Dt : 18/5/2022

Types of Applications:

=>Applications are categorized in the following:

(a)StandAlone Applications

(b)Web Applications

(c)Enterprise Applications

(d)Mobile Applications

(a)StandAlone Applications:

=>The applications which are installed in one computer and performs actions in the same computer are known as StandAlone Applications.

=>which are also known as DeskTop applications or Windows Applications.

Note:

=>According to developer StandAlone applications means,

No HTML input

No Server Environment

No DataBase Storage

=>Based on User interaction these StandAlone applications are Categorized into two types:

(i)CUI Applications

(ii)GUI Applications

(i)CUI Applications:

=>The applications in which the user interacts through Console are known as CUI Applications.

(CUI - Console User Interface)

(ii)GUI Applications:

=>The applications in which the user interacts through GUI components are known as GUI Applications.

(GUI - Graphical User Interface)

Note:

=>we use AWT(Abstract Window Toolkit) and Swings to design GUI components

---

(b)Web Applications:

=>The applications which are executed in Web Environment or Internet Environment are known as Web Applications.

=>To Construct WebApplications we use JDBC,Servlet and JSP.

Note:

=>These Web Applications are executed in 'WebContainer'

=>This WebContainer is available from WebServers and Application Servers.

(c)Enterprise Applications:

=>The applications which are executing in distributed Environment and depending on the features like Security, Load Balancing and Clustering process are known as Enterprise Applications or Enterprise Distributed Applications.

(d)Mobile Applications:

=>The applications which are executed in Mobile environment are known as Mobile Applications.

==========

faq:

define Server?

=>Server means service provider,which means accepting the request and providing the response.

define Client?

=>The user who generate request to the Server is known as Client.

imp

Types of Servers:

=>Servers are categorized into two types:

(i)WebServers

(ii)Application Servers

(i)WebServer:

=>Web Server contains only Web container .

=>A web server is good in case of static contents like static html pages.

=>Web server consumes less resources like CPU, Memory etc. as compared to application server.

=>Web Server provides the runtime environment for web applications.

=>Web Server supports HTTP Protocol

=>Apache Web Server.(Tomcat)

(ii)Application Server:

=>Application Server contains both Web Container and EJB Container.(EJB - Enterprise Java Bean)

=>Applcation server is relevant in case of dynamic contents like bank websites.

=>Application server utilizes more resources

=>Application server provides the runtime environment for enterprise applications.

=>Application Server suppots HTTP as well as RPC/RMI protocols.

(RPC - Remote Procedure call)

(RMI - Remote Method Invocation)

=>Weblogic, JBoss.

**imp*

Installing Tomcat Server:

step1 : Download Tomcat9.x WebServer

webserver - Tomcat 9.x

(Compatable with JDK1.8 and Above)

vendor - Apache org

default port no - 8080

download - www.apache.org(Open source)

Note:

=>WebContainer internally has two SubContainers

(i)Servlet container

(ii)JSP Container

Servlet container : Catelina

Jsp Container : Jasper

step-2 : Install the Tomcat Server

while Installation process,

Select the type of install : Full(click next)

Server Shutdown port : 8089

HTTP/1.1 Connector port : 8081 or 8082 or 8083...

User Name : Venkatesh

Password : nit

(click next)

step-3 : Start the Tomcat Server

=>To start Tomcat server click on 'startup' or 'Tomcat9w'

from 'bin' of Tomcat

C:\Tomcat 9.0\bin

step-4 : Access Tomcat Server from WebBrowser using the

following URL:

http://localhost:8081

step-5 : Stop the Tomcat Server

=>To stop Tomcat server click on 'shutdown' or 'Tomcat9w'

