

Test iz WEB programiranja

1. Napsati prvi i poslednji red HTTP zahteva (HTTP verzija 1.1) koji se dobija kada se klikne na dugme "Posalji" (a nista se ne unese u polja) u sledecoj formi:

```
<form action=http://localhost/Proba mehod="POST">
  <input type="text" name="polje1" value="tekst1">
  <input type="text" name="polje2" value="tekst2">
  <input type="submit">
</form>
```

POST /FormServlet HTTP/1.1

...

Content length: 27

\r\n //prazan red

polje1=tekst1&polje2=tekst2

2. Šta je to "Permanent connection" u HTTP protokolu verzije 1.1?

Klijent od servera zatrazi da se konekcija ne zatvara odmah po slanju odgovora, vec da se zadrzi jos malo. U zaglavlju http zahtjeva i odgovora se stavi atribut **Connection = Keep-Alive**

3. Navesti nazive atributa u HTTP zahtevu i HTTP odgovoru, a koji omogućava praćenje sesije u HTTP protokolu.

Koristi se cookie mehanizam. Provjerava se da li se u zaglavlju http zahtjeva nalazi cookie (dakle, atribut zahtjeva je **cookie : ime = vrijednost**). Ako se nalazi, to je iz sesije, pa se ta vrijednost smijesta u Http odgovor. Atribut odgovora: **Set-Cookie** (Set-Cookie : ime = vrijednost), a opcioni atributi su: domain, expires, path. Ako se ne nalazi- znaci da se klijent spaja prvi put, pa se generise cookie i smijesta u odgovor. (Isto Set-Cookie.)

4. Čime se označava kraj HTTP zahteva u GET metodi?

Praznim redom.

(Ali ne i ako je u pitanju POST metoda! Tada je kraj zahtjeva odredjen Content lengthom.)

5. Navesti osnovnu razliku između GET I POST zahteva.

Kod GET metode podaci iz forme se nalaze u url-u, a kod POST poslije zaglavlju zahtjeva:

GET /FormServlet?tekst=abcd HTTP/1.1

...

\r\n //prazan red

POST/FormServlet HTTP/1.1

...

Content-length: 10

\r\n //prazan red

tekst=abcd

6. Navesti ključne elemente HTTP odgovora koji se koriste kod redirekcije. Dati primer takvog HTTP odgovora.

Redirekcija se svodi na slanje poruke: 302 Object moved,
i postavljanje Http odgovora: Location: nova_adresa

Primer: `HTTP/1.1 302 Object moved`
`Location: neki_novi_url`

7. Navesti klasu koja se nasleđuje prilikom pravljenja servleta i navesti metodu (sa parametrima) koja se redefiniše za POST metod.

Klasa: `HttpServlet`
metoda: `protected void doPost(HttpServletRequest request, HttpServletResponse response)`

8. Koliko objekata neke servletske klase se kreira tokom rada?

Jedan objekat.

9. Kojom metodom koje klase se kod servleta obezbeđuje redirekcija po HTTP protokolu?

klasa: `HttpServletResponse`
metoda: `sendRedirect(nova_lokacija)` //kod u javi: `sendRedirect(String location)`;
atribut: `location`

10. Čime se podešava kodna strana rezultujućeg teksta koji se prikazuje iz servleta? Dati primer za UTF-8.

Metodom `setContentType` se podesava character encoding. Parametar (atribut) `charset` definise kodnu stranu kojom ce biti kodirani svi stringovi na klijentu.

```
response.setContentType("text/html; charset=UTF-8")
```

11. Kojom klasom je implementirano praćenje sesije u servletima? Koje metode ove klase se koriste?

Klasa: `HttpSession`
metode:
[`setAttribute\(java.lang.String name, java.lang.Object value\)`](#)
[`removeAttribute\(java.lang.String name\)`](#)
[`invalidate\(\)`](#)
[`getAttribute\(java.lang.String name\)`](#)
[`getSessionContext\(\)`](#)
[`getId\(\)`](#)
...

12. Ako su cookies isključeni u browseru, kojom tehnikom se sesija može pratiti? Dati primer.

Koristi se URL Rewriting mehanizam.

<a href = <http://www.mojserver.com/catalog/index.html;jsessionid=1234/>>

13. Koji atribut form taga obezbeđuje da forma služi za upload (slanje) datoteka? Navesti ime i vrednost.

Content Type:

enctype = 'multipart/form-data'

14. U kom atributu HTTP zahteva se podešava graničnik polja (boundary) prilikom upload-a (slanja) datoteka?

Content Type

15. Navesti četiri osnovna tipa dinamičkih izraza kod JSP tehnologije.

- izrazi
- skriptleti
- deklaracije
- direktive

16. Navesti dva načina pristupa podacima sa forme u JSP tehnologiji, preko primera.

1) request.getParameter()

2) <jsp: setProperty name="user" property="username" param="username"/>

17. Čime se podešava opseg vidljivosti JavaBean komponenti na JSP stranici? Dati primer za opseg vidljivosti tipa application.

<jsp: useBean id="bean" type="BeanClass" scope="application"/>

Umjesto type može stajati i class.

BeanClass je ime java klase.

Umjesto application može stajati i request, page ili session.

18. Navesti klasu i metode koje se koriste za kontrolu toka (flow, control, dispečer) u MVC modelu 2.

Klasa: RequeatDespatcher

metode: forward, include

19. Dati primer prenosa objekta iz servleta u JSP stranu, u sesiji, (dati kod servleta i JSP stranice).

Kod servleta:

```
BeanClass value = new BeanClass(...);  
HttpSession session = request.getSession(true);  
session.setAttribute("bean", value);
```

Kod JSP stranice:

```
<jsp:useBean id="bean" type="BeanClass" scope="session"/>
```

20. U servletu je kreiran niz messages na sledeći način:

```
String [] messages = new String [] {"ok","error", "warning"}  
i dodat u session scope. Dat je deo JSP stranice na koju se servlet redirektuje  
<c:forEach var="message" items="${messages}">  
    <c:out value="${message}" />  
</c:forEach>
```

Nakon poziva JSP stranice u browseru će biti ispisano

- a. ok ok ok
- b. Ništa
- c. ok
- d. ok error warning

Bice ispisano pod **d.:**

ok error warning.