

JAVA BEAN

classmate

Date 11/08/16
Page

A Java bean component is a reusable software component that can be mutually manipulated in builder tools.

Java bean is nothing but a java class swing are all the java bean component

COMPONENTS (Button, Textscreen, Label)

All Java bean classes must implement the Serializable interface.

The JavaBean technology is the component architecture for the Java platform. It defines Java software components and how they work together. New applications can be built using components.

PLATFORM INDEPENDENCE

Beans can be created according to the JavaBean API specification.

Beans run in any environment that supports Java.

PLATFORM INTEGRATION

Beans can bridge to native components model.

e.g. - activex, openDoc, Live Connect.

Software component that uses JavaBeans APIs are designed to be portable to various containers including Netscape, Internet Explorer, Visual Basic, Microsoft Word, Lotus Notes.

① Class Employee implements Serializable
java.io library

② {
 Private int i=10;
 Private String = "Hello";

③ Public Employee()
{

* Public void show()
{

 S.O.P ("Hello").

④ Public void setDisplay (int i)
{

 S.O.P (i);

}

⑤ Public void getD (int j)
{

 S.O.P (j);

Setter getter is normally used in the

password confirmation process.

* optional

Class A

{
P.S.V.M(—)

{
Employee e = new Employee();
e.setDisplay(10);
e.getD();

Public void setDisplay(int i)

{
3.D.P.---

}

Public int getD()

{
return i;

}

imp.

Class A1 implements java.io.Serializable.

{
Private String name;

Public A1()

Public void SetName(String name)

{
this.name = name;

}

Public String getName()

{
return name;

}

{

this. Keyword called Current
reference value

classmate

Date 11/08/13

Page

Glass A2

{

p.s.v.MC -)

{

```
    A1 a = new A1();
    S.O.P (a. Set Name (" KSB "));
    S.O.P (a. get Name ());
```

}

ADVANTAGES

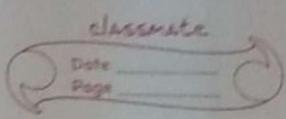
The property, method, events of bean that are exposed to another application can be controlled.

A bean may register to receive events from other objects and can generate event that are sent to those other objects.

Auxiliary software can be provided to help configure a java bean.

The Configuration Setting of a bean can be saved to persistant storage & restored.

interface → Method
implement → Variable



JAVA BEAN APIs

AppletInitializer

Methods in this interface are used to initialize beans that are also applets.

BEANINFO

This interface allows designer to specify information about the events, methods, properties of a bean.

CUSTOMIZER

This interface allows the designer to provide a graphic user interface through which a bean may be configured.

DESIGN MODE

Methods in this interface determine if the bean is executing in design mode.

EXCEPTIONALISTNER

The method in this interface is when an exception has occurred.

PROPERTYCHANGELISTENER

A method in this interface is involved when a bound property is changed.

PROPERTYEDITOR

objects that implements this interface allow the designer to change and display property value.

VETOABLECHANGELISTENER

- A method in this interface is invoked when a constrained property is changed.

VISIBILITY

Methods in this interface allow a bean to execute in environments where the GUI is not available

```
int rev=0;
int rem;
int num=121;
num1=num;
while (num>0)
{
    rem=num%10;
    rev=rev*10+rem;
    num=num/10;
}
if (rev==num)
{
    palindrome;
}
```

BDK BEANS DEVELOPMENT KIT

- The bean development kit is intended to support the early development of Java bean components. To act as standard reference base for both bean developers and tool vendors. The BDK provides a reference bean container, the bean box and a variety of reusable example source code for use by both bean developers and tool developers.

Class E

{

P.S.V.M (String S[])

{

Int a;

a = Integer · parseInt (S[0]);

if (a % 2 == 0)

{

S.O.P ("even");

}

else

{

S.O.P ("odd");

}

Java = 2 in 32 bit

4 in 64 bit

compareTo use for compare

ASCII value its results in int.

Class A

{

P-S-V-M (String S[7])

String SJ = new String ("KSB")

if (S[0] equals ("ksb"));

{

S.O.P ("They are equal");

}

else

{

S.O.P ("They are not");

}

}

EJB

Enterprise Java beans

EJB is acronym for Enterprise Java Beans. It is an specification provided by sun microsystems to develop secure, robust and scalable distributed application.

To run EJB application, you need an application server (EJB Container) such as Jboss, GlassFish, WebLogic, WebSphere (These are servers) It performs:

- a) Life Cycle Management
- b) Security
- c) Transaction Management
- d) Object Pooling

EJB application is deployed on the server, so it is called server site component also.

