Lab3-OOP-Drawing App

Project-tasks:

Task1:

1. Design an object-oriented model that covers the following geometric shapes: Line Segment, Circle, Ellipse, Triangle, Rectangle, and Square.

2. Draw a UML Class diagram that represents your model, showing all the classes, attributes and methods.

3. Apply the concepts of inheritance and polymorphism to your design.

Task2:

1. Implement your design from part 1.
2. Design and implement a GUI that allows the following functionalities for the user on all the shapes defined in part 1: Draw, Color, Resize, Move, Copy, and Delete. (Optional hint: check “Factory DP, and Prototype DP”).
3. Implement your application such that it would allow the user to undo or redo any action performed.
4. The cursor should be used to select the location of a shape while drawing it, or moving it to another location, for more accurate control of the shape parameters (ex: size), dialog boxes could be used, or you are free to implement it in a more user-friendly way of your choice. (Optional hint: draw by mouse dragging).

Task3:

1. Provide an option in UI to save the drawing in XML (encoding: ISO-8859-1) and JSON file (You should implement both).

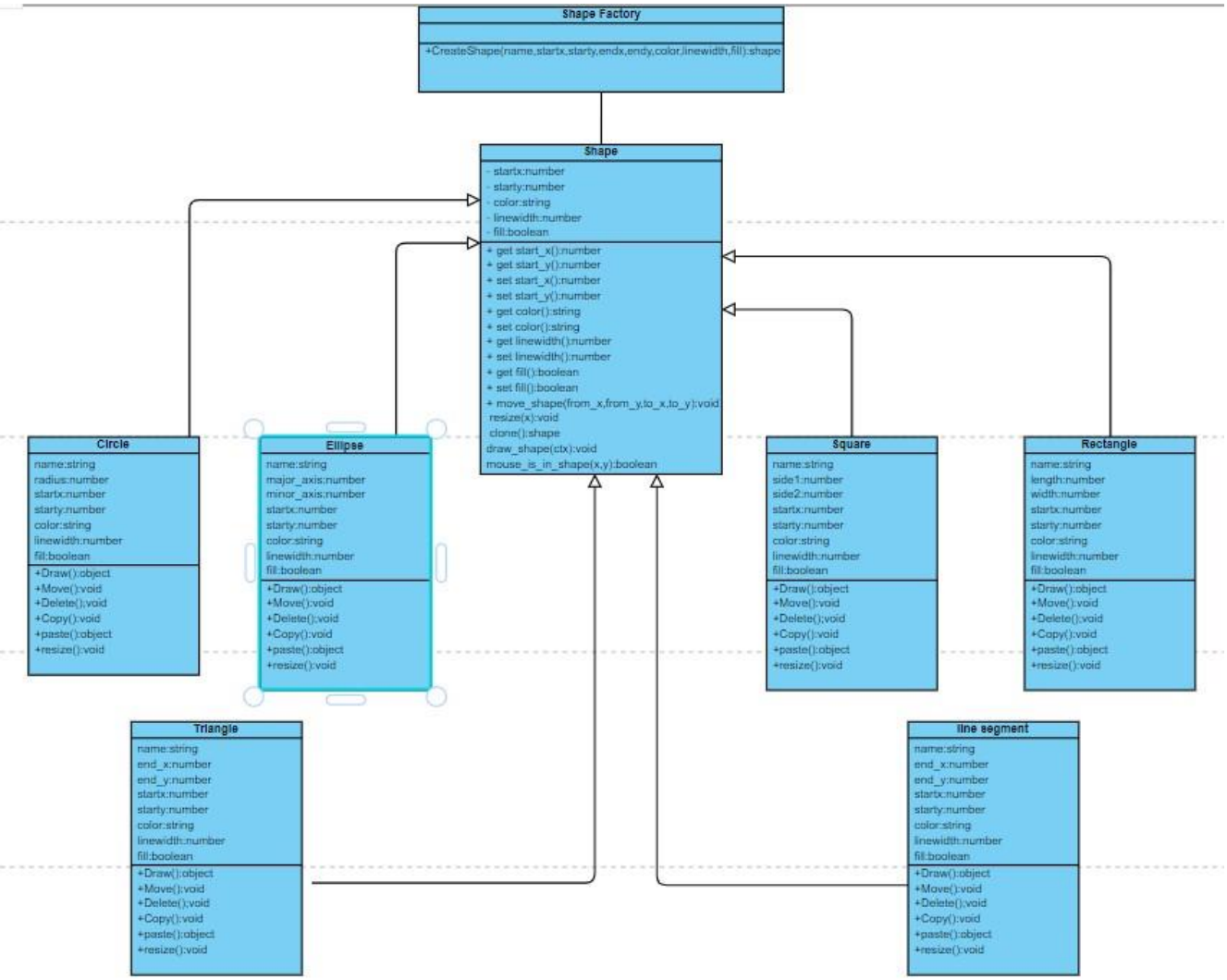
2. Provide an option to load previously saved drawings and modify the shapes.

