|  |  |
| --- | --- |
| **Web ispitna pitanja**  1. Napisati prvi i poslednji red HTTP zahteva (HTTP v.1.1) koji se dobija kada se klikne na dugme “Pošalji” (a ništa se ne unese u polja)  u sledećoj 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 http://localhost/Proba 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 osgovora, 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 omogucava 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 smjesta 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 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  **6. Navesti osnovnu razliku između GET I POST zahteva.**  Kod GET metode podaci iz forme se nalaze u url-u, a kod POST posle zaglavlju zahteva:  **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  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. Ako su cookies isključeni u browser, kojom tehnikom se sesija može pratiti? Dati primer.  Koristi se URL Rewriting mehanizam.  <a href = <http://www.mojserver.com/catalog/index.html;jsesionid=1234>/>  **11.Navesti četiri osnovna tipa dinamičkih izraza kod JSP tehnologije.**  -izrazi  -skriptleti  -deklaracije  -direktive  12. Kojom klasom je implementirano praćenje sesije u servletima? Koje metode ove klase se koriste?  Klasa: HttpSesion  Metode: setAttribute(java.lang.String name, java.lang.Object value)  removeAttribute(java.lang.String name), invalidate(), getAttribute(java.lang.String name), getSessionContext(), getId()  13. Koji atribut form taga obezbeđuje da forma služi za upload (slanje) datoteka? Navesti ime i vrednost.  enctype = ‘multipart/form-data’  14. 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 moze stajati i class.  BeanClass je ime java klase.  Umjesto application moze stajati i request, page ili session  15. Navesti klasu i metode koje se koriste za kontrolu toka (flow, control, dispečer) u MVC modelu 2.  Klasa: RequeatDespatcher  metode: forward, include  16. Dati primer prenosa objekta iz servleta u JSP stranu, u sesiji, (dati kod servleta i JSP stranice).  BeanClass value = new BeanClass(…);  HttpSession session = request.getSession(true);  session.setAttribute(“bean”, value);  Kod JSP stranice:  <jsp: useBean id=”bean” type=”BeanClass” scope=”session”/>  17. Osnovna razlika izmedju GET i HEAD zahteva?  Na GET zahtev, odgovor će imati telo dok na HEAD zahtev, odgovor neće imati telo nego samo heder http odgovora. HEAD uglavnom služi samo za proveru postojanja korisnika ili dokumenta.  18. Primer HTTP odgovora koji redirektuje klijenta na pera.jsp  Redirekcija se svodi na slanje poruke:  HTTP/ x.x 302 Object moved  Location: pera.jsp  19. Kojom metodom, koje klase kod servleta se obezbedjuje redirekcija po HTTP protokolu?  klasa: HttpServletResponse  metoda: sendRedirect(nova\_lokacija) //kod u javi: sendRedirect(String location);  atribut: location  20.Kako se prati sesija u JSP tehnologiji?  Pomocu cookie mehanizma ili URL Rewriting mehanizma  21. Dati primer prenosa objekta iz servleta u JSP stranu u zahtev (napisati prog kod servleta i JSP) bio jul 2014  Servlet:  User user = new User();  Request.setAttribude(“pera”, user);  JSP<jsp:useBean id=”pera” class=”User” scope=”request”/>  22. 2 tipa podataka koja se mogu dobiti Ajax pozivom?  html, json, xml, plain text  23. Koji atribut form taga obezbeđuje da forma služi za upload (slanje) datoteka? Navesti ime i vrednost.  enctype = ‘multipart/form-data’  24. 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”)  25. 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;jsesionid=1234/>  26. Navesti dva načina pristupa podacima sa forme u JSP tehnologiji, preko primera.  1) request.getParameter()  2) <jsp: setProperty name=”user” property=”username” param=”username”/>  27. U servletu je kreiran niz messages na sledeći način:  String [] massages = 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.  