EJB is like COM (Component Object Model) provided by Microsoft.

When use EJB.

Application needs remote access:

In other words, it is distributed.

Application needs to be scalable.

EJB application supports load balancing, clustering and failover.

Application need encapsulated business logic.

EJB application is separated from presentation & persistent layer.

Types of EJB

These are three types of EJB.

Session bean.

It contains business logic that can be invoked by local, remote or web service client.

Message driven bean

Like session bean ; It contains the business logic but it is invoked by passing message.

Entity bean.

It encapsulates the state that can be persistent in a data base.

DISADVANTAGES

Requires Application Server

Requires only java client. For other language client
• You need to go for web service.

Complex to understand & develop EJB application.

JSP - Java + HTML

HTTP → Stateless proto-ctd.

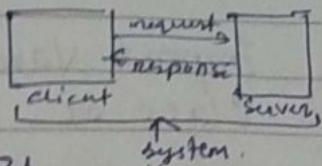
Because it doesn't contain info. of user.

Keshav. char At(0);
↓
S[0]. "(0);

Destroy → Method

Servlet → Class
PrintWriter - ~~for output~~

SERVELETS



- Java servelets are java classes run by a webserver. That has an interpreter that supports a java servelets specification.
- Servelets can be created using `java x.Servleter` or `javax.Servleter.http`. (packages), which are the standard part of the Java enterprise edition, an expended version of java class library that supports large scale development projects. These classes implement the java servleter & JSP (Java Server Page) specification.
- Java servelets have been created & Compiled just like any other java class.

After user installed the servleter package and them to your computer class path, you can compile servleters with the .JDK's, Java compiler or any other current compiler.

Serverite programming language.

```
try  
{  
}  
catch (Exception e)  
{  
    throw e;  
}
```

```
Class 2  
{  
    public void () throws Exception  
    {  
    }  
}
```

Check = Compiling throw - Use with variable
unchecked = runtime . . .
Throws - Use with Method
Throwable is top most Exception - class
class of all

```

import javax.servlet.http.*;
class S1
{
    public void doGet (HttpServletRequest request,
                      HttpServletResponse response)
    {
        PrintWriter pw = response.getWriter();
        its object pw, calling
        is "out" S.O.P ("Hello!");
        or
        pw.println ("alone");
        System.out.println
        class object method
    }
}

```

SERVELET LIFECYCLE

A servelet lifecycle can be defined as the entire process from its creation till the destruction. The following are the paths followed by a servelet

a..

The servelet is initialized by calling the init() method.

b..

The servelet calls service method to process a client request.

c..

The servelet is terminated by calling the destroy() method.

d..

Finally servlet is garbage collector by the garbage collector of the JVM.

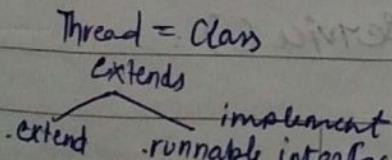
THE init() method

The init() method is designed to be called only once. If it is called when servlet is first created or not called again for each user request. So, it is used for One Time initialisation, just as with the init() method of applets.

The servlet is normally created when a user first invokes a URL corresponding to the servlet but you can also specified that the servlet preloaded when the server is first started.

When a user invokes a servlet, a single instance of each servlet gets created, with each user request resulting in a new thread i.e. handled by do get or do post.

The init() method simply creates or loads some data that will be used throughout the life cycle of servlet. The init method looks like as



Public void doGet (HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException.

doPost() method

A post request results from an HTML form that specifically lists POST as the method or it should be handled by doPost() method.

Public void doPost (HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException.

Class Add

```
{  
    public static void Main ( String A [ ] )
```

{

int a = Integer . parseInt (A [0]);

int b = Integer . parseInt (A [1]);

System . out . println (a + b);

}

}

The `destroy()` method call only once at the end.
this method gives your servlet a chance to
close database connections, halt background threads,
write cookies list, or hitcounts to disk,
or perform other such cleanup activities.

