

জাভা দিয়ে গ্রাফিক্স ডিজাইন

মো: আবদুল কাদের

আগের দুই পর্বে আমরা জাভা দিয়ে চ্যাটিং রুমে চ্যাট করার কৌশল দেখিয়েছি। এখানে এক্সক্লুসিভলি একজনের সাথে যেমন চ্যাট করার পদ্ধতি দেখানো হয়েছে, তেমনি গ্রুপে চ্যাট করার পদ্ধতিও দেখানো হয়েছে। এ লেখায় জাভার আরও কিছু অ্যাডভান্সড ফিচার যেমন গ্রাফিক্সের কাজ দেখানো হয়েছে। আগের পর্বগুলোতে কমান্ড প্রম্পটভিত্তিক প্রোগ্রাম দেখানো হয়েছে। কিন্তু গ্রাফিক্সের কাজের জন্য দরকার উইন্ডো। তাই জাভা দিয়ে কোড লিখে কীভাবে উইন্ডোনির্ভর বিভিন্ন ধরনের ডিজাইন তৈরি করা যায়, এ পর্বে তা দেখানো হয়েছে।

এ পর্বে জাভা দিয়ে স্মৃতিসৌধ বানানোর প্রোগ্রাম দেখানো হয়েছে। প্রোগ্রামটি রান করার জন্য অবশ্যই আপনার কমপিউটারে Jdk সফটওয়্যার ইনস্টল থাকতে হবে। এখানে সফটওয়্যারটির Jdk1.4 ভার্সন ব্যবহার করা হয়েছে এবং প্রোগ্রামগুলো D:\ ড্রাইভের java ফোল্ডারে সেভ করা হয়েছে।

নিচের এই প্রোগ্রামটি নোটপ্যাডে টাইপ করে MemorialStand.java নামে সেভ করুন।

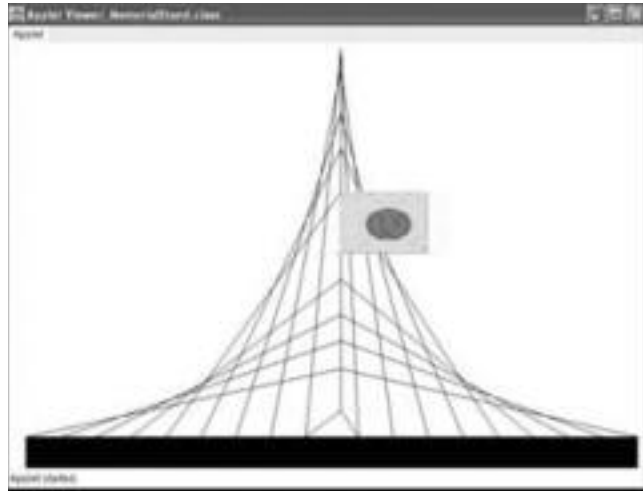
```
import java.awt.*;
import java.applet.Applet;
/*<applet code="MemorialStand.class"
width=300 height=300> </applet>*/
public class MemorialStand extends
Applet implements Runnable
{
    intx1[]={20,60,100,140,180,220,260,3
00,340,340}; //1
    inty2[]={372,340,310,270,170,120,80,
30,5, 420}; //2
    intx2[]={720,680,640,600,560,520,480
,440,400,400}; //3
    int k=0; //initialization
    public void init()
    {
        new Thread (this).start();
    }
    public void update (Graphics g)
    {
        g.fillRect(20,450,700,40);
        //Draw Memorial Stand
        for(k=0;k<=9;k++)
        {
            g.drawLine(x1[k],450,380,y2[k]);

```

```
g.drawLine(x2[k],450,380,y2[k]);
        }
        // draw flag
        g.drawLine (380,420,380,5);
        g.setColor(Color.green);
        g.fillRect(380,170,100,70);
        g.setColor(Color.red);
        g.fillOval(410,190,50,35);
    }
    public void run()
    {
        repaint();
    }
}
```



চিত্র - ১ : প্রোগ্রাম রান করার পদ্ধতি



চিত্র - ২ : প্রোগ্রাম রান করার পর আউটপুট

কোড বিশ্লেষণ

প্রোগ্রামটিতে <applet code> ব্যবহার করা হয়েছে উইন্ডো তৈরি করার জন্য, যার দৈর্ঘ্য ও প্রস্থ হবে ৩০০ ও ৩০০। ১, ২ ও ৩ চিহ্নিত লাইনে ইন্টিজার টাইপের তিনটি অ্যারে নেয়া হয়েছে। অ্যারে হলো অনেকগুলো ভেরিয়েবল রাখার জায়গা। এই অ্যারেগুলোতে ১০টি ভেরিয়েবল রাখা হয়েছে, যা প্রয়োজনমতো

প্রোগ্রামে ব্যবহার করা হবে। এরপর init() মেথড ব্যবহার করা হয়েছে, যার মাধ্যমে অ্যাপলেট চালু হবে। ফলে উইন্ডো ওপেন হবে। এরপর আপডেট মেথড ব্যবহার করা হয়েছে, যা মূলত রান মেথডের মধ্যে থেকে রান করবে। গ্রাফিক্স মেথডের মধ্যে গ্রাফিক্স সংক্রান্ত কোডগুলো লেখা হয়েছে। যেমন চতুর্ভুজ, সরলরেখা এবং বৃত্ত আঁকা হয়েছে। সবগুলোর সমন্বয়ে একটি স্মৃতিসৌধ নির্মিত হয়েছে।

রান করা

চিত্র-১-এর ১নং চিহ্নিত লাইনে Jdk-এর পাথ C:\ড্রাইভের Jdk ফোল্ডারের bin ফোল্ডারকে দেখানো হয়েছে। কারণ এই ফোল্ডারে জাভা রান করার সব প্রোগ্রাম রয়েছে। Jdk1.4 সফটওয়্যারটি

ইনস্টল করার পর ফোল্ডারের নাম যদি অনেক বড় হয় বা ভিন্ন হয়, তাহলে ফোল্ডারের নাম রিনেইম করে Jdk1.4 ব্যবহার করা যেতে পারে। এতে পাথ সেট করার সময় ভুল হওয়ার সম্ভাবনা কম থাকবে। জাভার কোনো প্রোগ্রাম রান করার জন্য পাথ সেটিং করতে হয়। এরপর D:\ ড্রাইভের java ফোল্ডারে চুকে ৪নং লাইন অনুযায়ী জাভা

ফাইলটিকে কম্পাইল করা হচ্ছে এবং ৫নং লাইন দিয়ে MemorialStand প্রোগ্রামটি রান হচ্ছে।

আমরা যদি স্মৃতিসৌধকে বিভিন্ন রঙে উপস্থাপন করতে চাই, তাহলে কোডের ভেতর initialization চিহ্নিত লাইনটি নিম্নের মতো করে পরিবর্তন করতে হবে।

```
int j=0, k=0,
red=0, green=0,
blue=0;
```

এবং Draw Memorial Stand-এর পর নিচের তিনটি লাইন

যুক্ত করতে হবে। এরপর সেভ করে আবার কম্পাইল করে রান করলে দেখা যাবে স্মৃতিসৌধটি ব্লিঙ্কিং করছে অর্থাৎ মনে হবে লাইটিং করা হয়েছে।

```
green=(int)(Math.random()*255.0);
blue=(int)(Math.random()*255.0);
g.setColor (new Color (red,green, blue));
```

ফিডব্যাক : balait@gmail.com