Bridge

```
// Implementation
public interface Color {
	public void setBlueColor(Shape shape);
	public void setRedColor(Shape shape);
}
public class ColoredCircle implements Color {
	private Circle circle = new Circle("circle");
	@Override
	public void setBlueColor(Shape shape) {
		if(shape.getShapeName().equals(circle.getShapeName())) {
			shape.setShapeColor("blue");
		}
	}
	@Override
	public void setRedColor(Shape shape) {
		if(shape.getShapeName().equals(circle.getShapeName())) {
			shape.setShapeColor("red");
		}
	}
	@Override
	public boolean equals(Object obj) {
		if(this == obj) {
			return true;
		}
		if(obj == null || getClass() != obj.getClass()) {
			return false;
		}
		Circle otherCircle = (Circle) obj;
		return circle.getShapeName().equals(otherCircle.getShapeName());
	}
}
public class ColoredSquare implements Color {
	private Square square = new Square("square");
	@Override
	public void setBlueColor(Shape shape) {
		if(shape.getShapeName().equals(square.getShapeName())) {
			shape.setShapeColor("blue");
		}
	}
	@Override
	public void setRedColor(Shape shape) {
		if(shape.getShapeName().equals(square.getShapeName())) {
			shape.setShapeColor("red");
		}
	}
	@Override
	public boolean equals(Object obj) {
		if(this == obj) return true;
		if(obj == null || getClass() != obj.getClass()) return false;
		Square otherSquare = (Square) obj;
		return square.getShapeName().equals(otherSquare.getShapeName());
	}
}
```

```
public abstract class Shape {
	private String shapeName;
	private String shapeColor;
	protected Color color;
	public Shape(String name) {
		this.setShapeName(name);
		this.shapeColor = "none";
	}
	public String getShapeName() {
		return shapeName;
	}
	public void setShapeName(String shapeName) {
		this.shapeName = shapeName;
	}
	public String getShapeColor() {
		return shapeColor;
	}
	public void setShapeColor(String shapeColor) {
		this.shapeColor = shapeColor;
	}
	public abstract void draw();
}
public class Square extends Shape {
	public Square(String name) {
		super(name);
	}
	@Override
	public void draw() {
System.out.println("[DRAWING]: " + getShapeName() + " with " + getShapeColor() + " color");
	}
}
public class Circle extends Shape {
	public Circle(String name) {
		super(name);
	}
	@Override
	public void draw() {
System.out.println("[DRAWING]: " + getShapeName() + " with " + getShapeColor() + " color");
	}
}
public class Client {
	public static void main(String[] args) {
		new Client().go();
	}
	void go() {
		Shape circle = new Circle("circle");
		circle.draw();
		new ColoredCircle().setBlueColor(circle);
		circle.draw();
		new ColoredCircle().setRedColor(circle);
		circle.draw();
		Shape square = new Square("square");
		square.draw();
		new ColoredSquare().setBlueColor(square);
		square.draw();
	}
}
```