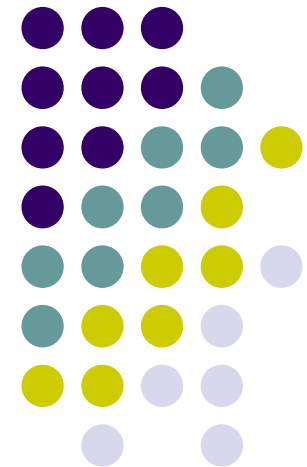
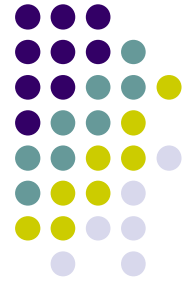


# Serverske tehnologije

---

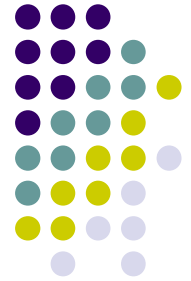
Servleti, JSP, SSI  
Maja Štula  
ak. god. 2010/2011





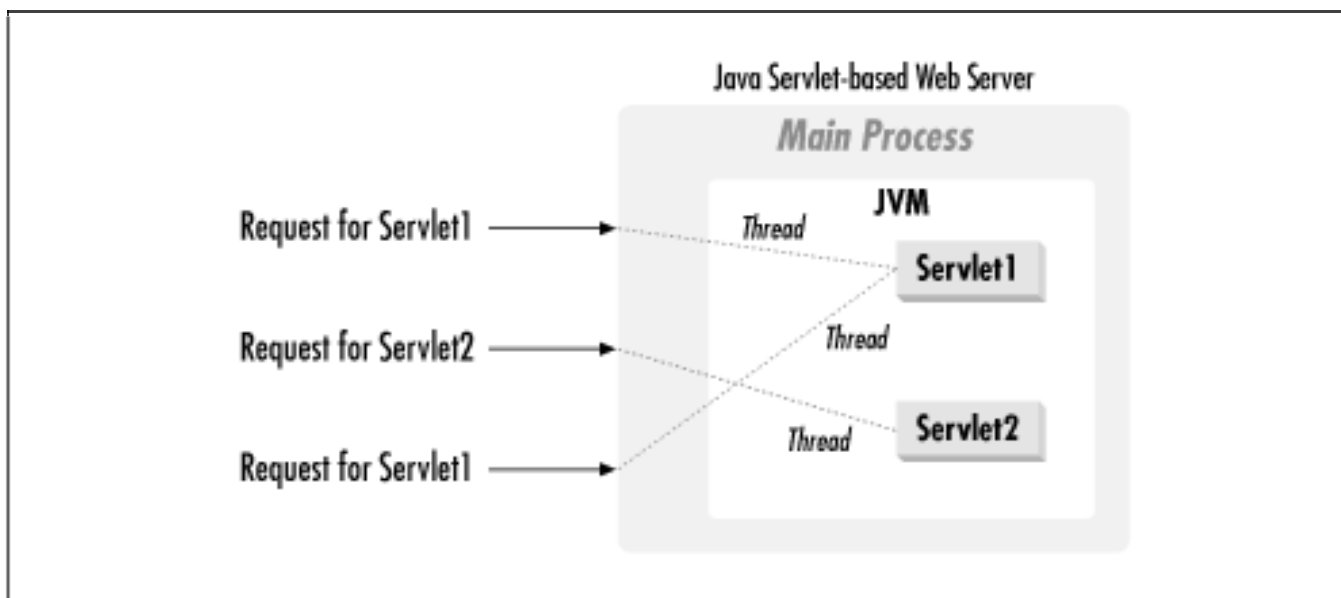
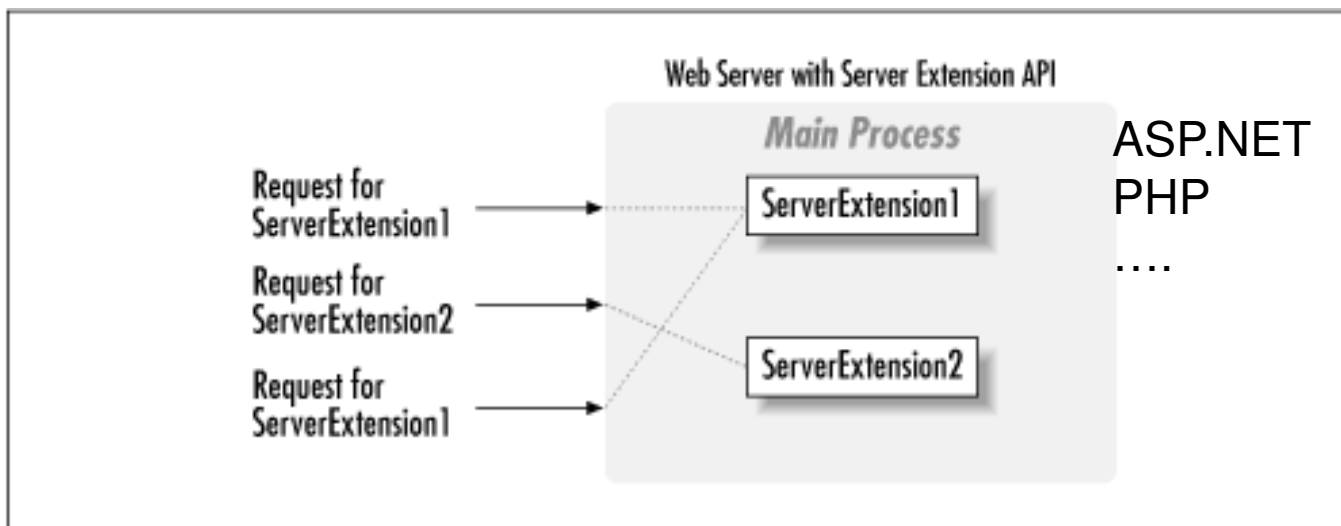
# Java Servleti