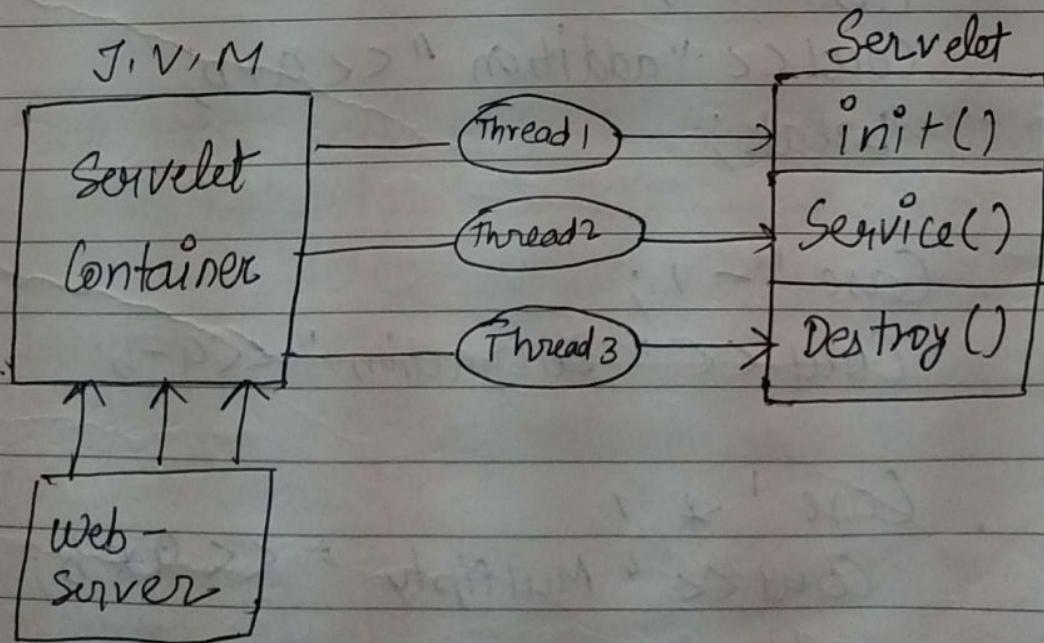
After the `destroy()` method is called
the servlet object is marked for garbage
collection.

Public void `destroy()`

{

}

SERVLET ARCHITECTURE



JAVA SCRIPT

classmate

Date 22/08/16

Page

7

Java Script is a programming language design for web pages.

The Java script is interpreted by the web browser.

Java script is platform independent or object oriented.

WHY TO USE JAVASCRIPT ?

Java script enhances webpages with dynamic or interactive features.

Java Script runs in client software

Unlike HTML, Java script is case sensitive.

Java script is a most popular scripting language on the internet, & works in all major browser such as Internet explorer, Mozilla, firefox, Netscape, opera, chrome.

OBJECTS

Objects refers to windows, documents, images, tables, forms, buttons or links.

Object should be named.

Object have properties that act as modifier.

PROPERTIES

Properties are object attributes.

Object properties are defined by using the object's name, a period, & property name.

Eg: document.bgcolor

↑ ↑
object property.

METHODS

Methods are actions applied to particular object.

Eg: document.write("Hello");

EVENTS

Events associated with an object with an action

for eg: Onmouseover Event handler action may change an image.

OnSubmit Event handler sends a form.

FUNCTIONS

functions are named statements that perform task.

Eg: Function add (int a)
{
 return a;
}

VARIABLES

Variables contain values & use the equal sign to specify the value.

Variables are created by declaration using the "Var" command. with or without an initial value state.

for eg - Var month = August;

HOW TO PUT JAVASCRIPT into HTML Page?

```
<html>
<body>
<Script type = "text / javascript.">
document . write ( "Hello" );
</script>
</body>
</html>
```

Write any 5 event-handlers in
java script? with proper program format.

classmate

Date 25/08/16
Page 11

WAD In Javascript whether the no. is +ve or -ve

```
<html>
<head> <script> </head> *
<body>
<script type = "text/java script"> ... int Var = 5;
if ( i > 0 )
{
    document.write ("+ve")
}
else
{
    document.write ("-ve")
}

</body>
</html>
```

★ EVENT HANDLER .

• ON SUBMIT

On submit is an event that is used
when you try to submit a form.
we use validate function

```
<html>
<head>
<script> ----- </script>
</head>
function validation()
{
    return True or False
}
```

```
<body>
<form method="get/post" action="path"
      onsubmit="return validate()">
```

Validate()
none,
cause:
See:
Arg
Submit()

```
<input type="button" value="Submit">
</form>
</body>
</html>
```

Q Write any 5 event handler in javascript
with proper program format.

On click

On submit-

On change

On mouse click

On Reset

On load

On abort

On unload

On Select

On mouse over

ONSUBMIT

On submit is an event that is fired
when you try to submit a form.
we use validate function.

```
<html>
<head>
<script> ----- </script>
</head>
function validation()
{
    return True or False
}
<body>
<form method = "get/post" action = "path">
    On submit = "return validate()"
    <input type = "button" . value = "Submit" >
</form>
```