

# জাভা দিয়ে অ্যাপলেট প্রোগ্রামিং

মো: আবদুল কাদের

অ্যাপলেট নিয়ে প্রোগ্রামিংয়ের এ পর্বে অ্যাপলেট তৈরির আরও কিছু নতুন পদ্ধতি দেখানো হয়েছে। গত পর্বে সুইং প্যাকেজ দিয়ে অ্যাপলেট তৈরির পদ্ধতি এবং মেইন মেথড ব্যবহার করে রান করার পদ্ধতি দেখানো হয়েছে। এ পর্বে মূলত তিনটি প্রোগ্রাম দেখানো হয়েছে। একটি প্যানেলের ওপর অ্যাপলেট তৈরি এবং একটি অ্যাপলেটের ওপর আরেকটি অ্যাপলেট তৈরির দুই ধরনের দুটি প্রোগ্রাম নিয়ে আলোচনা করা হয়েছে।

## ১ম প্রোগ্রাম

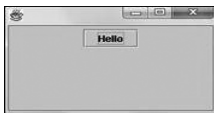
### (Applet\_On\_Panel.java)

এই প্রোগ্রামে প্যানেলের ওপর একটি অ্যাপলেট বসানো হবে। প্রথমে একটি প্যানেল (pan) নিয়ে তার ওপর বাটনবিশিষ্ট একটি অ্যাপলেট (japp) প্যানেলটিতে সংযুক্ত করা হবে। অ্যাপলেটে সংযুক্ত বাটনটি কোনো কাজ করবে না, কারণ বাটন কাজ করার জন্য ActionListener নামে যে ইন্টারফেস ইমপ্লিমেন্ট করতে হয়, তা এ প্রোগ্রামে করা হয়নি। এখানে শুধু কীভাবে প্যানেলে অ্যাপলেট সেট করা যায়, সে প্রক্রিয়াটুকু দেখানো হয়েছে। তাছাড়া এ প্রোগ্রামে মেইন মেথড ব্যবহার করা হয়েছে। তাই অন্যান্য জাভা প্রোগ্রাম রান করার মতোই এটি রান করতে হবে। এর জন্য appletviewer-এর প্রয়োজন নেই।

```
import javax.swing.*;
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
public class Applet_On_Panel extends JFrame
{
    Applet_On_Panel() {
        Container con = getContentPane();
        Panel pan = new Panel();
        JApplet japp = new JApplet();
        japp.setBackground(Color.green);
        JButton b1 = new JButton("Hello");
        (japp.getContentPane()).add(b1);
        pan.add(japp);
        con.add(pan);
    }
    public static void main(String args[]) {
        Applet_On_Panel x = new Applet_On_Panel();
        x.setSize(200,200);
        x.setVisible(true);
    }
}
```

```
C:\Users\CI-Nazmul\path=C:\jdk1.4\bin
C:\Users\CI-Nazmul>F:
F:\>cd java
F:\Java>javac Applet_On_Panel.java
F:\Java>java Applet_On_Panel
```