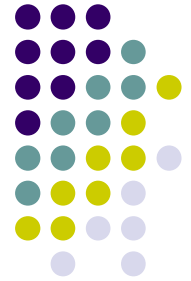
- Servleti su moduli Java kôda koji se izvode unutar aplikacije servera. Odatle i ime servleti slično appletima na strani klijenta.
- Slično CGI programima odgovaraju na HTTP zahtjev korisnika.
- Servleti nisu vezani uz određeni klijent-server protokol, ali se najčešće koriste s HTTP protokolom.



# CGI ↔ Servlet

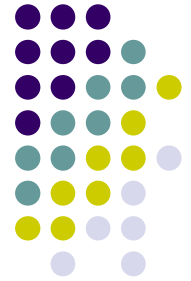
- Kod tradicionalnog CGI-a za svaki HTTP zahtjev se pokreće novi proces.
- Ako je CGI program brz najveći dio vremena se troši na pokretanje procesa.
- Kod servleta svaki HTTP zahtjev se obrađuje kao novi thread.
- Kod klasičnog CGI-a ako istovremeno postoji  $N$  zahtjeva na isti program, program se učitava u memoriju  $N$  puta.





# Java Servleti

- Servleti koriste klase iz Java paketa `javax.servlet` (osnova servleta) i `javax.servlet.http` (proširenje servleta tako da mogu odgovarati i na HTTP zahtjeve).
- Osnovne razlike između servleta i CGI-a:
  - Servlet se ne izvodi kao zasebni proces. To znači da se ne treba za svaki HTTP zahtjev stvarati novi proces.
  - Servlet ostaje u memoriji između dva zahtjeva dok se CGI program treba učitati nanovo za svaki HTTP zahtjev.
  - Postoji samo jedna instanca koja istovremeno odgovara na sve zahtjeve.



