

যেকোনো প্রোগ্রামিং ল্যাঙ্গুয়েজে লেআউট ব্যবহার করে উইন্ডো বা ফ্রেমে কম্পোনেন্টকে সুবিন্যস্তভাবে সাজানো যায়। ইতোপূর্বে Border Layout, Grid Layout, Flow Layout প্রোগ্রাম নিয়ে আলোচনা করা হয়েছে। এ পর্বে আগের লেআউটগুলো থেকে সম্পূর্ণ আলাদা আরেকটি লেআউট Spring Layout নিয়ে আলোচনা করা হয়েছে। আগের লেআউটগুলোয়

উইন্ডোতে কম্পোনেন্ট সংযুক্ত করলে কম্পোনেন্টগুলো স্বয়ংক্রিয়ভাবে তাদের পজিশন ঠিক করে নিত। যেমন- ফ্লো লেআউটে কম্পোনেন্টগুলো একটির পর আরেকটি, গ্রিড লেআউটে রো এবং কলাম অনুসারে এবং বর্ডার লেআউটে দিক অনুসারে (উত্তর, দক্ষিণ, পূর্ব, পশ্চিম বা কন্টেইনারের মাঝখানে) অবস্থান করে। কিন্তু এ লেআউটগুলো দিয়ে কন্টেইনারের একটি নির্দিষ্ট বিন্দুতে যদি কম্পোনেন্ট রাখতে চাইলে তা সম্ভব নয়। এ কাজটি Spring Layout দিয়ে করা সম্ভব এবং এ কাজটি করার জন্য অবশ্যই Swing প্যাকেজটিকে ইম্পোর্ট করে নিতে হবে।



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

নিম্নের এই প্রোগ্রামটি নোটপ্যাডে টাইপ করে Sp\_Layout.java নামে সেভ করতে হবে।

```
import java.awt.*;
import javax.swing.*;
class Sp_Layout extends JFrame {
public Sp_Layout() {
super("Spring Layout");
setSize(250, 200);
//1
JPanel c = (JPanel) getContentPane();
c.setLayout(new SpringLayout());
//Button-1
c.add(new JButton("1"), new SpringLayout.Constraints(
Spring.constant(10),
Spring.constant(10),
Spring.constant(120),
Spring.constant(70)));
//Button-2
c.add(new JButton("2"), new SpringLayout.Constraints(
Spring.constant(160),
Spring.constant(10),
Spring.constant(70),
Spring.constant(30)));
//Button-3
c.add(new JButton("3"), new SpringLayout.Constraints(
Spring.constant(160),
Spring.constant(50),
Spring.constant(70),
Spring.constant(30)));
//Button-4
c.add(new JButton("4"), new SpringLayout.Constraints(
Spring.constant(10),
Spring.constant(90),
Spring.constant(50),
Spring.constant(40)));
//Button-5
c.add(new JButton("5"), new SpringLayout.Constraints(
Spring.constant(120),
Spring.constant(90),
Spring.constant(50),
Spring.constant(40)));
}
public static void main(String args[]) {
Sp_Layout frame = new Sp_Layout();
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.setVisible(true);
}
}
```

# জাভাতে স্প্রিং লেআউট প্রোগ্রাম

মো: আবদুল কাদের

## কোড বিশ্লেষণ

প্রোগ্রামটিতে JFrame-কে extends করা হয়েছে। ফলে একটি উইন্ডো তৈরি হবে। এর আকার প্রস্থ ২৫০ এবং উচ্চতা ২০০ পিক্সেল (১নং কমেন্ট চিহ্নিত লাইন)। উইন্ডোর টাইটেল বারে Spring Layout লেখাটি দেখানোর জন্য এর super ক্লাসে একটি স্ট্রিং 'Spring Layout' পাঠানো হচ্ছে, যাতে ফ্রেমের

টাইটেল বারে তা প্রদর্শিত হয়। প্রোগ্রামে ব্যবহৃত ফ্রেমে কম্পোনেন্ট বা বাটন বসানোর জন্য একটি প্যানেল c নেয়া হয়েছে। প্যানেলে কম্পোনেন্টগুলোকে সাজানোর জন্য setLayout মেথডের মাধ্যমে Spring Layout সেট করা হয়েছে।

এবার বাটন তৈরি করে বাটনটিকে একই সাথে প্যানেলে সংযুক্ত করা হবে। এভাবে মোট ৫টি বাটন 1, 2, 3, 4 এবং 5 তৈরি করা হবে। প্রথম বাটনটিকে প্যানেলে সংযুক্ত করে বাটনটি কোন পজিশনে থাকবে Spring.constant-এর মাধ্যমে তা উল্লেখ করা হয়েছে। প্রথম দুটি Spring.constant দিয়ে x ও y মান দেয়া হয়েছে। এর ফলে (x, y) অর্থাৎ (10, 10) বিন্দু হতে বাটনটি তৈরি হবে এবং এর প্রস্থ ও উচ্চতা পরবর্তী দুটি Spring.constant-এর মাধ্যমে সেট করা হয়েছে। এর প্রস্থ দেয়া হয়েছে ১২০ এবং উচ্চতা ৭০ পিক্সেল, যা (10, 10) বিন্দু থেকে প্রয়োজনানুযায়ী প্রসারিত হবে।



সবশেষে মেইন মেথডে

Sp\_Layout ক্লাসের অবজেক্ট তৈরি হবে। অবজেক্ট তৈরি হওয়ার সাথে সাথেই উপরের বর্ণনা মতে কাজগুলো

সংঘটিত হবে। মূলত মেইন মেথড থেকেই প্রোগ্রাম রান করে। প্রোগ্রামটি রান করার পর উইন্ডোর ক্লোজ বাটনে ক্লিক করলে যাতে তা পুরোপুরি বন্ধ হয়ে যায়, সে জন্য

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE) ব্যবহার করা হয়েছে। এই মেথডটি ব্যবহার না করা হলে ক্লোজ বাটনে ক্লিক করলে উইন্ডোটি দৃশ্যমান না থাকলেও রানিং অবস্থায় থাকে এবং মেমরি ব্যবহার করতে থাকে। এরপর প্রোগ্রামটি রান করার পর যাতে উইন্ডোটি দেখা যায় সে জন্য setVisible(true) কোড ব্যবহার করা হয়েছে। এটি ব্যবহার না করলে উইন্ডো দেখা যাবে না যদিও তা তৈরি হয়।

## প্রোগ্রাম রান করা

প্রোগ্রামটি রান করার জন্য কমান্ড প্রম্পট ওপেন করে নিচের চিত্রের মতো করে কোডগুলো রান করতে হবে।



চিত্র-প্রোগ্রাম রান করার পদ্ধতি



চিত্র-প্রোগ্রাম রান করার পর আউটপুট