

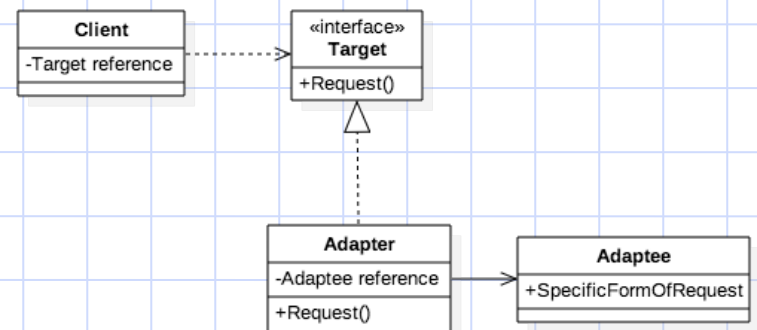
Adapter Example

Class Exercise

General Template

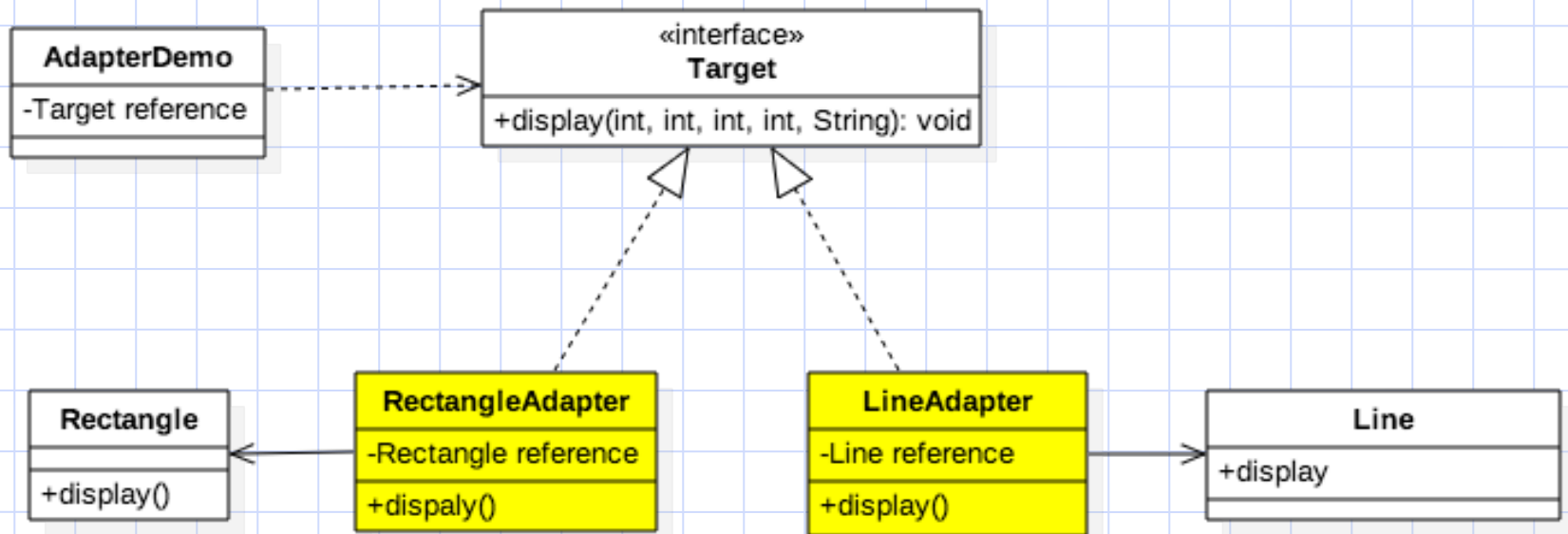
```
class Adaptee {  
    legacyMethod(...) {  
    }  
};  
  
interface Target {  
    clientMethod(...);  
}  
  
// a wrapper class  
class Adapter implements Target {  
    clientMethod(...) {  
        adapteeMethod(...)  
        // MORE  
    }  
}
```

```
class Client {  
    useAdapter() {  
        Target x = new Adapter();  
        x.clientMethod(...);  
    }  
};
```



