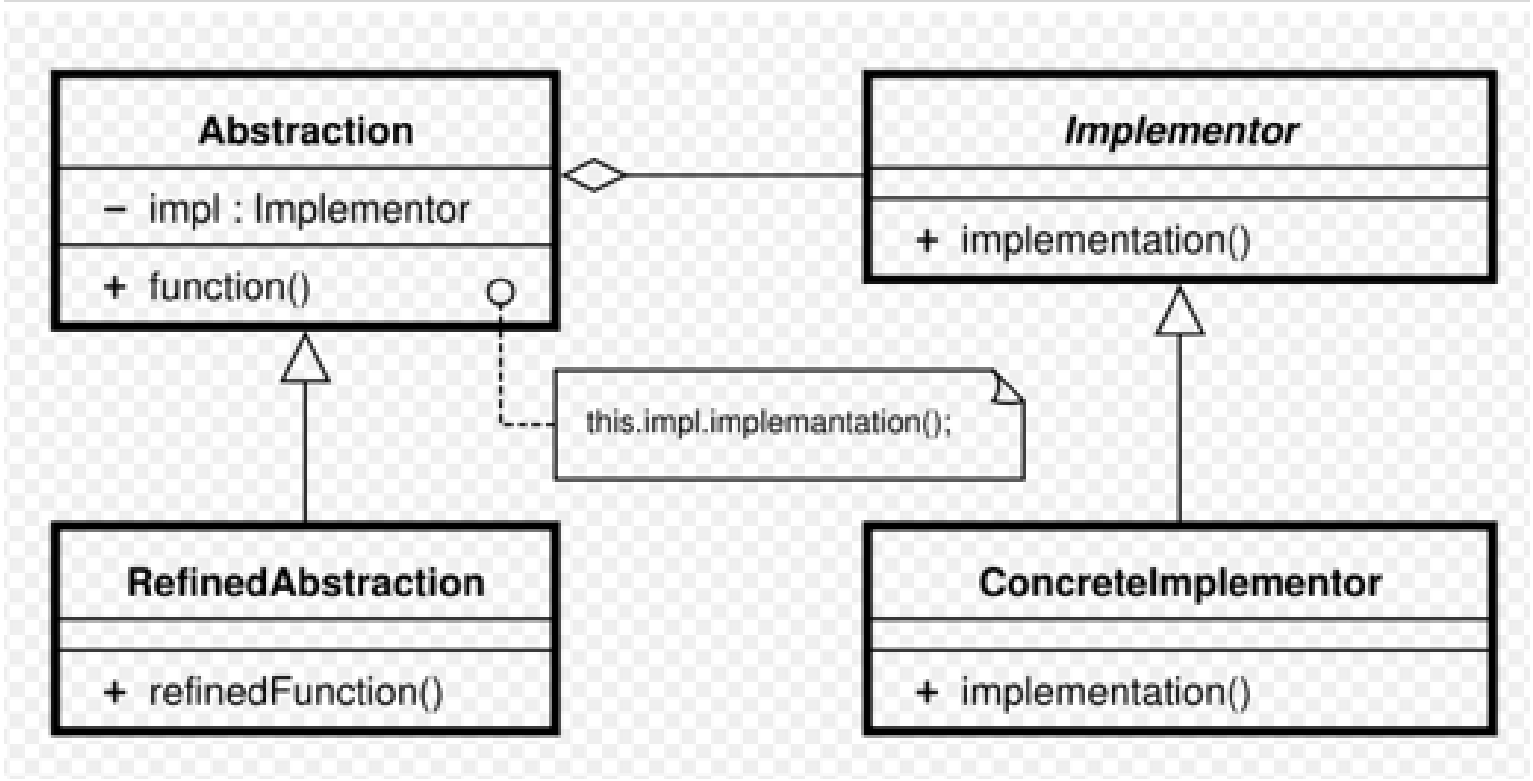
bridge pattern

Separation of Abstraction and Implementation

for abstraction and implementation


Abstraction
- impl : Implementor
+ function()
Implementor
+ implementation()
this.impl.implemantation();
RefinedAbstraction
+ refinedFunction()
ConcreteImplementor
+ implementation()

```
interface Shape {
    public void draw();
    public void resizeByPercentage(double pct);
}
```

```
class CircleShape implements Shape {
    private double x, y, radius;
    private DrawingAPI drawingAPI;
    public CircleShape(double x, double y, double radius,
                       DrawingAPI drawingAPI) {
        this.x = x;  this.y = y;  this.radius = radius;
        this.drawingAPI = drawingAPI;
    }
    public void draw() {
         drawingAPI.drawCircle(x, y, radius);
    }
    public void resizeByPercentage(double pct) {
         radius *= pct;
    }
}
```

```
interface DrawingAPI {
    public void drawCircle(double x, double y, double radius);
}
```

```
class DrawingAPI1 implements DrawingAPI {
   public void drawCircle(double x, double y, double radius) {
        System.out.printf("API1.circle at %f:%f
                           radius %f₩n", x, y, radius);
   }
}
```

```
class DrawingAPI2 implements DrawingAPI {
   public void drawCircle(double x, double y, double radius) {
        System.out.printf("API2.circle at %f:%f
              radius %f₩n", x, y, radius);
   }
}
```

```
class BridgePattern {
    public static void main(String[] args) {
        Shape[] shapes = new Shape[2];
        shapes[0] = new CircleShape(1, 2, 3, new DrawingAPI1());
        shapes[1] = new CircleShape(5, 7, 11, new DrawingAPI2());

        for (Shape shape : shapes) {
            shape.resizeByPercentage(2.5);
            shape.draw();
        }
    }
}
```

```
API1.circle at 1.000000:2.000000 radius 7.500000
API2.circle at 5.000000:7.000000 radius 27.500000
```

maintainability issue

```
class BridgePattern {
    public static void main(String[] args) {
        Shape[] shapes = new Shape[3];
        shapes[0] = new CircleShape(1, 2, 3, new DrawingAPI1());
        shapes[1] = new CircleShape(5, 7, 11, new DrawingAPI2());
        shapes[2] = new RectangleShape
                (5, 7, 8, 9, new DrawingRectangleAPI1());
        for (Shape shape : shapes) {
            shape.resizeByPercentage(2.5);
            shape.draw();
        }
    }
}
```