Java Programming II

Lab9

514770-1
Fall 2025
11/26/2025

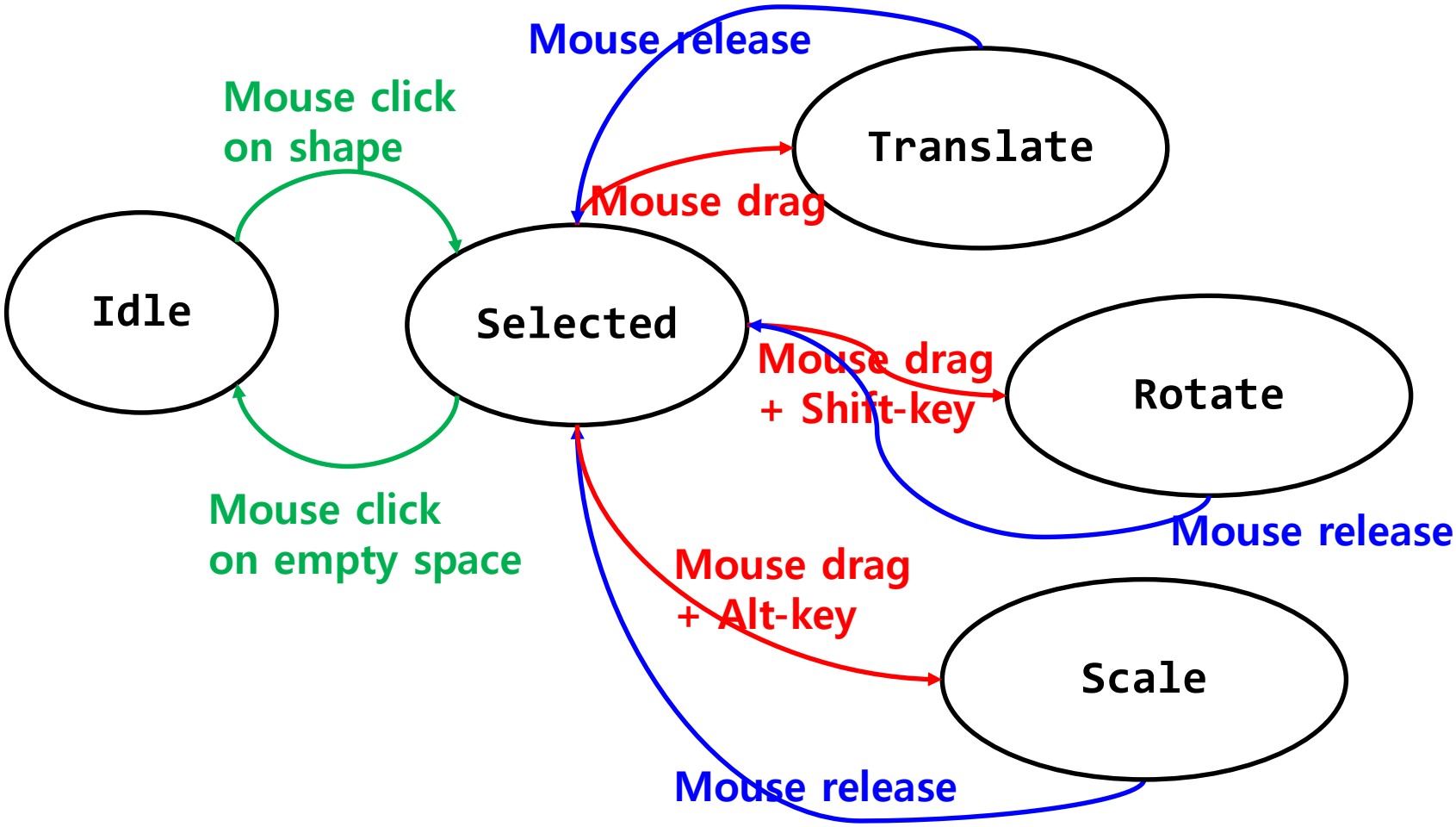
Kyoung Shin Park
Computer Engineering
Dankook University

Lab9

- Practice to write a program that **design a State pattern** in which the state of a shape dynamically transitions based on **user input (click, drag, release).**
  - **IShapeControlState interface**
  - **IdleState, SelectedState, TranslateState, RotateState, ScaleState**

Lab9

- **IShapeControlState** Finite State Machine(FSM)



