Assignment #3:Paint

1 Objectives

Upon completion of this assignment, you will be able to:

- Design an object-oriented model for geometric shapes.
- Draw a UML class diagram that represents your model.
- Apply the OOP concepts of inheritance and polymorphism to your design.
- Create an advanced UI with 2D Graphics capabilities.
- Dealing with JSON for requests and responses handling.

2 Part 1: Geometric Shapes Data Model

2.1 Description

Geometric shapes belong to different groups (ex: Elliptical Shapes, Polygons, Sectors...etc.); members of these different groups are related to each other in the sense that they share common properties. To be able to implement an efficient and object-oriented drawing application, it is essential to design a model that takes these relations into consideration.

2.2 Tasks

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.

3 Part 2: Drawing and Painting Application

3.1 Description

Drawing and painting applications are very popular and have a huge user base; they generally offer a big number of features that include but are not limited to: Drawing, Coloring, and Resizing. They also include several built in, and possibly extensible set of geometric shapes,and classically, they allow the user to undo or redo any instructions to make the application more usable.

3.2 Tasks

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 on 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).

4 Part 3: Save and Load Description

4.1 Description

One of the main features in any paint application is saving user's drawings in a file and modifying it later.

4.2 Tasks

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.

5 Deliverables

- You should work in groups of four.
- Develop this assignment in Java Spring Boot and Angular.
- You should provide implementation for the given requirements.
- You should deliver a report that:
  - Describes thoroughly a full list of the steps required to run your code.
  - Includes a UML diagram describing your code design thoroughly.
  - Describes thoroughly how you have applied the required design pattern in your code.
  - Includes any design decisions that you have made should be listed clearly.

  - Includes snapshots of your UI and a user guide that explains how to use your application.

- Upload your report, and source code zipped to Microsoft teams.
- Be creative! The required features are only the beginning of what you can do, add more features or spice up the required ones, bonus marks will be given to those with eye-catching extra features and user-friendly interfaces.
- Delivering a copy will be severely penalized for both parties, so delivering nothing is so much better than delivering a copy.