# Java Programming II

# Lab8

514770-1
Fall 2025
11/19/2025

Kyoung Shin Park
Computer Engineering
Dankook University

# Lab8

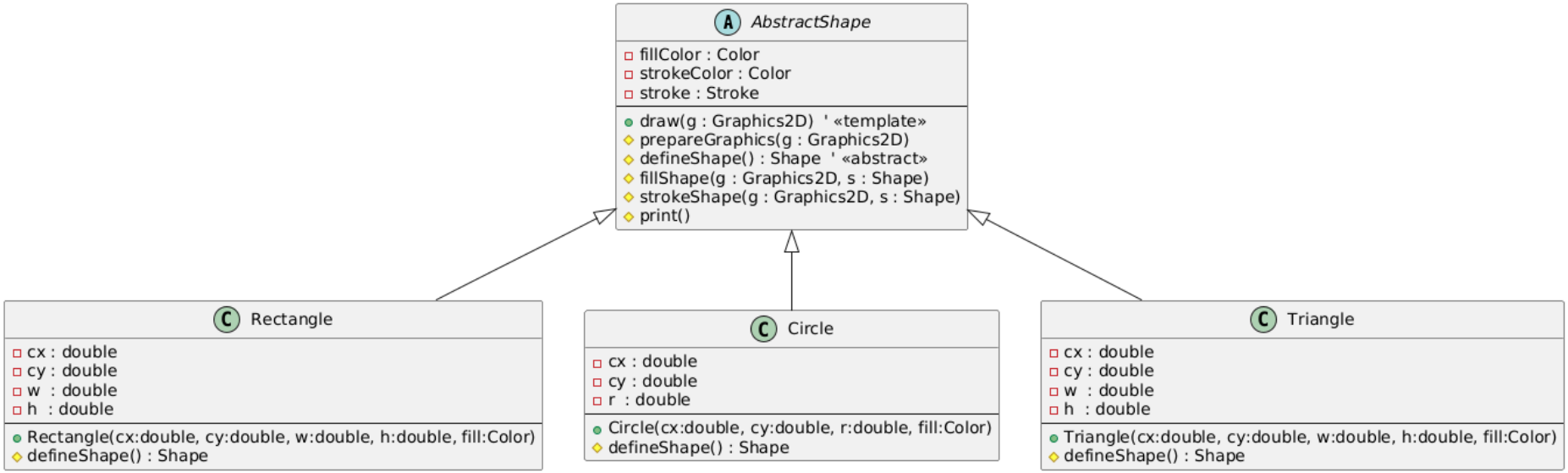
- Practice to write a program that **draw shape** using **Template Method pattern.**
  - **AbstractShape** class uses template method called **draw().**
  - **Rectangle, Circle, Triangle** class implements abstract method.
    - **protected abstract Shape defineShape() // define Rectangle2D/Ellipse2D/Path2D**

# Lab8

- **AbstractShape** class use the Template Method
  - public final void draw(Graphics2D g); // template method (not changed)
    - protected void prepareGraphics(Graphics2D g) { … }
    - protected abstract Shape defineShape(); // changeable
    - protected void fillShape(Graphics2D g, Shape s) { … }
    - protected void strokeShape(Graphics2D g, Shape s) { … }
    - protected void print() { … }

```
public abstract class AbstractShape {
…
    protected void prepareGraphics(Graphics2D g) {
        g.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
RenderingHints.VALUE_ANTIALIAS_ON);
    }
    protected abstract Shape defineShape(); // abstract method
    protected void fillShape(Graphics2D g, Shape s) {
        g.setColor(fillColor);
        g.fill(s);
    }
    protected void strokeShape(Graphics2D g, Shape s) {
        g.setColor(strokeColor);
        g.setStroke(stroke);
        g.draw(s);
    }
    protected void print() {
        System.out.println("[Shape] drawn: " + getClass().getSimpleName());
    }
}
```