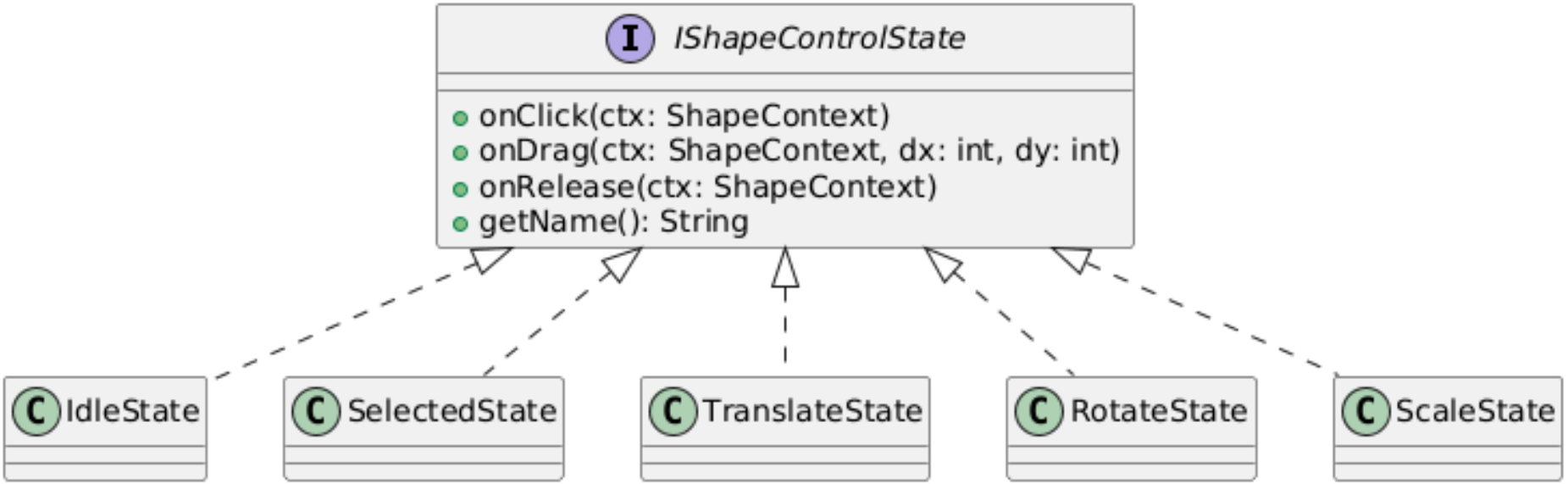
Lab9

- **IShapeControlState interface**
  - **void onClick(ShapeContext ctx)**
  - **void onDrag(ShapeContext ctx, int dx, int dy)**
  - **void onRelease(ShapeContext ctx);**
  - **default String getName() { return getClass().getSimpleName(); }**
- **IdleState class**
  - IdleState transits to **SelectedState** if **onClick() shape**
- **SelectedState class**
  - SelectedSate transits to **IdleState** if **onClick() empty space**
  - SelectedSate transits to **TranslateState** if **onDrag() shape**