# Java programski jezik

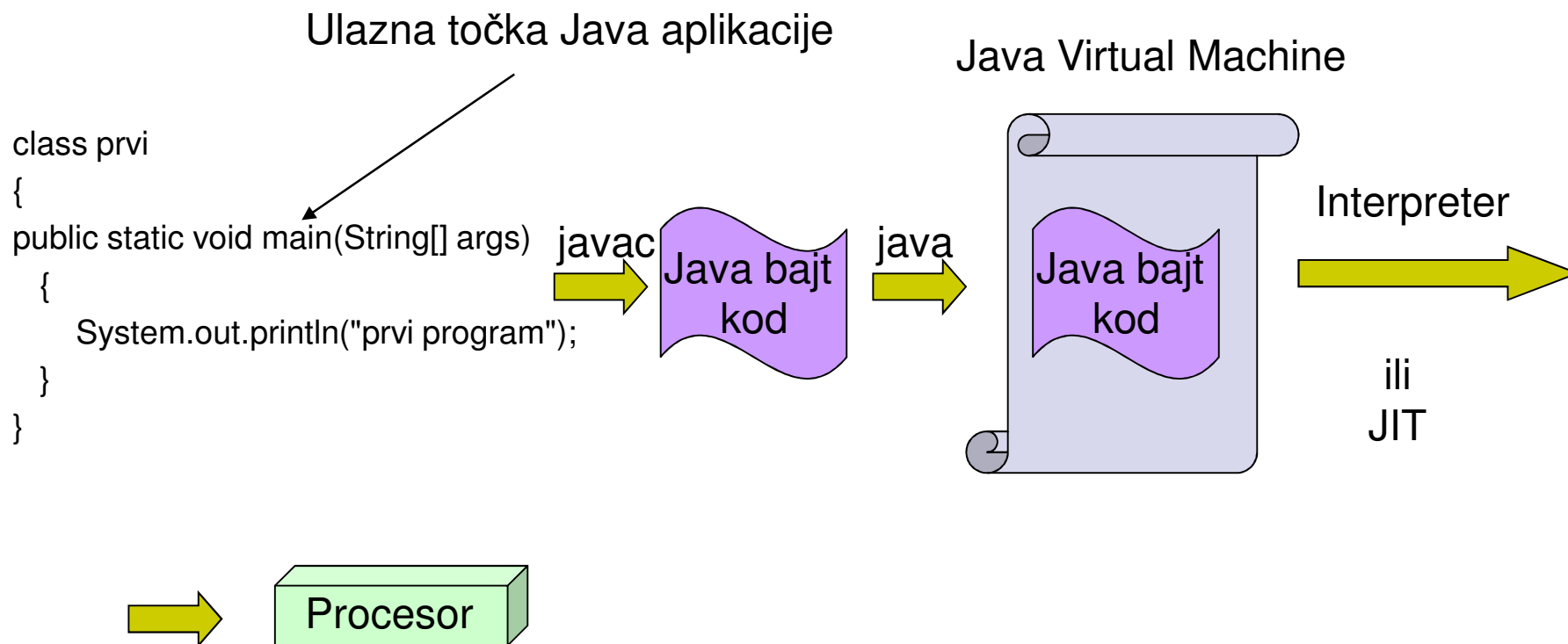
- Java je objektno orijentirani programski jezik koji je razvio Sun (<http://java.sun.com>) i platforma za izvršavanje Java programa.
- Trenutna verzija Jave je Java 2 Platform.
- Java 2 Platform dolazi u tri verzije: Micro, Standard (Standard Edition (J2SE)) i Enterprise.
- Različite verzije imaju podržane različite funkcionalnosti.
- Micro Edition se koristi za male aplikacije dok se Enterprise Edition koristi za velike, serverske aplikacije.
- Primjeri su rađeni u J2SE.

# Java programski jezik



- Java kôd se prevodi (*javac*) u Java bajt kôd (Java byte code) koji se onda interpretira ili kompajlira sa JIT (*Just-In-Time*) kompajlerom u JVM (*Java Virtual Machine*).
- JVM specifikacija predstavlja specifikaciju platforme za izvršavanje Java programa. To je programski model izmišljenog procesora.
- Programi napisani u Javi se prevode za izvršavanje na ovakvoj platformi. Platforma je dizajnirana da što manje zavisi o specifičnim karakteristikama konkretnog hardvera. Ideja je da se jednom napisan i *preveden* program izvršava na bilo kojoj platformi koja podržava Javu (Unix, Win...)
- Time je riješen problem distribucije binarnih formata koji su različiti za različite arhitekture, operacijske sustave i različita sučelja.

