

গত পর্বে সুইং (Swing) প্রোগ্রাম দিয়ে GridBagLayout-এর ওপর একটি লেখাকে বিভিন্নভাবে উপস্থাপন করার প্রোগ্রাম দেখানো হয়েছিল। এ পর্বে সুইং দিয়ে ছবিযুক্ত বাটন তৈরি, বাটনটিকে অ্যাকটিভ করার পদ্ধতি এবং সেই সাথে বাটনটিতে ক্লিক করার পর প্রতিক্রিয়া নিয়ে একটি প্রোগ্রাম দেখানো হয়েছে। এ লেখায় বাটনগুলো প্রদর্শনের ক্ষেত্রে উইন্ডো হিসেবে অ্যাপলেট ব্যবহার করা হয়েছে।

সাধারণভাবে বাটন তৈরির জন্য জাভার abstract windowing toolkit (awt) প্যাকেজ প্রয়োজন হয়। তবে অ্যাডভান্সড ফিচার যেমন ছবিযুক্ত বাটন, দৃষ্টিনন্দন ইন্টারফেসের জন্য সুইং প্যাকেজ প্রয়োজন। এ লেখায় প্রোগ্রামটিতে তিনটি বাটন তৈরি করা হয়েছে, যার প্রত্যেকটিতে ছবি সংযুক্ত থাকবে। এজন্য আমাদের তিনটি ছবি প্রয়োজন হবে। gif ফরম্যাটের তিনটি ছবি D:\java ফোল্ডারে রাখা হয়েছে, যাতে রান করার সময় প্রোগ্রামটি ছবিগুলোকে খুঁজে পায়। এছাড়া এ প্রোগ্রামে বাটন ইভেন্ট নিয়ে কাজ করা হবে। উইন্ডোতে প্রদর্শিত বাটনগুলোতে ক্লিক করলে কী হবে, তা বাটন ইভেন্টে লিপিবদ্ধ থাকবে। সেই সাথে একটি টেক্সট বক্স নেয়া হবে, যেখানে বাটনে ক্লিক করার সাথে সাথে কোন বাটনে ক্লিক করা হচ্ছে সেই বাটন সংক্রান্ত মেসেজ প্রদর্শন করে।

প্রোগ্রামটি রান করার পদ্ধতি অন্যান্য জাভা প্রোগ্রামের মতোই। এজন্য অবশ্যই আপনার কমপিউটারে Jdk সফটওয়্যার ইনস্টল থাকতে হবে। সফটওয়্যারটির Jdk1.4 ভার্সন ব্যবহার করা হয়েছে এবং প্রোগ্রামগুলো D:\ড্রাইভের java ফোল্ডারে সেভ করা হয়েছে।

