



Tarea: Práctica 5

Pizarrón

Instituto Politécnico Nacional

Unidad Profesional Interdisciplinaria

Alumno: Orta Domínguez Tristán Eduardo

Materia: Programación Orientada a
Objetos

Profesor: Montes Casiano Hermes
Francisco

Desarrollo:

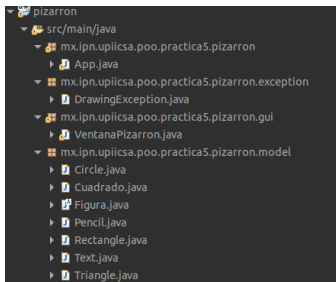
Se comenzó con un proceso similar al realizado en la calculadora, instanciando el panel y los respectivos botones comenzando por el círculo y terminado con lápiz.

```
private void initComponents() {  
    setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);  
    setResizable(true);  
    setExtendedState(JFrame.MAXIMIZED_BOTH);  
  
    instantiateComponents();  
    buildLayout();  
    initializeListener();  
  
    setVisible(true);  
}
```

```
public class Pizarron extends JFrame {  
    private static final int TOOL_UNSELECT = -1;  
    private static final int TOOL_CIRCLE = 1;  
    private static final int TOOL_TRIANGLE = 2;  
    private static final int TOOL_SQUARE = 3;  
    private static final int TOOL_RECTANGLE = 4;  
    private static final int TOOL_TEXT = 5;  
    private static final int TOOL_PENCIL = 6;  
  
    private static final boolean DRAWING_ACTIVE = true;  
    private static final boolean DRAWING_INACTIVE = false;  
  
    private JPanel toolPanel;  
    private JPanel dashboardPanel;  
    private JPanel btnPanel;  
  
    private JButton btnCircle;  
    private JButton btnTriangle;  
    private JButton btnSquare;  
    private JButton btnRectangle;  
    private JButton btnText;  
    private JButton btnPencil;  
  
    private int selectedTool;  
    private int drawingState;  
  
    private Figura pencilTemp;  
}
```

```
private Figura getFigureFrom(int x, int y) throws DrawingException {  
    Figura figura = null;  
    if (selectedTool == TOOL_UNSELECT) {  
        throw new DrawingException();  
    } else if (selectedTool == TOOL_CIRCLE) {  
        figura = Circle.getDefault(x, y);  
    } else if (selectedTool == TOOL_PENCIL) {  
        figura = Pencil.getDefault(x, y);  
    } else if (selectedTool == TOOL_TRIANGLE) {  
        figura = mx.ipn.uplicsa.poo.practica5.pizarron.model.Rectangle.getDefault(x, y);  
    } else if (selectedTool == TOOL_SQUARE) {  
        figura = Circle.getDefault(x, y);  
    } else if (selectedTool == TOOL_RECTANGLE) {  
        figura = Text.getDefault(x, y);  
    } else if (selectedTool == TOOL_TEXT) {  
        figura = Text.getDefault(x, y);  
    } else if (selectedTool == TOOL_TRIANGLE) {  
        figura = Triangle.getDefault(x, y);  
    }  
    return figura;  
}
```

Se crea el método pizarrón y se inicializan los diferentes métodos del mouse, ventana figuras de los botones y por último el color del fondo de los botones.



```
private void initializeListener() {  
    btnCircle.addActionListener(new ActionListener() {  
        @Override  
        public void actionPerformed(ActionEvent e) {  
            selectedTool = TOOL_CIRCLE;  
        }  
    });  
  
    btnTriangle.addActionListener(new ActionListener() {  
        @Override  
        public void actionPerformed(ActionEvent e) {  
            selectedTool = TOOL_TRIANGLE;  
        }  
    });  
  
    btnSquare.addActionListener(new ActionListener() {  
        @Override  
        public void actionPerformed(ActionEvent e) {  
            selectedTool = TOOL_SQUARE;  
        }  
    });  
}
```