3. Users must choose where to save the file.

How to run the code:

1. We want to open "front" folder: open vs code → choose (oop_lab3) folder then select "front" folder to be in vs code
2. Drawing →Src → app → app.component.ts
3. Open a new terminal → write "cd Drawing"
4. Then write "ng serve"
5. The localhost 4200 will open automatically
6. If the server doesn't open, write (http://localhost:4200/) on any browser such as google, chrome → Drawing app must appear
7. Now, we need to open Spring boot.
8. We want to open "myFirstApp" folder: Open new vscode window →choose "oop_lab3" folder→ then choose "back" folder → then select "myFirstApp" folder to be in vs code
9. Src →main→java→MyFirstAppApplication.java→Run the program → the localhost 8080 will start working but don't open it →if server work without saying "application failed to start" →drawing app is ready
10. If the application failed to start → try to delete all the terminals you open from right down corner in vs code → then run the code again→ server must work now.
11. Finally, try to paint in the app at server 4200 and it must work correctly.

UML Class Diagram Used:



How I applied the design pattern:

1. Factory Design Pattern:

our app uses the mouse to draw shapes

Just like normal Paint (draw by mouse dragging), when the mouse is clicked within the canvas (mouse is down) after choosing the desired shape the start point of the shape is selected then the user moves the mouse (while it is still clicked) until they reach their desired end point when they stop clicking (mouse is up or left the canvas) the end point is selected.

the name of the shape, the start point, the end point and other shape attributes such as line width, color, filled or not is passed to CreateShape() function in ShapeFactory to create the shape.

the function calculates other properties of the shape using the start and end point for example

radius for circle

major and minor axis for ellipse

width and length for rectangle

Side length for square

then uses the calculated properties and the other attributes to create and return the shape.

2. prototype Design Pattern

prototype is used whenever we want to create a new instance of an existing shape.

this is used in copy and other parts of the code.

the clone function (in the Shape Class) returns new instance of the shape with the same properties.

Design decisions:

we divided the design into:

- In the Back-end “Springboot”:

  we use it for the following function:

  1. Save
  2. Load

- In the Front-end “Angular”:

  we use it for the following function:

  1. Move
  2. Copy
  3. Delete
  4. Resize
  5. Undo
  6. Redo
  7. Drawing shapes:
     1. Line segment
     2. Square
     3. Rectangle
     4. Triangle
     5. Circle
     6. Ellipse

  We used “canvas” to help in drawing shapes.

  The “ShapeFactory” was used in the front-end part.