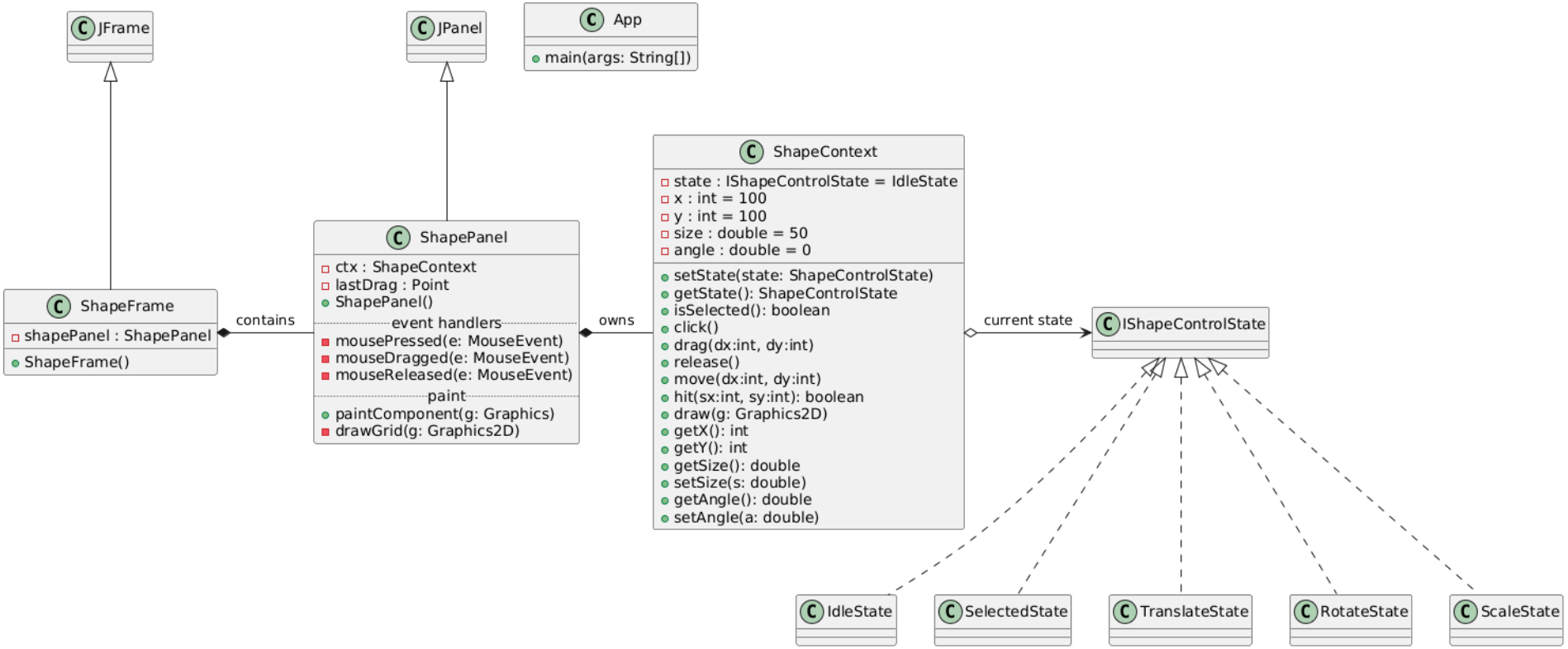
Lab9

- **TranslateState**
  - **onDrag() – move(dx, dy)**
  - TranslateSate transits to **SelectedState** if **onRelease()**
- **RotateState**
  - **onDrag() – setAngle(ctx.getAngle() + dx * 0.5)**
  - RotateSate transits to **SelectedState** if **onRelease()**
- **ScaleState**
  - **onDrag() – setSize(ctx.getSize() + (dx + dy) * 0.5)**
  - ScaleSate transits to **SelectedState** if **onRelease()**

Lab9


IShapeControlState
onClick(ctx: ShapeContext)
onDrag(ctx: ShapeContext, dx: int, dy: int)
onRelease(ctx: ShapeContext)
getName(): String
IdleState
SelectedState
TranslateState
RotateState
ScaleState

Lab9


JFrame
JPanel
App
main(args: String[])
ShapeContext
state : IShapeControlState = IdleState
x : int = 100
y : int = 100
size : double = 50
angle : double = 0
setState(state: ShapeControlState)
getState(): ShapeControlState
isSelected(): boolean
click()
drag(dx:int, dy:int)
release()
move(dx:int, dy:int)
hit(sx:int, sy:int): boolean
draw(g: Graphics2D)
getX(): int
getY(): int
getSize(): double
setSize(s: double)
getAngle(): double
setAngle(a: double)
ShapePanel
ctx : ShapeContext
lastDrag : Point
ShapePanel()
event handlers
mousePressed(e: MouseEvent)
mouseDragged(e: MouseEvent)
mouseReleased(e: MouseEvent)
paint
paintComponent(g: Graphics)
drawGrid(g: Graphics2D)
ShapeFrame
shapePanel : ShapePanel
ShapeFrame()
contains
owns
current state
IShapeControlState
IdleState
SelectedState
TranslateState
RotateState
ScaleState

Lab9

```
public class ShapeContext {
    private int x = 100, y = 100;
    private double size = 50;
    private double angle = 0;
    private IShapeControlState state = new IdleState();
// getAngle, getSize getter/setter …
    public boolean isSelected() {
        return (state instanceof SelectedState);
    }
    public void click() {   state.onClick(this);     }
    public void drag(int dx, int dy) {
        state.onDrag(this, dx, dy);
    }
    public void release() {  state.onRelease(this);     }
    public void move(int dx, int dy) {
        x += dx; y += dy;
    }
```

```
    public boolean hit(int sx, int sy) {
        AffineTransform inv = new AffineTransform();
        inv.rotate(-Math.toRadians(angle), x, y);
        Point p = new Point(sx, sy);
        inv.transform(p, p);
        int half = (int)(size / 2);
        return (p.x >= x - half && p.x <= x + half && p.y >= y
- half && p.y <= y + half);
    }
    public void draw(Graphics2D g) {
        AffineTransform save = g.getTransform();
        g.rotate(Math.toRadians(angle), x, y);
        int half = (int)(size/2);
        Rectangle2D rect = new Rectangle2D.Double(x - half, y -
half, size, size);
        g.setColor(new Color(90, 180, 255));
        g.fill(rect);
        g.setColor(new Color(30, 90, 180));
        g.setStroke(new BasicStroke(2f));
        g.draw(rect);
        g.setTransform(save);
    }   }
```

Lab9

```
public class ShapePanel extends JPanel {
    private final ShapeContext ctx = new ShapeContext();
    private Point lastDrag;

    public ShapePanel() {
        setBackground(Color.white);
        setFocusable(true);
        setDoubleBuffered(true);

        addMouseListener(new MouseAdapter() {
            @Override // 마우스 클릭
            public void mousePressed(MouseEvent e) {
            …}
            @Override // 마우스 릴리즈
            public void mouseReleased(MouseEvent e) {
            …}
        }
```

Lab9

```
        addMouseMotionListener(new MouseMotionAdapter() {
                @Override // 마우스 드래깅
                public void mouseDragged(MouseEvent e) {
                    if (lastDrag == null) return;
                    int dx = e.getX() - lastDrag.x;
                    int dy = e.getY() - lastDrag.y;

            }
        }
        @Override
        protected void paintComponent(Graphics g0) {
            // draw grid
            // draw shape
            // draw HUD
        }
}
```

Submit to e-learning

- Add your code (e.g., other class or design pattern, etc) in the Lab9 assignment.
- Submit the Lab9 assignment (JAVA25-2-Lab9-YourID-YourName.zip including the report) to e-learning due by 12/2.