Se crea una clase para los diferentes tipo de figuras que se desean realizar y cada una con un método diferente con ayuda de Graphics 2D en java.

```
package mx.ipn.uplicsa.poo.practica5.pizarron.model;  
import java.awt.Color;  
  
public class Rectangle extends Figura {  
    private static final int DEFAULT_WIDTH = 150;  
    private static final int DEFAULT_HEIGHT = 100;  
    private static final Color DEFAULT_BORDER_COLOR = Color.PINK;  
    private static final Color DEFAULT_FILL_COLOR = Color.BLACK;  
    private int width;  
    private int height;  
  
    protected Rectangle(int x, int y) {  
        super(x, y);  
        width = DEFAULT_WIDTH;  
        height = DEFAULT_HEIGHT;  
        fillColor = DEFAULT_FILL_COLOR;  
        borderColor = DEFAULT_BORDER_COLOR;  
    }  
  
    @Override  
    public void paint(Graphics g) {  
        Graphics2D g2d = (Graphics2D) g;  
        g2d.setColor(this.borderColor);  
        g2d.drawRect(this.x, this.y, this.width, this.height);  
        g2d.setColor(this.fillColor);  
        g2d.fillRect(this.x, this.y, this.width, this.height);  
    }  
  
    public static Rectangle getDefault(int x, int y) {  
        return new Rectangle(x, y);  
    }  
  
    public int getWidth() {  
        return width;  
    }  
  
    public void setWidth(int width) {  
        this.width = width;  
    }  
}
```

```
package mx.ipn.uplicsa.poo.practica5.pizarron.model;  
import java.awt.Color;  
  
public class Triangle extends Figura {  
    private static final Color DEFAULT_BORDER_COLOR = new Color(0, 0, 0);  
    private static final Color DEFAULT_FILL_COLOR = Color.RED;  
  
    protected Triangle(int x, int y) {  
        super(x, y);  
        this.borderColor = DEFAULT_BORDER_COLOR;  
        this.fillColor = DEFAULT_FILL_COLOR;  
    }  
  
    @Override  
    public void paint(Graphics g) {  
        int xcateto[] = { x + 50, x + 100, x + 1 };  
        int ycateto[] = { y + 1, y + 100, y + 100 };  
        Graphics2D g2d = (Graphics2D) g;  
        g2d.setColor(this.borderColor);  
        g2d.drawPolygon(xcateto, ycateto, 3);  
        g2d.setColor(this.fillColor);  
        g2d.fillPolygon(xcateto, ycateto, 3);  
    }  
  
    public static Triangle getDefault(int x, int y) {  
        return new Triangle(x, y);  
    }  
}
```

```
public class Pencil extends Figura {  
    private static final int DEFAULT_WIDTH = 1;  
    private static final Color DEFAULT_BORDER_COLOR = Color.BLACK;  
    private static final Color DEFAULT_FILL_COLOR = Color.BLACK;  
    private java.util.List<Point> points;  
  
    protected Pencil(int x, int y) {  
        super(x, y);  
        borderColor = DEFAULT_BORDER_COLOR;  
        fillColor = DEFAULT_FILL_COLOR;  
        stroke = DEFAULT_WIDTH;  
        points = new ArrayList<Point>();  
        points.add(new Point(x, y));  
    }  
  
    @Override  
    public void paint(Graphics g) {  
        if (points.size() > 0) {  
            Graphics2D g2d = (Graphics2D) g;  
            g2d.setColor(borderColor);  
  
            Point a = points.get(0);  
            for (int i = 1; i < points.size(); i++) {  
                Point b = points.get(i);  
                g2d.drawLine((int) a.getX(), (int) a.getY(), (int) b.getX(), (int) b.getY());  
                a = b;  
            }  
        }  
    }  
  
    public static Pencil getDefault(int x, int y) {  
        return new Pencil(x, y);  
    }  
  
    public void addPoint(int x, int y) {  
        points.add(new Point(x, y));  
    }  
  
    public java.util.List<Point> getPoints() {  
        return points;  
    }  
}
```