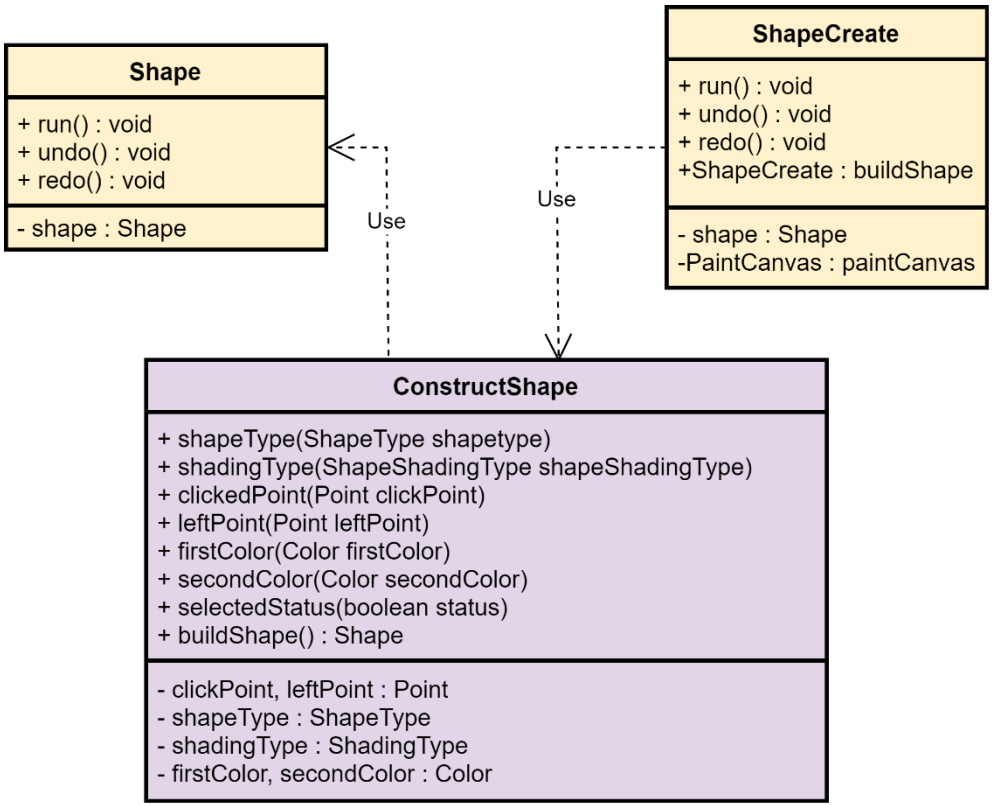
3. Strategy Pattern

Implemented “SolidColor”, “Outline” and “OutlineAndSolidFill”, “Coloring” classes and Interface “InterColoring” in Strategy Pattern.

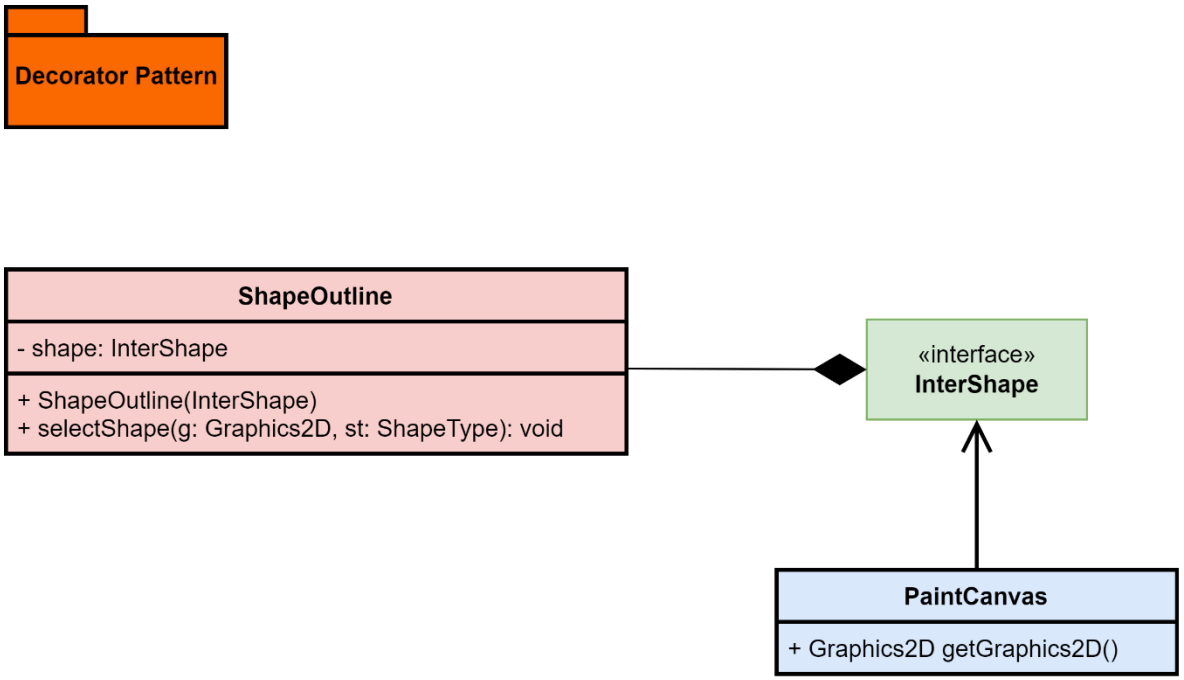


4. Builder Pattern

Implemented “ConstructShape” to draw all the shapes using Builder Pattern.



5. Decorator Pattern

Implemented the function(class) to draw a dashed outline border when a shape is selected using the decorator pattern.

The "ShapeOutline" class wraps the "InterShape" interface object and provides additional functionality to draw a border around the shape object. The "selectShape" method of the "ShapeOutline" class adds a black border with a specific stroke around the shape object. The Decorator pattern is used to add functionality to an object dynamically at runtime without affecting the behaviour of other objects in the same class hierarchy.