# Java programski jezik



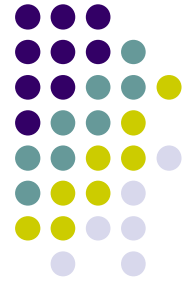




# Java programski jezik

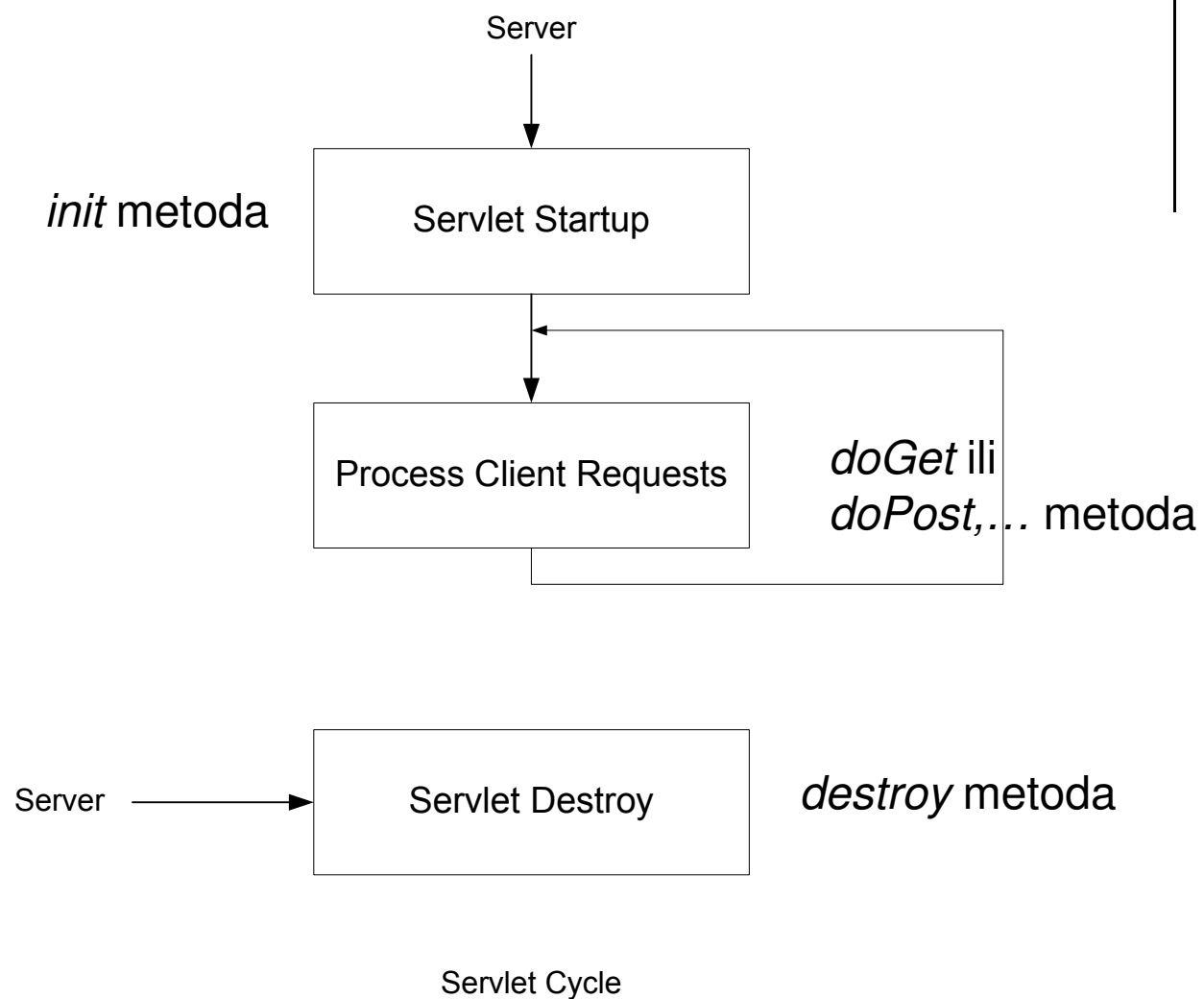
- Naredba import koristi se za uključivanje Java paketa u aplikaciju (npr. `import java.io.*;`).
- Java paket je kolekcija klasa slične funkcionalnosti koja se kreira korištenjem naredbe `package` na početku koda prije bilo koje druge naredbe, također omogućava kreiranje imenskih prostora jer su sve klase unutar paketa u istom imenskom prostoru. Obično se imenski prostor označava sa više labela (`java.io`).
- Npr.

```
package moj.prvi
class prvi
{
    public static void main(String[] args)
    {
        System.out.println("prvi program");
    }
}
```