Now Let's Learn More By An Example

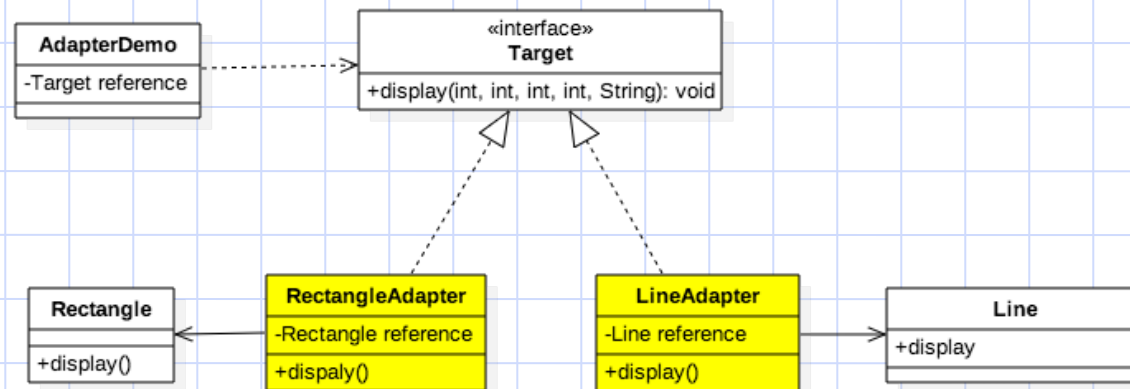
- Let's assume we would like to use a legacy code for a few geometric shapes (line, rectangle), and a client needs to use an adapter, as client's interface doesn't match with the legacy code.



Step 1: Creating Legacy Classes (Adaptees)

```
class Line {  
    public void display(int x1, int y1, int x2, int y2)  
    {  
        System.out.print("Coordintes of Line are: (" + x1 + "," +  
            + y1 + "), and (" + x2 + "," + y2 + ")");  
    }  
}
```

```
class Rectangle {  
    public void display(int x, int y, int width, int height) {  
        System.out.print("Coordinates of the Left-corner are (" + x + "," + y +  
            + "), width: " + width + ", height: " + height);  
    }  
}
```



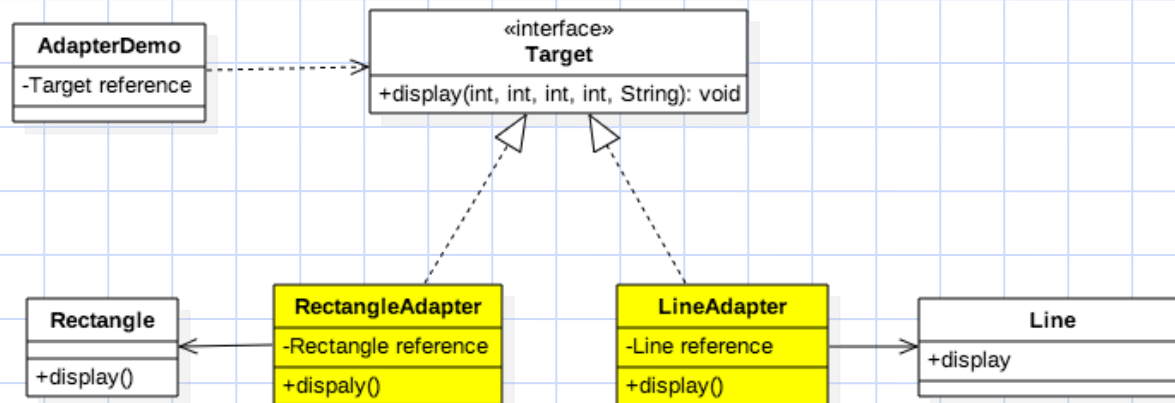
Step 2: Creating Target Interface

```
interface Target
```

```
{
```