from 'bin' of Tomcat

C:\Tomcat 9.0\bin

==========

Dt : 19/5/2022

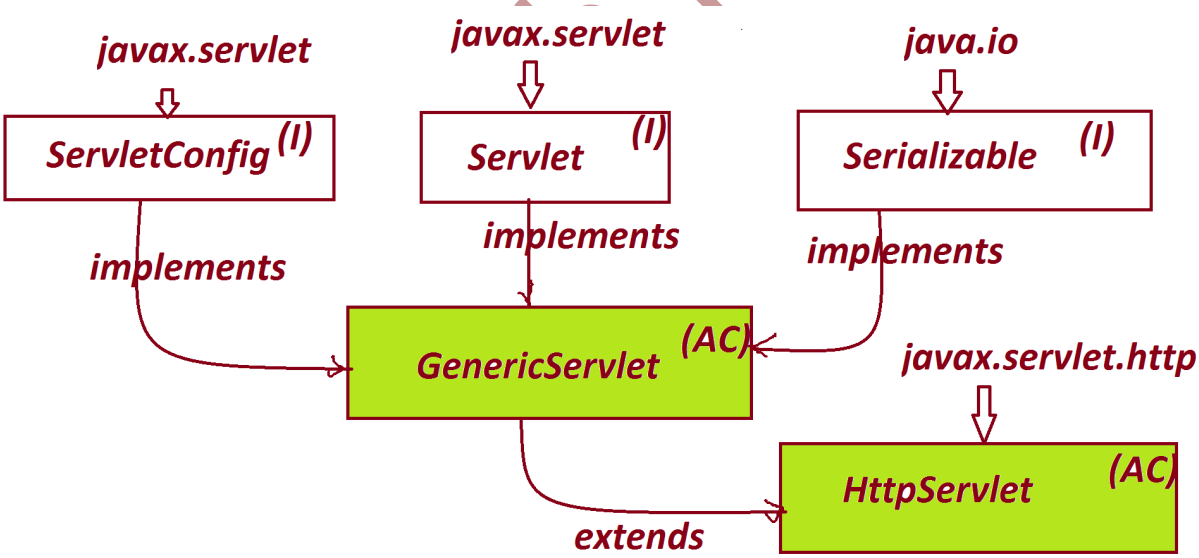
**imp*

Servlet API:

=>'javax.servlet' package is known as Servlet API and which provide classes and interfaces used in Servlet application development.

=>'javax.servlet.Servlet' interface is the root of Servlet API.

Hierarchy of Servlet API:



Note:

=>In the process of constructing Servlet programs,the programs

must be extended from any one of the following:

(a)GenericServlet

(b)HttpServlet

(a)GenericServlet:

=>When ServletProgram is extended from GenericServlet then the following Life-Cycle methods are available

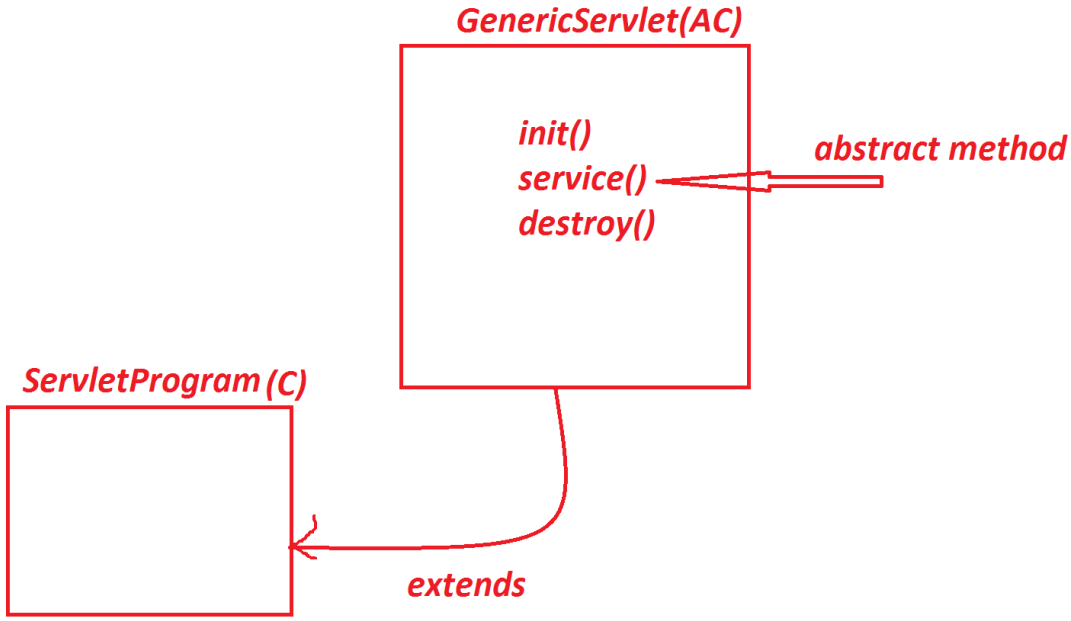
(i)init()

(ii)service()

(iii)destroy()

=>These methods are automatically executed in the same order.

=>GenericServlet is Protocol independent,which means ready to accept request from any type of protocol.