# Java Servleti

- Servlet je u svom najopćenitijem obliku instanca klase koja implementira `javax.servlet.Servlet` sučelje (*interface*).
- Većina servleta ipak koristi već standardne implementacije `javax.servlet.Servlet` sučelja, a to su klase `javax.servlet.GenericServlet` i `javax.servlet.http.HttpServlet`.
- `HttpServlet` je apstraktna klasa za kreiranje servleta. Za definiranje funkcionalnosti servleta potrebno je samo *overajdati* metodu `doGet` ili `doPost`.
- Da bi inicijalizirao servlet, na korisnikov zahtjev Web server učitava klasu koja implementira sučelje `Servlet` i kreira instancu procesa pozivajući konstruktor za tu klasu.
- Tada se poziva *init* metoda (iz sučelja `javax.servlet.Servlet`) koja vrši setup i inicijalizaciju servleta.





# Java Servleti

- Kada se neki servlet pozove prvi put učitava se u memoriju.
- Nakon što je HTTP zahtjev obrađen servlet ostaje u memoriji sve dok je Web server aktivan.
- Prvi put kada se servlet učitava u memoriju poziva se njegova `init()` metoda. Ta metoda se poziva samo prvi put tj. za sve naknadne zahtjeve servletu ta se metoda više ne poziva.
- Većina Web server ima mogućnost učitavanja često korištenih servleta prilikom podizanja servera tako da su ti servleti već učitani u memoriju prije no što dođe prvi zahtjev za tim servletima.

# Java programski jezik

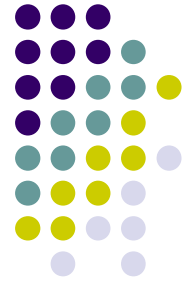
Pravo pristupa može biti private, public, protected



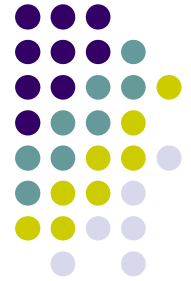
```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class HelloWorld extends HttpServlet
{
    public void doGet(HttpServletRequest request, HttpServletResponse response) throws IOException,
        ServletException
    {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println("<html>");
        out.println("<head>");
        out.println("<title>Hello World!</title>");
        out.println("</head>");
        out.println("<body>");
        out.println("<h1>Hello World!</h1>");
        out.println("</body>"); out.println("</html>");
    }
}
```

Nasljeđivanje

## Ispis tekstualnog izlaza

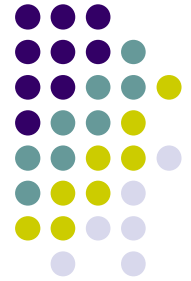


```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class HelloWorld extends HttpServlet
{
    public void doGet(HttpServletRequest request, HttpServletResponse
        response) throws IOException, ServletException
    {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println("<html>");
        out.println("<head>");
        out.println("<title>Hello World!</title>");
        out.println("</head>");
        out.println("<body>");
        out.println("<h1>Hello World!</h1>");
        out.println("</body>"); out.println("</html>");
    }
}
```