# Lab8

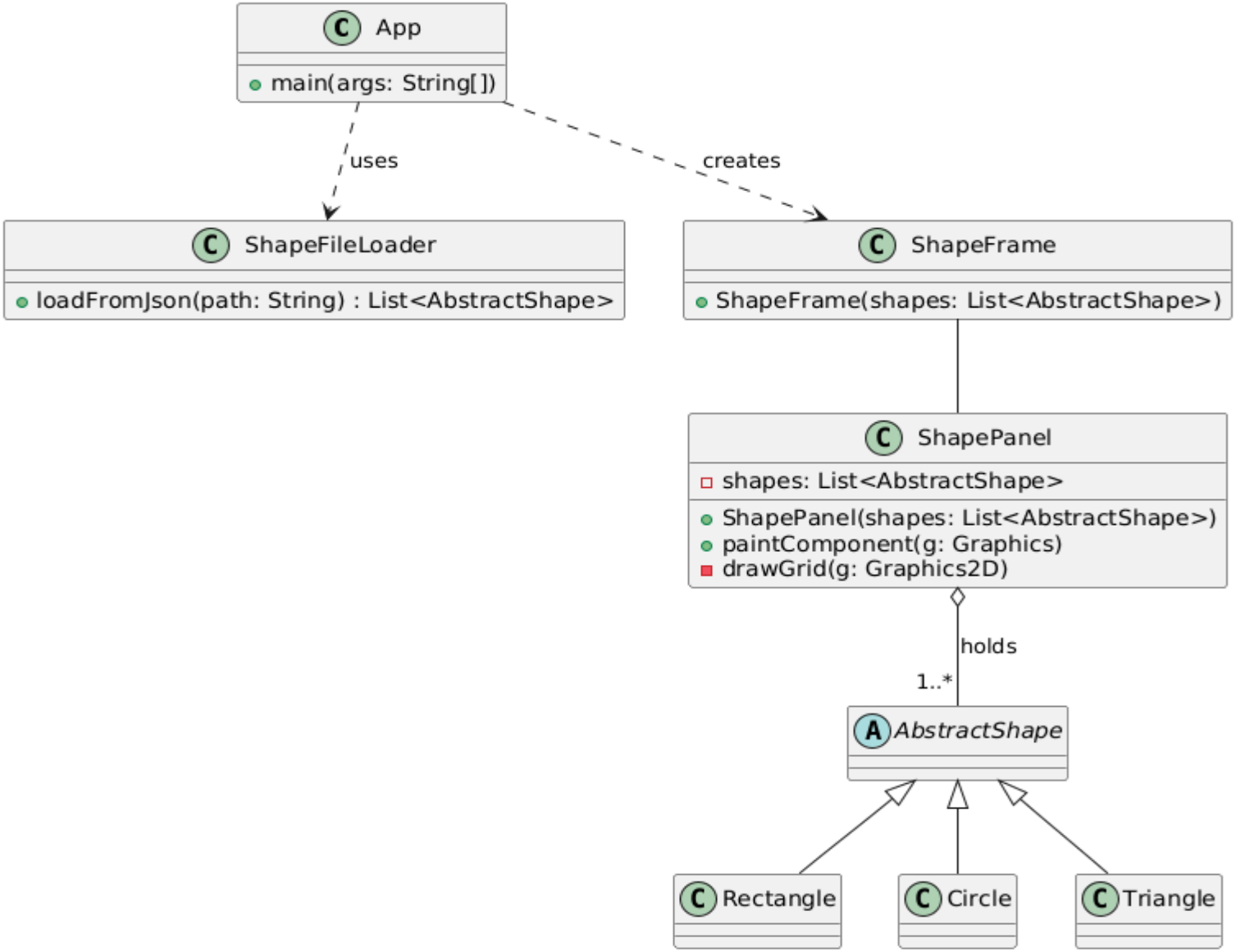

AbstractShape
- fillColor : Color
- strokeColor : Color
- stroke : Stroke
+ draw(g : Graphics2D) ' «template»
# prepareGraphics(g : Graphics2D)
# defineShape() : Shape ' «abstract»
# fillShape(g : Graphics2D, s : Shape)
# strokeShape(g : Graphics2D, s : Shape)
# print()
Rectangle
- cx : double
- cy : double
- w : double
- h : double
+ Rectangle(cx:double, cy:double, w:double, h:double, fill:Color)
# defineShape() : Shape
Circle
- cx : double
- cy : double
- r : double
+ Circle(cx:double, cy:double, r:double, fill:Color)
# defineShape() : Shape
Triangle
- cx : double
- cy : double
- w : double
- h : double
+ Triangle(cx:double, cy:double, w:double, h:double, fill:Color)
# defineShape() : Shape

# Lab8

- Rectangle, Circle, Triangle extends AbstractShape
  - Rectangle2D r = new Rectangle2D.Double(cx, cy, w, h);
  - Ellipse2D c = new Ellipse2D.Double(cx - r, cy - r, 2*r, 2*r);
  - Path2D p = new Path2D.Double();

```
p.moveTo(cx, cy - h/2);
p.lineTo(cx - w/2, cy + h/2);
p.lineTo(cx + w/2, xy + h/2);
p.closePath();
```

# Lab8


App
main(args: String[])
uses
creates
ShapeFileLoader
loadFromJson(path: String) : List<AbstractShape>
ShapeFrame
ShapeFrame(shapes: List<AbstractShape>)
ShapePanel
shapes: List<AbstractShape>
ShapePanel(shapes: List<AbstractShape>)
paintComponent(g: Graphics)
drawGrid(g: Graphics2D)
holds
1..*
AbstractShape
Rectangle
Circle
Triangle

```
public class App {
    public static void main(String[] args) throws Exception {
        List<AbstractShape> shapes = ShapeFileLoader.loadFromJson("shapes.json");
        SwingUtilities.invokeLater(() -> {
            new ShapeFrame(shapes);
        });
    }
}
public class ShapeFrame extends JFrame {
    public ShapeFrame(List<AbstractShape> shapes) {
        super("Shape with Template Method Pattern (Swing)");
        var shapePanel = new ShapePanel(shapes); // shapePanel
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(800, 600);
        setLocationRelativeTo(null);
        setContentPane(shapePanel);
        setResizable(false); // fixed size
        setVisible(true);
    }
}
```

# Submit to e-learning

- Add your code (e.g., additional website, additional Design pattern, etc) in the Lab8 assignment.
- Submit the Lab8 assignment (JAVA25-2-Lab8-YourID-YourName.zip including the report) to e-learning due by 11/25.