(b)HttpServlet:

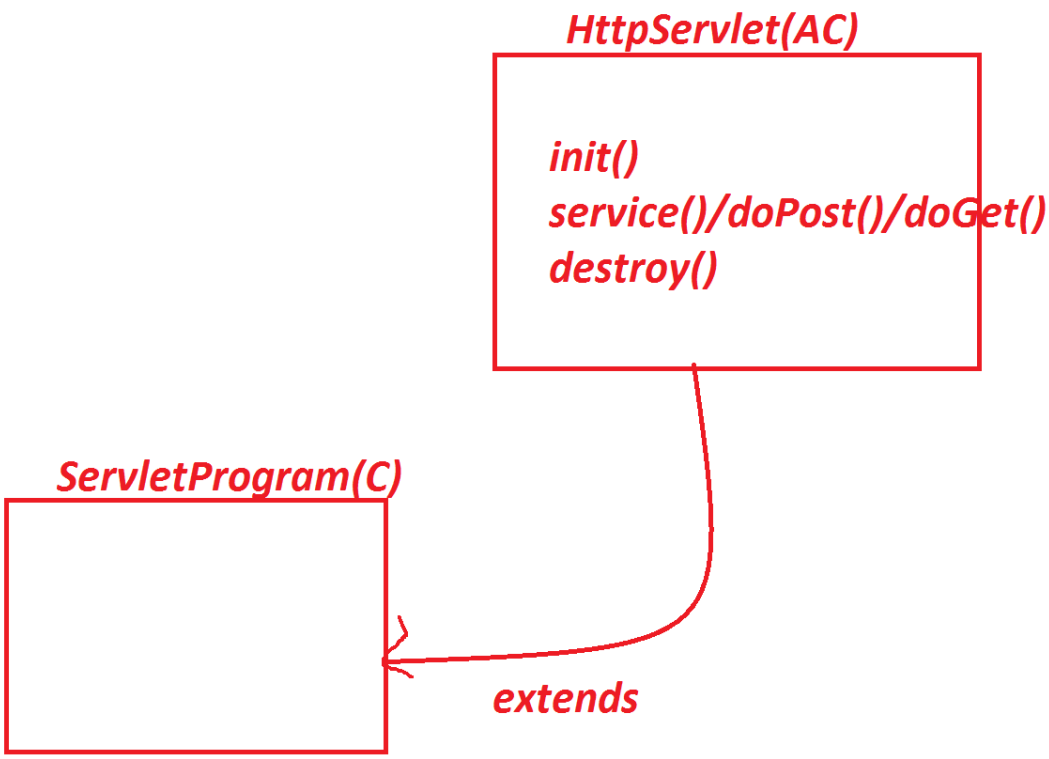
=>when ServletProgram is extended from HttpServlet then the following Life-Cycle methods are available

(i)init()

(ii)service()/doPost()/doGet()

(iii)destroy()

=>HttpServlet is Protocol dependent,which means accepts the request from Http Protocol.



==========

**imp*

Constructing Servlet Application using IDE Eclipse:

step-1 : Open IDE Eclipse,while opening name the WorkSpace and

click finish

step-2 : Create Dynamic Web Project

Click on File->new->Project->Web->select 'Dynamic Web Project'->

click 'next'->name the Project and click 'finish'

step-3 : Add 'servlet-api.jar' to Dynamic Web Project

RightClick on Dynamic Web Project->Build Path->

Configure Build path->Libraries->select 'classpath and click

Add External Jars'->Browse and select 'servlet-api.jar' file

from 'lib' of Tomcat->open-Apply->Apply and Close.

step-4 : Add Tomcat Server to IDE environment

Click Servers->click 'click this link to create new server'->

select the type of server and click next-> Browse Tomcat

Installation directory and click finish.

step-5 : Construct the Servlet Application.

input.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<form action="dis" method="post">
UserName:<input type="text" name="uname"><br>
MailId:<input type="text" name="mid"><br>
<input type="submit" value="Display">
</form>
```

```
</body>
</html>
```

DisplayServlet.java

```
package test;

import java.io.*;

import javax.servlet.*;

import javax.servlet.annotation.*;

@SuppressWarnings("serial")

@WebServlet("/dis")

public class DisplayServlet extends GenericServlet{

        public void init()throws ServletException{

                //NoCode

        }

   public void service(ServletRequest req,

                ServletResponse res)throws ServletException,

       IOException{

        String uName = req.getParameter("uname");

        String mId = req.getParameter("mid");

        PrintWriter pw = res.getWriter();

        res.setContentType("text/html");

        pw.println("====UserDetails===");

        pw.println("<br>UserName:"+uName);

        pw.println("<br>MailId:"+mId);

   }

   public void destroy() {
```

```
        //NoCode
    }
}
```

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app>
    <welcome-file-list>
        <welcome-file>input.html</welcome-file>
    </welcome-file-list>
</web-app>
```

step-6 : Execute the Servlet Application

RightClick on Dynamic Web Project->Run as->Run on Server->

select the server and click finish.

==========