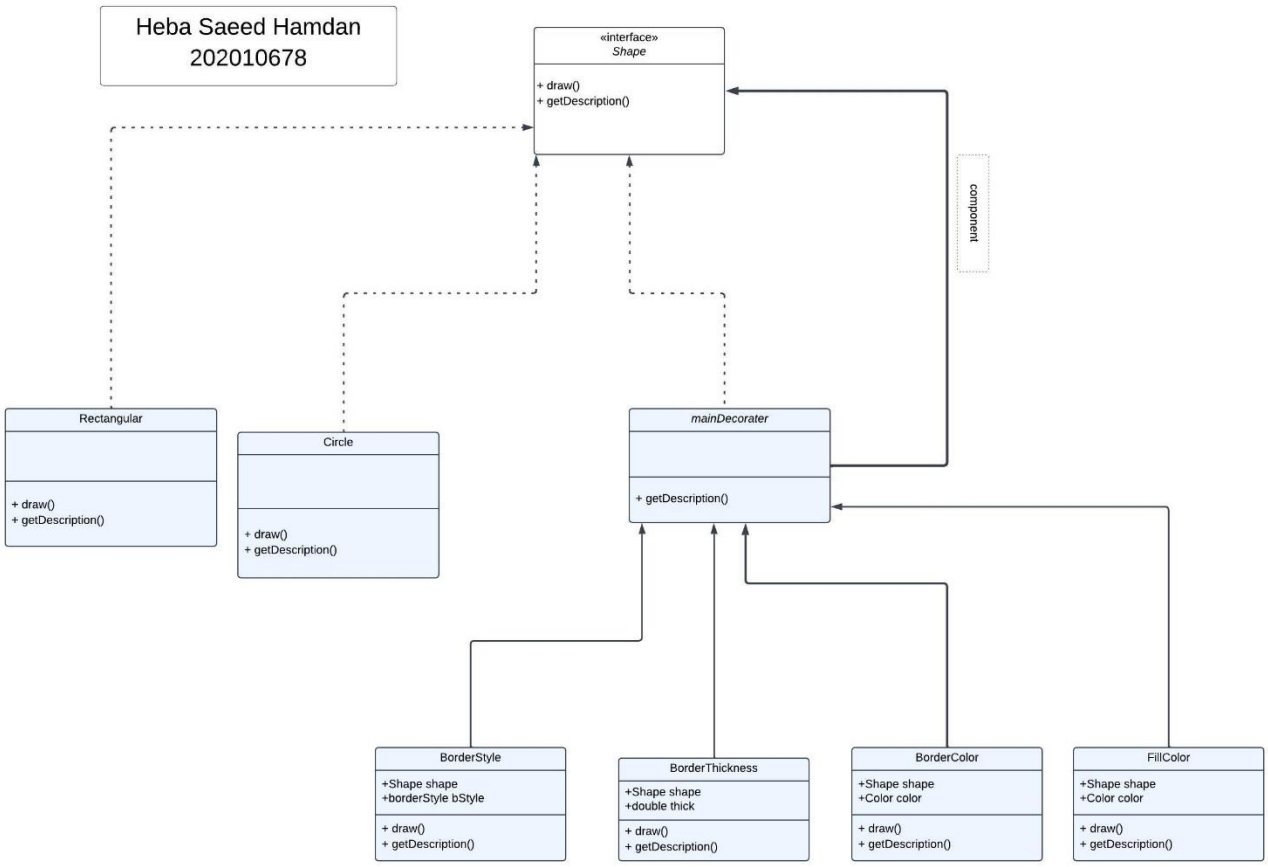
Design Patterns-Assignment1

Class diagram:



- Shape: main component.
- Rectangular , Circle: concrete components.
- mainDecorator: main Decorator.
- BorderStyle, BorderThickness, BorderColor, FillColor: concrete decorators.

We can suppose that border as 1 class and the concrete decorators will be:

- Border ” with color, thickness and style parameters ” .
- FillColor.

- **Shape “interface” main component**

```
public interface Shape {
   void draw();
   String getDescription();
}
```

- **Rectangular “Class” concrete component:**

```
public class Rectangular implements Shape{
   @Override
   public void draw() {
      System.out.println("Drawing Rectangular");
   }
   @Override
   public String getDescription() {
   return "Rectangular";
   }
}
```

- **Circle “Class” concrete component:**

```
public class Circle implements Shape{
   @Override
   public void draw() {
      System.out.println("Drawing Circle");
   }

   @Override
   public String getDescription() {
   return "Circle";
   }
}
```

- **mainDecorator “abstract class” main decorator:**

```
public abstract class mainDecorater implements Shape{
   @Override
   public abstract String getDescription();
}
```