# Java Server Pages (JSP)

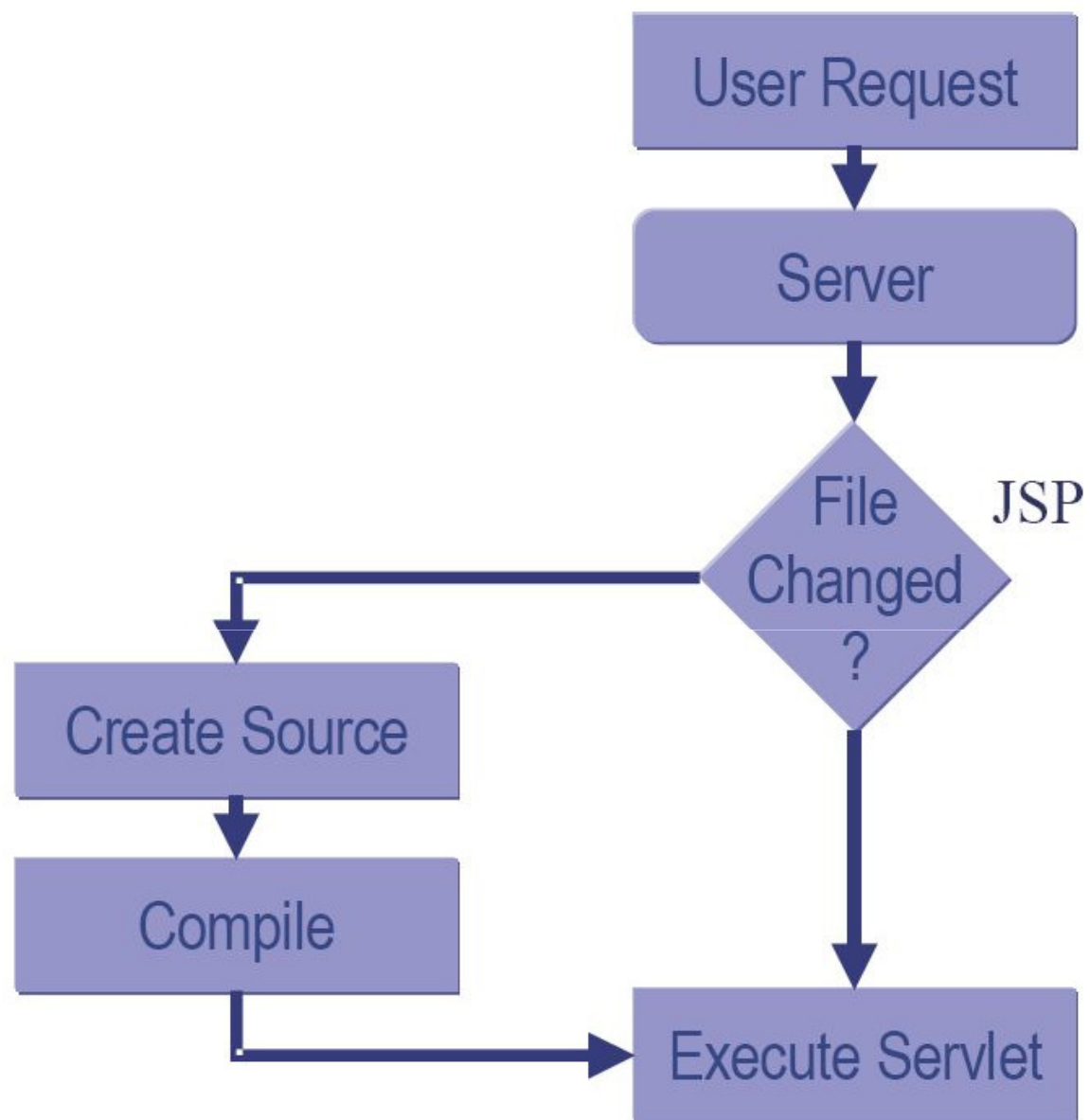
- Java Server Pages (JSP) omogućava kombiniranje HTML kôda sa programskim kôdom (slično kao i ASP).
- JSP se implementira tako da kada korisnik pozove JSP stranicu, iz JSP stranice se kreira servlet i izvrši.
- JSP stranica sadrži tekst (HTML kôd), dijelove Java kôda, akcije, deklaracije, izraze i direktive koje server pretvara u odgovarajući servlet.
- Tekst se prebacuje u naredbu PrintWriter za ispis na standardni izlaz servleta, a Java kôd, akcije, deklaracije, izrazi i direktive se prebacuju direktno u servlet.



