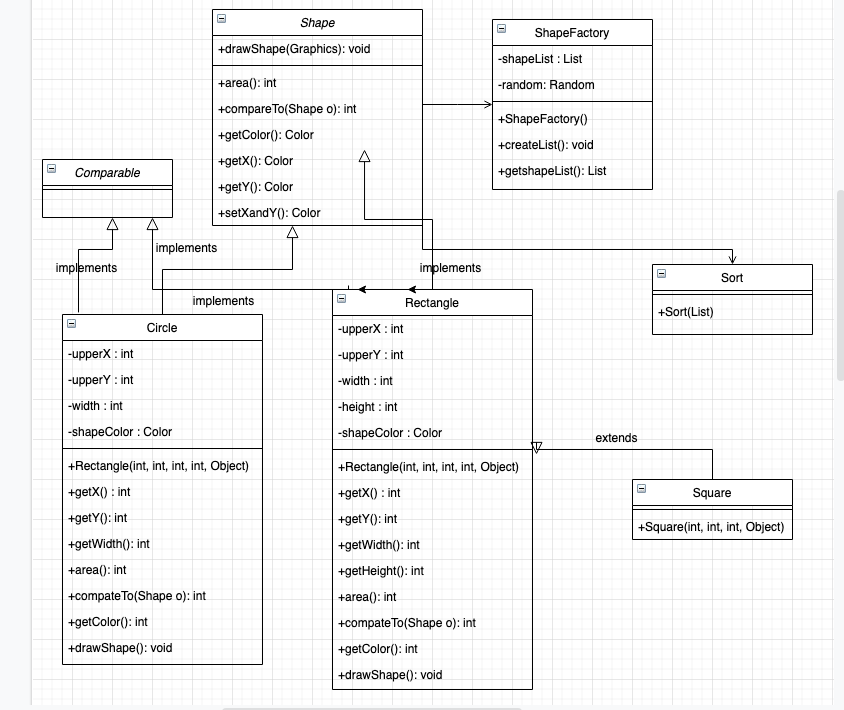
**Part 1: Introduction**

First of all, this report will include 4 different parts to explain the project detailly. First part is Introduction, second part is going to be about design of the solution, third part will be implementation of the solution, last part will be conclusion. This project aims to make an application to display an interface with two buttons in order to create random shapes and sort them by using those buttons, such as load and sort button. In order to make this project, I have utilized some of the Object-Oriented Programming concepts, such as interface and inheritance, and some of the design patterns like singleton and factory design pattern. Also, this project is created with Java. The main challenges of this project is to learn these design patterns that I mentioned above and how JPanel works generally.

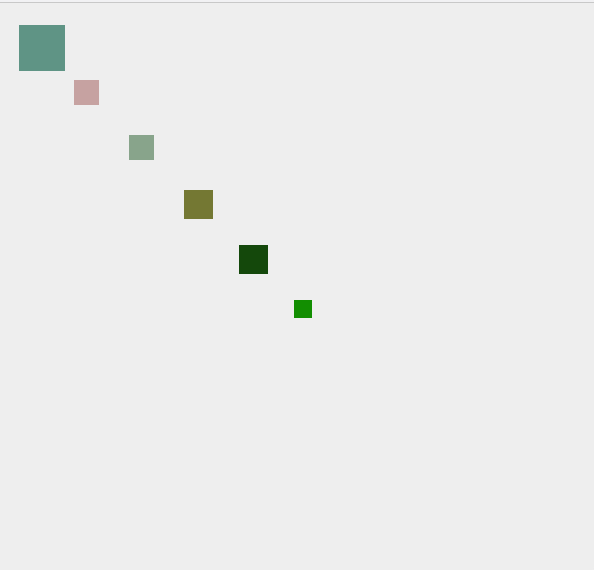
Go to Next Page for Rest of the parts 🡪

**Part 2: Design of the Solution**

Regarding my design of this project (you can check the UML Diagram Above first) is that I mainly used interface concept from Object oriented programming to reach what I have planned. Rectangle and Circle classes connects to Shape interface in my solution. This shape interface includes some methods which must be overridden and used in Rectangle and Circle classes. These methods are drawShape, compareTo, getColor, and area. The reason of why we include these into the Shape interface is that these methods are the general methods that have to be included in Rectangle and Circle classes to complete the project. Other class that we have is ShapeFactory class. This class’s purpose is to utilize factory design pattern in this project which helps us create shapes and put them in a list to display in the panel randomly. Therefore, we can say that this class is the heart of this project because the main information and process is done here. Another class that I have used is sorting, I have done the sorting with selection sort and the reference of my sorting solution is https://www.geeksforgeeks.org/selection-sort/

Another approach for this project’s design would be using inheritance for Shape, Rectangle, Circle, and Square classes. Then, the subclasses, Rectangle, Circle, and Square could get the needed methods and attributes from the parent class, Shape. However, I believe that using inheritance makes this design of the project more complicated and decreases the usability. Therefore, this is the reason why I chose interface design for Shape class to make it more usable, easier, and understandable.

**Part 3: Implementation of the solution:**

****

For the implementation of the solution, I utilized Eclipse IDE in order to write and run the code. As I have mentioned, this project is created with Java and Eclipse is one of well known IDE to make projects with Java. I used Java-13 SDK for this project. For how I made the classes, first thing I have done is to create a panel called MyPanel class in this project to test that it works fine or not. Then, I moved on to make the Shape Interface first, it was very quick and easy because the only thing that we add there is the methods that we want to use in rest of the classes like Rectangle, Square and Circle. After that, I finished my Rectangle and Circle classes. First, I initialized the variables of these classes like upperX and upperY integers. Then, I made their constructors and created the methods that we need to implement (which come from the Shape Interface) and made the getters/setters. After that, I made the Square class and connected it to Rectangle by extending. Then, I worked on the ShapeFactory class and initialized the shapeList. After that, I made the needed methods like creating the shape, creating the shape list, and getting that list. Lastly, I made my sorting class with selection sort algorithm and as I referenced above, I used geekforgeeks website for this sorting method. This is all I have done for the project.

**Part 4: Conclusion**

The project generally went well. I have learned many concepts and design patterns from this project. The problem I had was the timing because I had so many job interviews in this week. Because of these kinds of problems, I have struggled with time. If someone wants to start this project, he/she should start by learning the concepts and then begin the coding part.