

JAVA SWING :-

is a part of JFC (Java Foundation Classes)

→ to create window-based applications

Package: javax.swing

→ GUI Applications

JLabel, JTextField, JButton, JPanel, JTabbedPane, JFrame

AWT

- 1) Platform Dependent
- 2) Heavy weight
(It uses OS resources.)
- 3) NO PLuggable look & feel.
- 4) It has less components.

SWING

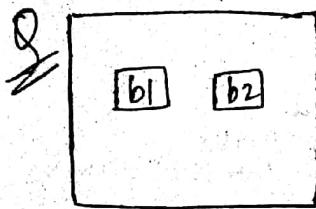
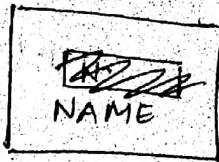
- 1) Platform Independent
- 2) Light weight. (use OS resources)
- 3) Pluggable look & feel.
- 4) It has more components than AWT.

WAP in JAVA for creating label in swing

```
Code
import java.awt.*;
import javax.swing.*;

class S1
{
    S1()
    {
        JFrame f = new JFrame();
        JLabel l = new JLabel("Name");
        f.setSize(200,300);
        f.setVisible(true);
    }
}
```

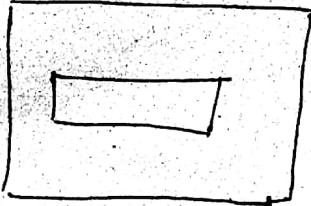
f.add();
B
public static void main (String args [])
{
 ~~fee~~ f = new fee(); sl f = new sl();
}
}



import javax.swing.*;
import java.awt.*;
class sl
{
 sl()
 {
 JFrame f = new JFrame();
 JButton b1 = new JButton("b1");
 JButton b2 = new JButton("b2");
 f.setSize(200,300);
 f.setVisible(true);
 }
 f.add(b1);
 f.add(b2);
 public static void main (String [] args)
 {
 sl f = new sl();
 }

To set position of button → b.setBounds (20,50,300,700)

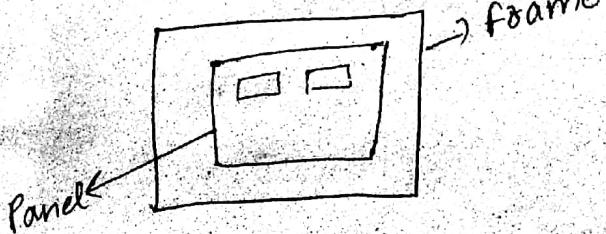
Buttons



Textfield

```
import javax.swing.*;  
import java.awt.*;  
class sl  
{  
    sl()  
    {  
        JFrame f = new JFrame();  
        JTextField t = new JTextField();  
        f.add(t);  
        f.setSize(200,300);  
        f.setVisible(true);  
    }  
    public static void main (String [] args)  
    {  
        sl f = new sl();  
    }  
}
```

How to create Panel in JAVA swing?



```
import javax.swing.*;  
import java.awt.*;  
class sl  
{  
    sl()
```

```

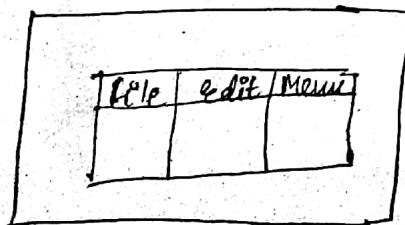
JFrame f = new JFrame();
 JPanel p = new JPanel();
 f.add(p);
 JButton b1 = new JButton();
 JButton b2 = new JButton();
 p.add(b1);
 p.add(b2);
 f.setSize(200,300);
 f.setVisible(true);
}

public static void main (String [] args)
{
    sl f= new sl();
}
}

```

Tabbed Pane

→ switch to different panels using tabbed pane



```

import javax.swing.*;
import java.awt.*;
class sl{
    sl()
    {
        JFrame f= new JFrame();
        JFrame tp = new JPanel();
        JPanel p = new JPanel ("file");
        Jp
    }
}

```

```

JPanel p1 = new JPanel();
JPanel p2 = new JPanel();
JPanel p3 = new JPanel();

JTabbedPane tp1 = new JTabbedPane("file", p1);
JTabbedPane tp2 = new JTabbedPane("edit", p2);
JTabbedPane tp3 = new JTabbedPane("Menu", p3);

f.add(p1);
f.add(p2);
f.add(p3);
f.add(tp1);
f.add(tp2);
f.add(tp3);

f.setSize(200, 300);
f.setVisible(true);

```

}
public static void main (String [] args)

{
 sl f = new sl();

}

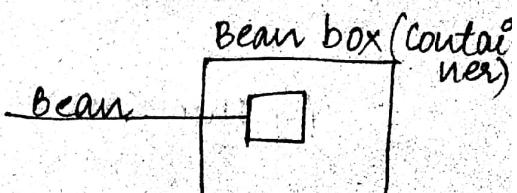
JAVA BEANS

Reusable software component of Java.