# Java Server Pages (JSP)

- Znači datoteke sa ekstenzijom .jsp se kompajliraju u servlet i onda izvode.
- Ako je datoteka .jsp novija nego je servlet koji se trenutno nalazi u memoriji tada se taj servlet briše, stranica se ponovo kompajlira i pokreće.







# Java Server Pages (JSP)

- Blok servlet kôda se umeće između tagova `<% %>`
- Java deklaracija se umeće između tagova `<%! %>`
- Java izraz se umeće između tagova `<%= %>`, izraz se računa i pretvara u string koji se prikazuje na rezultirajućoj stranici

`<%! declaration %>`

`<%! String[] fruits; %>`

`<% scriptlet %>`

```
<%   if (fruits != null) {  
for (int i = 0; i < fruits.length; i++) {  
%>
```

`<%= expression %>`

`<body bgcolor=<%= cb.getColor1() %>`



# Java Server Pages (JSP)

- Direktiva se odnosi na nekakve globalne aspekte JSP stranice poput npr. uključivanja nekog drugog resurs u stranicu.

```
<%@ directive ...%>
```

```
<%@ page language="java" import="cal.*" %>
```

- Osim page direktive postoje još i

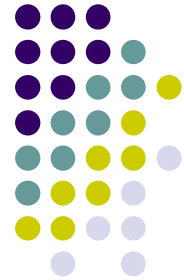
```
<%@ include file="foo.jsp" %>
```

```
<%@ taglib prefix="tags" tagdir="/WEB-INF/tags" %>
```

```

<HEAD><TITLE> Calendar: A JSP APPLICATION</TITLE></HEAD>
<BODY BGCOLOR="white">
<%@ page language="java" import="cal.*" %>
<jsp:useBean id="table" scope="session" class="cal.TableBean" />
<%
    table.processRequest(request);
    if (table.getProcessError() == false) {
%>
<CENTER><TABLE WIDTH=60% BGCOLOR=yellow CELLPADDING=15>
<TR><TD ALIGN=CENTER> <A HREF=cal1.jsp?date=prev> prev </A>
<TD ALIGN=CENTER> Calendar:<%= table.getDate() %></TD>
<TD ALIGN=CENTER> <A HREF=cal1.jsp?date=next> next </A>
</TR></TABLE><TABLE WIDTH=60% BGCOLOR=lightblue BORDER=1 CELLPADDING=10>
<TR><TH> Time </TH><TH> Appointment </TH>
</TR><FORM METHOD=POST ACTION=cal1.jsp>
<%
    for(int i=0; i<table.getEntries().getRows(); i++) {
        cal.Entry entr = table.getEntries().getEntry(i);
%>
        <TR><TD>
            <A HREF=cal2.jsp?time=<%= entr.getHour() %>>
                <%= entr.getHour() %> </A></TD>
            <TD BGCOLOR=<%= entr.getColor() %>>
                <% out.print(util.HTMLFilter.filter(entr.getDescription())); %>
            </TD> </TR>
<%
    }
%>
</FORM></TABLE><BR>
<TABLE WIDTH=60% BGCOLOR=yellow CELLPADDING=15><TR>
<TD ALIGN=CENTER> <% out.print(util.HTMLFilter.filter(table.getName())); %> :
                <% out.print(util.HTMLFilter.filter(table.getEmail())); %> </TD>
</TR></TABLE></CENTER>
<%
    } else {
%>
<font size=5>    You must enter your name and email address correctly.</font>
<%
    }
%>
</BODY></HTML>

```



# Java Server Pages (JSP) tagovi

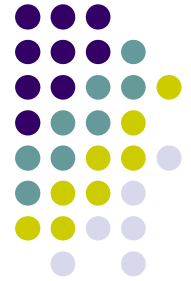


- Akcije se definiraju preko JSP tagova. JSP tag je sličan HTML tagu.
- Može imati početni i krajnji tag. Imena tagova imaju pridružen znak :, dio prije dvotočke opisuje tip taga.
- Npr.:  
`<some:tag> body </some:tag>`
- Postoje predefinirani JSP tagovi određene funkcionalnosti.

