



জাভা দিয়ে অ্যাডভান্সড প্রোগ্রামিং

মো: আবদুল কাদের

এ পর্বে জাভা দিয়ে অ্যাডভান্সড প্রোগ্রামিং কোড দেখানো হয়েছে। জাভা ল্যান্সুয়েজ দিয়ে গ্রাফ পেপারের মতো স্থানে বৃত্তচাপ তৈরির পদ্ধতি দেখানো হয়েছে। এটি বাস্তব ক্ষেত্রে বিভিন্ন তুলনামূলক চিত্র উপস্থাপনের সময় প্রয়োগ করা যেতে পারে।

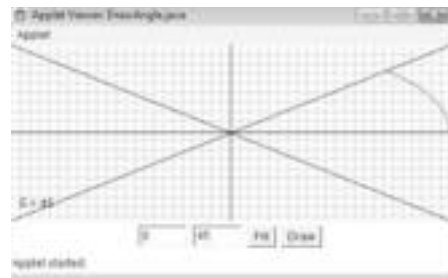
```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
/*<applet code="DrawAngle.java" width=500 height=200>
</applet>*/
public class DrawAngle extends Applet {
    ArcControls controls;
    ArcCanvas canvas;
    public void init() {
        setLayout(new BorderLayout());
        canvas = new ArcCanvas();
        add("Center", canvas);
        add("South", controls = new ArcControls(canvas));
    }
    public void destroy() {
        remove(controls);
        remove(canvas);
    }
    public void start() {
        controls.setEnabled(true);
    }
}
class ArcCanvas extends Canvas {
    int startAngle = 0;
    int endAngle = 45;
    Boolean filled = false;
    Font font;
    public void paint(Graphics g) {
        Rectangle r = getBounds();
        int hlines = r.height / 10;
        int vlines = r.width / 10;
        g.setColor(Color.pink);
        for (int i = 1; i <= hlines; i++) {
            g.drawLine(0, i * 10, r.width, i * 10);
        }
        for (int i = 1; i <= vlines; i++) {
            g.drawLine(i * 10, 0, i * 10, r.height);
        }

        g.setColor(Color.red);
        if (filled) {
            g.fillArc(0, 0, r.width - 1, r.height - 1, startAngle,
endAngle);
        } else {
            g.drawArc(0, 0, r.width - 1, r.height - 1, startAngle,
endAngle);
        }

        g.setColor(Color.black);
        g.setFont(font);
        g.drawLine(0, r.height / 2, r.width, r.height / 2);
        g.drawLine(r.width / 2, 0, r.width / 2, r.height);
        g.drawLine(0, 0, r.width, r.height);
        g.drawLine(r.width, 0, 0, r.height);
        int sx = 10;
        int sy = r.height - 28;
        g.drawString("S = " + startAngle, sx, sy);
        g.drawString("E = " + endAngle, sx, sy + 14);
    }
    public void redraw(boolean filled, int start, int end) {
        this.filled = filled;
        this.startAngle = start;
    }
}
```

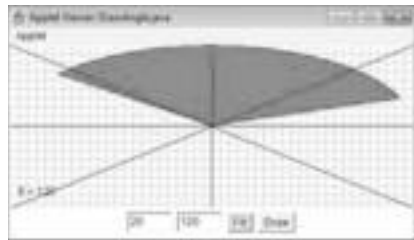
```
this.endAngle = end;
repaint();
}
}
class ArcControls extends Panel implements ActionListener {
    TextField s;
    TextField e;
    ArcCanvas canvas;
    public ArcControls(ArcCanvas canvas) {
        Button b = null;
        this.canvas = canvas;
        add(s = new TextField("0", 4));
        add(e = new TextField("45", 4));
        b = new Button("Fill");
        b.addActionListener(this);
        add(b);
        b = new Button("Draw");
        b.addActionListener(this);
        add(b);
    }
    public void actionPerformed(ActionEvent ev) {
        String label = ev.getActionCommand();
        canvas.redraw(label.equals("Fill"),
Integer.parseInt(s.getText().trim()),
Integer.parseInt(e.getText().trim()));
    }
}
```

উপরের প্রোগ্রামটি নোটপ্যাড টাইপ করে DrawAngle.java নামে সেভ করতে হবে এবং প্রোগ্রামটি কমান্ড প্রম্পটে নিচের চিত্রের মতো রান করতে হবে।



প্রোগ্রামটি রান করলে পাশের চিত্রের মতো দেখাবে।

প্রোগ্রামটিতে দুটি টেক্সট বক্স ব্যবহার করা হয়েছে। প্রথমটিতে যে সংখ্যা দেয়া



হবে, সেখান থেকে দ্বিতীয় টেক্সট বক্সের সংখ্যা অনুযায়ী বৃত্তচাপ তৈরি হবে, যা পাশে দেয়া চিত্রের মাধ্যমে দেখানো হলো। টেক্সট বক্সে সংখ্যা দেয়ার পর 'Draw' বাটনে ক্লিক

করলে শুধু রেখা দেখা যাবে এবং 'Fill' বাটনে ক্লিক করলে রং দিয়ে বৃত্তচাপ প্রদর্শিত হবে।

ফিডব্যাক : balaith@gmail.com