Adapter Pattern

```
<<Interface>>
IShapes

  +draw ()
```

```
<<Interface>>
IShapesColors

  +Colored ()
  +Border ()
```

```
FactoryMethods

  +getShapes ()
  +setShapesStyles()
```

```
Circle

  +draw()
```

```
Colored

  +Colored()
```

```
Square

  +draw ()
```

```
Border

  +Border()
```

```
Triangle

  +draw()
```

```
ShapesBuilder

  -Styles: String[]
  -Shapes: String[]
  +main ()
```

Code:

Interface:

```java
public interface IShapesColors {
   void Colored(String nameOfShape);
   void Border(String nameOfShape);
}
```

```java
public interface IShapes {
   void draw();
}
```

```java
public class FactoryShapes {

   public IShapes getShapes(String name){
    IShapes NumberOfShapes;
      if (name == "Circle"){
         NumberOfShapes = new Circle();
      }else if (name == "Triangle") {
         NumberOfShapes = new Triangle();
      }else if (name == "Square") {
         NumberOfShapes = new Square();
      }
      else {
         return null;
      }
      return NumberOfShapes;
   }

   public void setShapeStyle(String style, String Shape) {
      if (style == "Border") {
         (new BorderShapes()).Border(Shape);;
      } else if (style == "Colored") {
         (new ColoredShapes()).Colored(Shape);
      } else {

      }
   }
}

class Circle implements IShapes{
   @Override
   public void draw() {
      System.out.println("Circle is Draw");
   }
}
class Square implements IShapes{
   @Override
   public void draw() { System.out.println("Square is Draw"); }
}
```

```java
class Triangle implements IShapes{
    @Override
    public void draw() {
        System.out.println("Triangle is Draw");
    }
}
```

```java
public class BorderShapes implements IShapesColors {

    @Override
    public void Colored(String nameOfShape) {
//      do nothings
    }

    @Override
    public void Border(String nameOfShape) {
        System.out.println("Give border to the shape of "+nameOfShape);
    }
}
```

```java
public class ColoredShapes implements IShapesColors {

    @Override
    public void Colored(String nameOfShape) {
        System.out.println("Give Colored to the shape of "+nameOfShape);
    }

    @Override
    public void Border(String nameOfShape) {
        // do nothings
    }
}
```

```
Square is Draw
Give border to the shape of Square
Triangle is Draw
Give Colored to the shape of Triangle
Circle is Draw
Give border to the shape of Circle
```