Java Bean Development Phase :-

- i) Construction Phase (Create Java Bean)
- ii) Build Phase
- iii) Execution Phase

JDBC
Java Beans
Java servlet

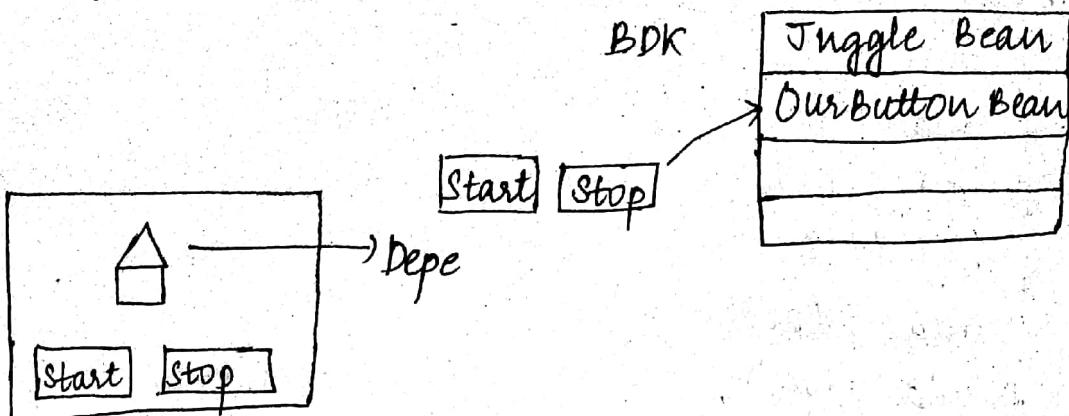


Building an application by using existing Bean

1) Identify the software for Java Bean

eg BDK, KAWA, FORTE for Java,
IBM Visual Age

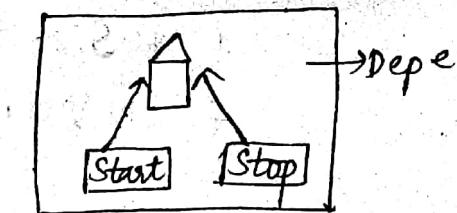
2) Identify Java Bean



3) Associate events with the Beans

4) Execute the Bean.

5) Save the Bean.



Using Java Beans in JSP Page :-

JSP → Java Server Page

<jsp:usebean> → Initializing Java Bean

id = " "

Session

scope = "Page/Request/Application"

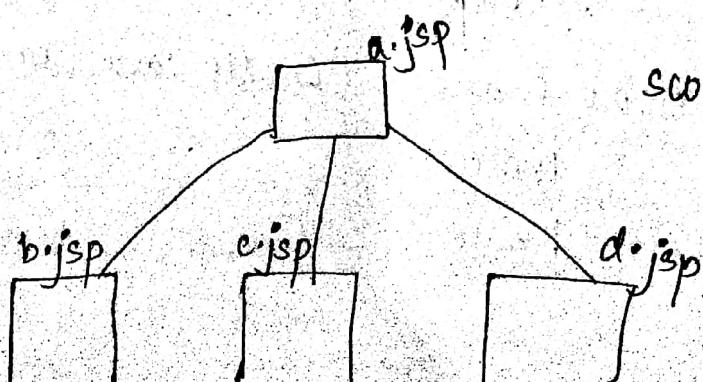
class = " "

type = " "

beanName = " "

</jsp:usebean>

scope = "Page"



(2) Specifying Property of Java Bean

```
<jsp:setProperty  
    Name = " "  
    Property = " " />
```

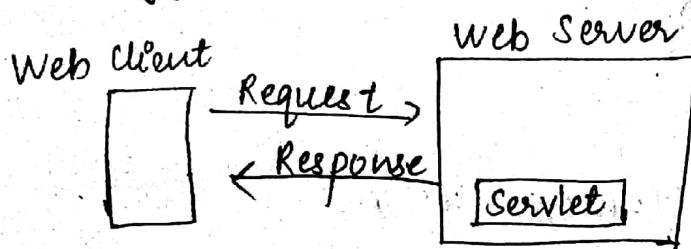
(3) Displaying property of Java Bean

```
<jsp:getProperty  
    Name = " "  
    Property = " " />
```

JAVA SERVLET

→ Server side component

→ Plugged into Java enabled Web Server



SERVLET API

↓ Application Programming Interface

javax.servlet
javax.servlet.http

Servlet Life Cycle

- 1) init() → establishing connection & open files.
- 2) service() → Accepts request and generate response
- (3) destroy() → clean resources

init (servletConfig config) ^{Object}

service (Servlet Request req, Servlet Response res)

Running the Servlet

- 1) Create the servlet
- 2) Compile the servlet
- 3) Start the server
- 4) Access the servlet

Servlet can be created by :

- 1) Create the servlet

- i) By extending Generic Servlet class
- ii) By extending HttpServlet class

```
res.setContentType("text/HTML");
```

```
PrintWriter pw = res.getWriter();
```

```
pw.println("Hello");
```

- 2) Compile the Server

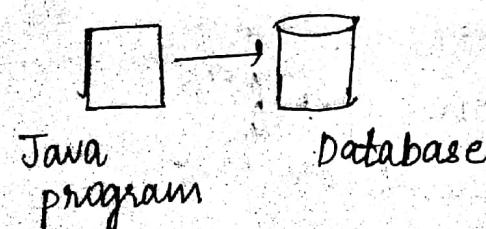
- 3) Start the Server (Apache Tomcat Server)

```
startup.bat
```

- 4) http://localhost:8080/Demo/Web

JDBC

Java Database Connectivity



JDBC :- It is an API

JDBC Drivers

- 1) JDBC - ODBC Bridge Driver
- 2) Native API Driver

3) Network Protocol Driver

4) Thin Protocol Driver

JDBC Connection steps

Java Pg to Database Connection

(1) Declaring variables.

• Connection con = null;

Statement Stmt = null;

```
Resultset rbt = null;
```

a) Establish connection

ii) Make connection

```
con = DriverManager.getConnection(URL, username, password)
```

③ Accessing Database

(Executing query)

SRNO	SNAME	SDOB

```
stmt = con.createStatement ("SELECT * FROM STUDENT")
```

```
ret = stmt.executeQuery()
```

4) Retrieving Result

list.next()

5) Close database connection

con.close()