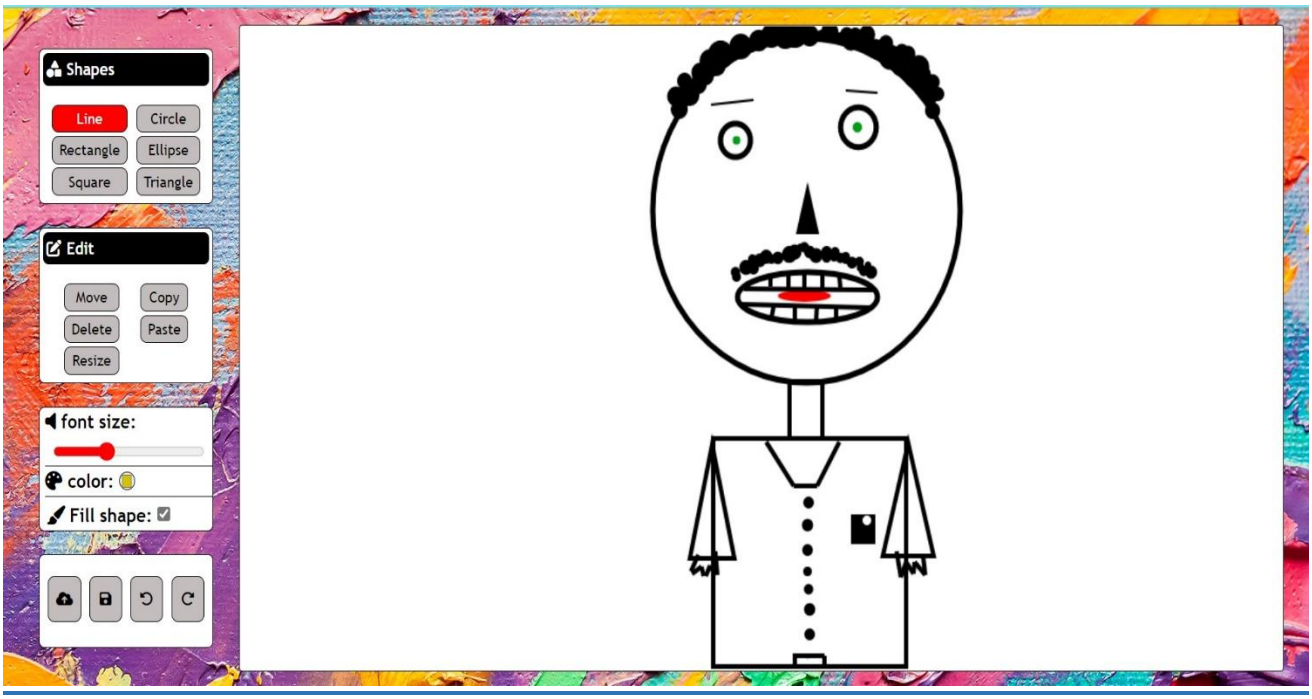
The programming language used:

Back-end → Java (Springboot)

Front-end → HTML, CSS, TypeScript (like JavaScript) (Angular)

Ui design:


Shapes
Line
Triangle
Rectangle
Circle
Square
Ellipse
Edit
Move
Copy
Delete
Paste
Resize
font size:
color:
Fill shape:


Shapes
Line
Circle
Rectangle
Ellipse
Square
Triangle
Edit
Move
Copy
Delete
Paste
Resize
font size:
color:
Fill shape:

User Guide:

The app is very simple to understand:

Drawing:

1. If you want to draw a line segment press on it, start to draw and the cursor will be at one of its endpoints.
2. If you want to draw a circle or ellipse press on them and start drawing the cursor will be at the center
3. If you want to draw a rectangle or square press on them and the cursor will be at the edge.
4. If you want a triangle press on it and the cursor start at the vertex

Editing:

1. If you want to Move any shape press on it and drag it to another position.
2. If you want to delete any shape, press the delete button and select the shape.
3. If you want to Copy, press the copy button then select the shape, then press the paste button then select the position.

Note: you can paste the shape more than once.

4. If you want to Resize the shape, press the resize button then select the shape a slider will appear drag it and select the size you want.

Advanced edit:

There are four icons at the end of the sidebar used for:

1. Undo to return to the previous step
2. Redo used to go to the following step if you make undo.
3. Save button is used to save the drawing you draw in the JSON file or XML file
4. Load button used to load your work again.

How to save:

1. Draw your work
2. After finishing, you press the save button
3. An alert will appear to enter the file name (you can name the file whatever you want but add the extension in the end like: “name.xml” or “name.json”)).
4. You can save the file in two formal (.xml, .json)

How to load:

1. Press the load button
2. An alert will appear to enter the file name(enter the same file name you entered while saving like: “name.xml” or “name.json”)
3. The file then will automatically load to the drawing page.
4. You can make edits to it whatever you want.

Note 1: in case you loaded any work you can’t return to the previous work you made by undo button, so be careful to save your work before loading any other work.

Note 2: you can distinguish between the icon by hovering on the icon (put the cursor over it) then a title of the icon name will appear.

Colors:

There is a button for choosing a color, press it and select the range

Font size:

There is a slider to select range from 1px →10px

Fill checkbox:

Used to make the shape is filled from the center (ex: the whole shape become black).