চিত্র : রান করার পদ্ধতি



চিত্র : আউটপুট

## ২য় প্রোগ্রাম

### (Applet\_Over\_Applet.java)

এই প্রোগ্রামে একটি অ্যাপলেটের ওপর আরেকটি অ্যাপলেট বসানো হবে। প্রথমে বাটনবিশিষ্ট একটি অ্যাপলেট (app) তৈরি করে তার ব্যাকগ্রাউন্ড সাদা সেট করা হয়েছে। ২য় অ্যাপলেটে একটি টেক্সট সংযুক্ত করা হয়েছে এবং অ্যাপলেটের ব্যাকগ্রাউন্ড দেয়া হয়েছে সবুজ রঙের, যাতে দুটি অ্যাপলেটের মধ্যে পার্থক্য সহজেই বোঝা যায়। এরপর ১ম অ্যাপলেটটিকে ২য় অ্যাপলেটে বসানো হয়েছে। এখানে ২য় অ্যাপলেটের সাইজ Dimension (Width, Height) দিয়ে নির্দিষ্ট করা হয়েছে। এ প্রোগ্রামে মেইন মেথড ব্যবহার করা হয়নি। তাই এটি রান করার জন্য appletviewer ব্যবহার করতে হবে। সবশেষে অ্যাপলেট বন্ধ করে দিলে Applet Stopped মেসেজ দেখাবে এবং মেমরি থেকে মুছে দেয়ার পর Destroyed মেসেজ দেখাবে।

```
import java.awt.*;
import java.applet.*;
import java.awt.Color.*;
/*<applet code = "Applet_Over_Applet.class"
name = "Applet_Over_Applet" width = 100 height = 100 ></applet>*/
public class Applet_Over_Applet extends
Applet {
    Dimension d = new Dimension(200,150);
    public void init() {
        try {
            Applet app = new Applet();
            app.setSize(200,100);
            app.setBackground(Color.white);
            Button b1 = new Button("Hello");
            app.add(b1);
            b1.setLocation(85, 85);
            add(app);
            setSize(200, 300);
            setBackground(Color.green);
            showStatus("Applet Initiated.");
            repaint();
        }
        catch (ClassCastException e) { }
    }
    public void paint(Graphics g) {
        resize(d);
        g.drawString ("String to be drawn on the
applet",20,100);
    }
    public void stop() {
        System.out.println("Applet Stopped.");
    }
    public void destroy() {
        System.out.println("Destroyed.");
    }
}
```

```
F:\Java>javac Applet_Over_Applet.java
F:\Java>appletviewer Applet_Over_Applet.java
```

চিত্র : রান করার পদ্ধতি



চিত্র : আউটপুট

## Applet\_On\_Applet.java

এই প্রোগ্রামেও একটি অ্যাপলেটের ওপর আরেকটি অ্যাপলেট বসানো হবে। তবে আগের প্রোগ্রামটি থেকে এর পার্থক্য হচ্ছে, এখানে যে অ্যাপলেটটি বসানো হবে, তা একটি কনটেন্টের মতো, অর্থাৎ বাটন বা টেক্সট বক্সের মতো কাজ করবে। প্রথমে বাটনবিশিষ্ট একটি অ্যাপলেট (app) তৈরি করে তার ব্যাকগ্রাউন্ড সবুজ সেট করা হয়েছে। এরপর ২য় অ্যাপলেট (japp) তৈরি করা হয়েছে। তারপর ২য় অ্যাপলেটের মধ্যে ১ম অ্যাপলেটকে সংযুক্ত করে একটি কন্টেইনারের (con) মাধ্যমে ফ্রেমে সংযুক্ত করা হয়েছে। এ প্রোগ্রামে মেইন মেথড ব্যবহার করা হয়েছে। প্রোগ্রামটি অন্যান্য জাভা প্রোগ্রামের মতো অথবা appletviewer-এর মাধ্যমে রান করা যাবে।

```
import javax.swing.*;
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
/* <applet code="Applet_On_Applet.class"
width=300 height=300> </applet> */
public class Applet_On_Applet extends JFrame
{
    Applet_On_Applet() {
        Container con = getContentPane();
        JApplet japp = new JApplet();
        Applet app = new Applet();
        app.setBackground(Color.green);
        JButton b1 = new JButton("Hello");
        app.add(b1);
        (japp.getContentPane()).add(app);
        //japp.setBackground(Color.white); //Try
        con.add(japp);
    }
    public static void main(String args[]) {
        Applet_On_Applet x = new
Applet_On_Applet();
        x.setSize(200,200);
        x.setVisible(true);
    }
}
```

```
F:\Java>javac Applet_Over_Applet.java
F:\Java>appletviewer Applet_Over_Applet.java
```

চিত্র : রান করার পদ্ধতি



চিত্র : আউটপুট

২য় ও ৩য় প্রোগ্রামের মধ্যে উল্লেখযোগ্য বৈশিষ্ট্য হলো ২য় প্রোগ্রামে দুটি ব্যাকগ্রাউন্ডই দেখা যাচ্ছে, কিন্তু ৩য় প্রোগ্রামে শুধু যে অ্যাপলেট সংযুক্ত করা হয়েছে, তার ব্যাকগ্রাউন্ড দেখা যাচ্ছে। অ্যাপলেট সংক্রান্ত প্রোগ্রামিংয়ের এটিই শেষ পর্ব। পরবর্তী পর্ব থেকে অন্য কোনো গুরুত্বপূর্ণ বিষয় নিয়ে আলোচনা করা হবে।

ফিডব্যাক : [balaith@gmail.com](mailto:balaith@gmail.com)