# Java Server Pages (JSP) tagovi

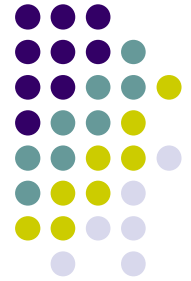


- Npr. `<jsp:getProperty>` tag dohvaća vrijednosti Bean svojstva
- **Sintaksa**  
`<jsp:getProperty name="beanInstanceName"  
property="propertyName" />`
- **Primjer**  
`<jsp:useBean id="calendar" scope="page"  
class="employee.Calendar" />`  
`<h2>`  
Calendar of  
`<jsp:getProperty name="calendar"  
property="username" />`  
`</h2>`



# Java Beans

- Java Beans su Java komponente izvršnog koda.
- Java bean je samo instanca neke Java klase.
- JSP ima tri taga za rada sa Java Beans-ovima:
  - `<jsp:useBean id="bean name" class="bean class" scope = "page | request | session | application" />`
  - `<jsp:setProperty name = "id" property = "someProperty" value = "someValue" />`
  - `<jsp:getProperty name = "id" property = "someProperty" />`



# Java Beans

- Atribut *scope* definira doseg bean-a.
- Page scope : objekt postoji samo za vrijeme generiranje stranice, ovo je defaultni doseg.
- Request scope : objekt se kreira u sklopu promatranog zahtjeva.
- Session scope : objekt je vezan uz promatranu sesiju.





# Java Server Pages 2.x

- EL (Unified Expression Language) jezik je dio JSTL (JavaServer Pages Standard Tag Library ).
- EL (Unified Expression Language) jezik je uključen u JSP 2.0 tehnologiju.
- Omogućava korištenje jednostavnih izraza za pristup vanjskim podatkovnim objektima iz stranice.
- JSP omogućen server izračunava i rješava izraze kako nailazi na njih i odmah vraća rezultat, ali su takvi izrazi read-only.
- JUEL (Java Unified Expression Language) je implementacija EL jezika koja je razvijena kao dio JSP 2.1 standarda ([JSR-245](#)). Sada je (od JSP 2.1) EL API odvojen u zasebni paket javax.el bez zavisnosti o osnovnim JSP klasama.



# SSI tehnologija

- SSI (*Server-Side Includes*) omogućava uključivanje instrukcija unutar HTML dokumenta na strani servera.
- Instrukcije se uključuju pomoću dodatnih HTML tagova.
- HTML dokument ima posebnu ekstenziju, .shtml, koja serveru signalizira da unutra dokumenta postoje SSI naredbe.
- Server onda parsira dokument i izvršava SSI naredbe.
- Rezultat izvršavanja SSI naredbi se onda uključuje u taj dokument i tek onda se dokument isporučuje nazad klijentu koji ga je zatražio.



# SSI tehnologija

- NCSA (*National Center for Supercomputing Applications*) je definirala format SSI naredbi kao SGML (*Standard Generalized Markup Language*) /HTML komentar.  
(<http://www.w3.org/Jigsaw/Doc/User/SSI.html>)
- Ako server ne prevede SSI naredbe u HTML kod prije no što ih isporuči klijentu SSI naredbe se neće vidjeti u pretraživaču.  
Format naredbi je:  

```
<!--#command tag1="value1" -->
```
- Skup standardnih SSI naredbi:
- config, include, echo, fsize, flastmod, exec
- <http://www.fesb.hr/~kiki/proba.shtml> (više nije podržan na marjan serveru !)

# PHP, ASP



- Princip rada sa ASP (*Active Server Pages*) i PHP stranicama je nešto drukčiji i od servelta i od CGI.
- PHP se može instalirati i da radi kao CGI, ali se uglavnom instalira kao modul web servera (najčešće Apache) tako da server sve stranice sa PHP ekstenzijom prosljeđuje tom modulu koji ih obrađuje i vraća rezultat izvršavanja.
- ASP je napravljen kao dll koji je sastavni web servera koji obrađuje stranice sa ASP ekstenzijom.