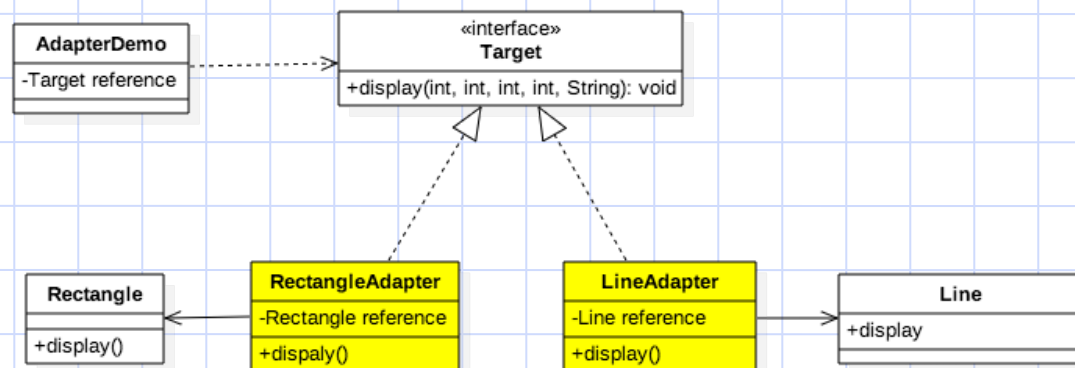
```
    void display(int x, int y, int z, int w, String color);
```

```
}
```



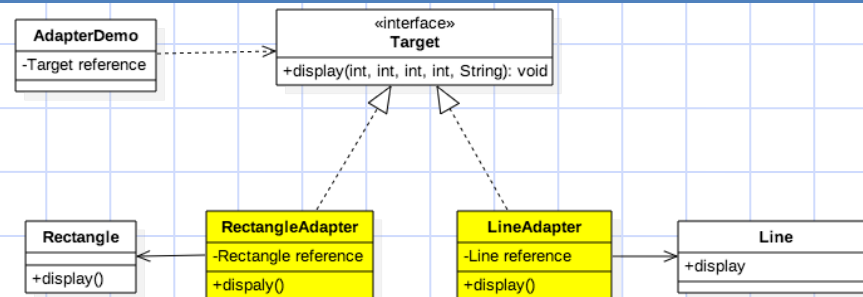
Step 3: Create An Adapter for class Line

```
class LineAdapter implements Target {  
    private Line adaptee;  
  
    public LineAdapter(Line line)  
    {  
        this.adaptee = line;  
    }  
  
    @Override  
    public void display(int x1, int y1, int x2, int y2, String color)  
    {  
        adaptee.display(x1, y1, x2, y2);  
        System.out.println(" and its Color is: " + color);  
    }  
}
```



Step 4: Create Another Adapter for class Rectangle

```
class RectangleAdapter implements Target {  
    private Rectangle adaptee;  
  
    public RectangleAdapter(Rectangle rectangle) {  
        this.adaptee = rectangle;  
    }  
  
    @Override  
    public void display(int x, int y, int z, int w, String color) {  
        adaptee.display(x, y, z, w);  
        System.out.println(" and its color is: " + color);  
    }  
}
```



Step 5: Lets See if it Works

```
public class AdapterDemo {  
    public static void main(String[] args)  
    {  
        Target[] shapes = {new RectangleAdapter(new Rectangle()),  
                           new LineAdapter(new Line())};  
  
        int x1 = 10, y1 = 20;  
        int x2 = 30, y2 = 60;  
        for (Target shape : shapes) {  
            shape.display(x1, y1, x2, y2, "Red");  
        }  
    }  
}
```