নিচের এই প্রোগ্রামটি নোটপ্যাডে টাইপ করে picturebutton.java নামে সেভ করতে হবে।

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
/*
<applet code="picturebutton" width=600 height =180>
</applet>
*/
public class picturebutton extends JApplet implements
ActionListener
{
    JTextField showText;
    public void init()
    {
        Container C = getContentPane();
        C.setLayout(new FlowLayout());ImageIcon sayNo = new
ImageIcon("no.gif"); //1
        JButton jb = new JButton(sayNo);
        jb.setActionCommand("No");
        jb.addActionListener(this);
        C.add(jb);
        ImageIcon astonish = new ImageIcon("astonish.gif"); //2
        jb = new JButton(astonish);
        jb.setActionCommand("Astonish");
        jb.addActionListener(this);
        C.add(jb);
        ImageIcon good = new ImageIcon("Good.gif"); //3
        jb = new JButton(good);
        jb.setActionCommand("Good");
        jb.addActionListener(this);
        C.add(jb);
        showText = new JTextField(15);
        C.add(showText);
    }
    public void actionPerformed(ActionEvent e)
    {
        showText.setText(e.getActionCommand());
    }
}
```

সুইং প্রোগ্রামে ছবিযুক্ত বাটন তৈরি

মো: আবদুল কাদের



কোড বিশ্লেষণ

প্রোগ্রামে প্রথম তিনটি লাইনের মাধ্যমে তিনটি প্যাকেজ ইম্পোর্ট করা হয়েছে। এখানে বাটন বা উইন্ডো সংক্রান্ত কাজ করার জন্য awt ইভেন্ট নিয়ে কাজ করার জন্য ইভেন্ট এবং সুইংয়ের গ্রাফিক্সসহ উইন্ডোর কম্পোনেন্ট ব্যবহার করার জন্য Swing প্যাকেজ নেয়া হয়েছে। কমান্ড লাইনের মাধ্যমে অ্যাপলেটকে রান করার জন্য কমেন্টের

(/* */) ভেতর অ্যাপলেটের জন্য ক্লাস এবং এর উইন্ডোর প্রস্থ ও দৈর্ঘ্য মান দিয়ে নির্দিষ্ট করা হয়েছে। এরপর প্রোগ্রামটিতে অ্যাপলেট তৈরির জন্য JApplet-কে extends করা হয়েছে। সেই সাথে বাটন ইভেন্ট নিয়ে কাজ করার জন্য ActionListener ইন্টারফেসকে ইমপ্লিমেন্ট করা হয়েছে।

অ্যাপলেটের init() মেথডে প্রথমেই একটি কন্টেইনার নেয়া হয়েছে। এই কন্টেইনারে পরবর্তী সময়ে একেকটি কম্পোনেন্টকে সংযুক্ত করা হবে এবং প্রয়োজন মতো কম্পোনেন্টগুলোকে সাজানো হবে। এ প্রোগ্রামে কম্পোনেন্টগুলোকে সাজানোর জন্য FlowLayout ব্যবহার করা হয়েছে। এই লেআউটের বিশেষত্ব হচ্ছে কন্টেইনারের ভেতরে সংযুক্ত করার পরে কম্পোনেন্টগুলো নিজেদের আকার অনুযায়ী একটির পর আরেকটি অবস্থান করে এবং প্রয়োজনীয় সংখ্যক রো ও কলাম তৈরি করে। এখানে GridBagLayout-এর মতো নির্দিষ্ট করে কোনো রো ও কলাম সংখ্যা বলে দিতে হয় না।

এরপর 1, 2 ও 3 নং চিহ্নিত লাইনে তিনটি ইমেজ আইকন sayNo, astonish এবং good তৈরি করা হয়েছে। এর মাধ্যমে তিনটি ছবিকে আইকনে রূপান্তরিত করা হয়েছে, যাকে পরবর্তী লাইনের মাধ্যমে বাটন হিসেবে কাজ করার জন্য বাটনের সাথে সংযুক্ত করা হয়েছে। বাটনগুলোতে ক্লিক করলে যে কাজ করবে তা ঠিক করে দেয়ার জন্য setActionCommand মেথড ব্যবহার করা হয়েছে। এই মেথডের মাধ্যমে আমরা মূলত একটি স্ট্রিং টাইপের লেখা বা একটি মেসেজ প্রদর্শন করব, যা পরবর্তী সময় একটি টেক্সট বক্সের মাধ্যমে দেখাবে। এরপর বাটনগুলোকে অ্যাকটিভ করার জন্য এর সাথে ইভেন্টের সংযোগ ঘটানো হয়েছে। ইমেজ আইকন থেকে শুরু করে বাটন অ্যাকটিভ করা পর্যন্ত সব কাজ সম্পন্ন করে প্রতিটি বাটনকে একটি একটি করে কন্টেইনারে সংযুক্ত করা হয়েছে। সবশেষে showText নামে একটি টেক্সটবক্স তৈরি করে তাকেও কন্টেইনারে সংযুক্ত করা হয়েছে।

এরপর বাটনে ক্লিক করলে actionPerformed মেথড ActionEvent অবজেক্টের সাহায্যে পূর্বে সেট করে দেয়া setActionCommand-এর মাধ্যমে দেয়া মেসেজ getActionCommand-এর মাধ্যমে সংগ্রহ করে setText মেথডের সাহায্যে টেক্সট বক্সে প্রদর্শন করবে।

প্রোগ্রাম রান করা

জাভার আগের প্রোগ্রামগুলোর মতো কমান্ড প্রম্পট ওপেন করে নিচের চিত্রের মতো করে রান করতে হবে



প্রোগ্রাম রান করার পদ্ধতি



প্রোগ্রাম রান করার পর আউটপুট

ফিডব্যাক : balaith@gmail.com