- **FillColor “Class” concrete decorator:**

```
public class FillColor extends mainDecorater{
 Color color;
 Shape shape;
public enum Color{
 RED,
GREEN,
BLUE,
WHITE,
BLACK
}
public FillColor(Shape shape,Color color){
this.color=color;
this.shape=shape;
}
   @Override
   public String getDescription() {
   return shape.getDescription()+"\nfill-color: "+color;
   }

   @Override
   public void draw() {
   shape.draw();
   System.out.println("fill-color: "+color);}}
```

- **BorderColor “Class” concrete decorator:**

```
public class BorderColor extends mainDecorater {
    Color color;
    Shape shape;
public enum Color{
 RED,
GREEN,
BLUE,
WHITE,
BLACK
}
public BorderColor(Shape shape,Color color){
this.shape=shape;
this.color=color;
}
   @Override
   public String getDescription() {
      return shape.getDescription()+"\nborder-color: "+color;
   }

   @Override
   public void draw() {
   shape.draw();
   System.out.println("border-color: "+color);}}
```

- **BorderThickness “Class” concrete decorator:**

```
public class BorderThickness extends mainDecorater {
   Shape shape;
   double thick;
   public BorderThickness(Shape shape,double thick){
      this.shape=shape;
      this.thick=thick;
   }

   @Override
   public String getDescription() {
      return shape.getDescription()+"\nborder-thickness: "+thick;
   }

   public void draw() {
   shape.draw();
      System.out.println("border-thickness: "+thick);
   }
}
```

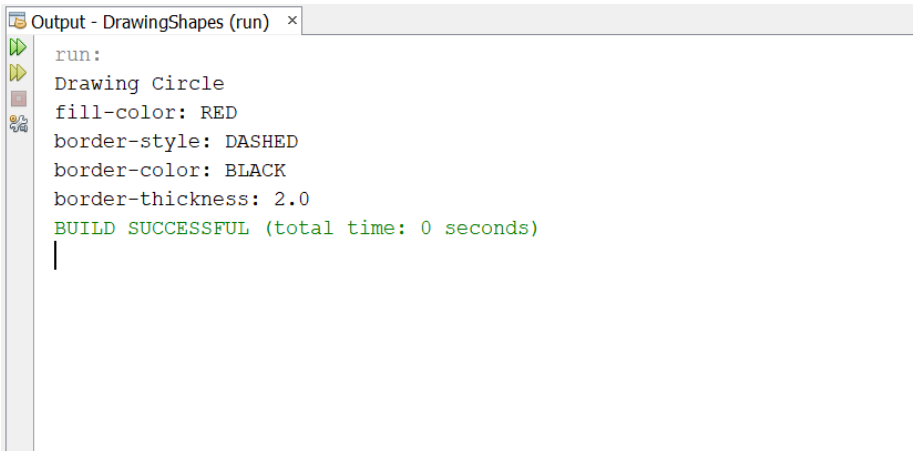
- **BorderStyle “Class” concrete decorator:**

```
public class BorderStyle extends mainDecorater {
   Shape shape;
   borderStyle bStyle;
 public enum borderStyle {
DASHED,
SOLID,
DOTTED,
WHITE,
BLACK
   }
   public BorderStyle(Shape shape,borderStyle bStyle){
   this.shape=shape;
   this.bStyle=bStyle;
   }
   @Override
   public String getDescription() {
   return shape.getDescription()+"\nborder-style: "+bStyle;
   }

   @Override
   public void draw() {
   shape.draw();
   System.out.println("border-style: "+bStyle);}}
```

Test Code:

```
public class Drawingtool {
    public static void main(String[] args) {
        Shape n=(Shape) new Circle();
        n=new FillColor(n, FillColor.Color.RED);
        n=new BorderStyle(n, BorderStyle.borderStyle.DASHED);
        n=new BorderColor(n, BorderColor.Color.BLACK);
        n=new BorderThickness(n,2.0);
        n.draw();
    }
}
```



- note: We can send the color or border style as String to the constructor and make check to the values using Switch then assign the suitable enum value as the code below:

```
public BorderStyle(Shape shape,String bStyle){
this.shape=shape;
switch (bStyle.toUpperCase()) {
            case "DASHED":
              bStyle=borderStyle.DASHED;
               break;
            case "SOLID":
               bStyle=borderStyle.SOLID;
               break;
            case "DOTTED":
               bStyle=borderStyle.DOTTED;
               break;
            case "BLACK":
              bStyle=borderStyle.BLACK;
               break;
            default:
```

```
            bStyle=borderStyle.WHITE;
            break;
        }
}
```

هبة سعيد حمدان

202010678