28. Minimalan broj atributa u HTTP zahtevu?  3  29. Kako se brisu svi objekti vezani za sesiju?  JSP:  <c:remove var=”manageuserscontroller” scope=”session”  Ili java:  request.getSession().removeAttribute(“manageuserscontroller”);  30. Kako se podacima forme pristupa uz pomoc EL tehnologija?  EL izrazima tipa ${model.atribut}  31. Kako se prilikom upload-a datoteke dobije datoteka cija velicina prelazi neku zadanu?  Content-Type: multipart/form-data  32. U cemu je razlika RedirectDispatcher.forward() i response.SendRedirect() ?  Postoji mnogo razlika izmedju ove dve metode, a u oba slucaja rezultat ce biti isti:  1. ako koristite sendRedirect(), mozete se konektovati na bilo koji URL izvan vase web aplikacije,a ako koristite forward() samo u okviru vase aplikacije  2. sendRedirect() je sporiji jer server salje URL za redirekciju klijentu, koji zatim ponovo salje upit serveru za dobijeni URL  3. ako koristite sendRedirect(), request i response necete moci da koristite na prosledjenoj stranici,dok sa forward() mozete te iste request i response da prosledite stranici  4. ako koristite sendRedirect(), URL u browser-u se menja, za razliku od forward()  5. sendRedirect() se obavlja i na klijentskoj strani, a forward() kompletno na serverskoj  6. browser ne vidi RequestDispatcher, za razliku od sendRedirect()  7. ako koristite sendRedirect(), browser moze da gadja dobijeni URL i direktno, cime nastaje problem ako je potrebno da se prvo izvrsi servlet  33. Objasni scope  Scope nam omogucava da objekat pripada odredjenom delu aplikacije. Postoje cetiri osnovna a to su: request,page,  session i aplication. Kada kazemo npr “objekat vozilo ima scope session” to znaci da objekat vozilo postoji  samo u okviru odredjene sesije.<jsp:useBean id=”greeter” class=”thePackage.Greetings” scope=”session” />  aplication - svi korisnici aplikacije dele istu instancu bean-a  session - svaki korisnik ima svoju instancu bean-a  page - svaka stranica ima svoju instancu bean-a. Ako zelimo nesto poput privatne promenljive u stranici.  request - svaki zahtev za stranicom ima svoju instancu bean-a.  Sve JSP stranice i servleti koji dele isti request dele i taj bean. Npr: ako jedan servlet forward-uje request nekoj JSP stranici, a ona uradi include neke druge JSP stranice, onda i servlet i obe JSP stranice imaju pristup tom bean-u.  34. Cookies  Postoje dve osnovne vrste cookie-a: session cookie i permanent cookie.  Session cookies se cuvaju u memoriji browsera sve dok se ne raskine veza sa serverom. Oni obicno cuvaju  ID sesije koji ne identifikuje korisnika, omogucavajuci im da se krecu od stranice do stranice bez potrebe da  se stalno loguju.  Permanent cookies se cuvaju u trajnoj memoriji web browser-a, i brisu se kad isteknu (server podesava  vreme vazenja cookie-a). Obicno sadrze podatke o korisniku odredjenog sajta, i omogucavaju da se ti podacikoriste u buducim sesijama. Mogu se koristiti za informacije o broju posetilaca, prosecnom vremenu provedenom  na stranici,..  35. UTF-8 encoding  UTF-8 koristi od jednog do cetiri bajta za kodiranje pojedinacnih karaktera. Za ascii vrednosti (ima ih 128)  prvi bit je 0 i onda sledi ascii kod (0xxxxxxx). Karakteri se dalje enkodiraju po sledecem algoritmu:  2 bajta: 110xxxxx 10xxxxxx  3 bajta: 1110xxxx 10xxxxxx 10xxxxxx  4 bajta: 11110xxx 10xxxxxx 10xxxxxx 10xxxxxx  36. Razlika izmedju MVC1 i MVC2  MVC model 1 se sastoji iz povezanih jsp stranica. Model, prezentacija i kontroler su ugradjeni u stranice. Podaci su u najboljem slucaju modelirani sa java beans. Prelazak na sledecu stranicu je odredjen klikom na hiperlink ili submit dugme. Browser salje request direktno jsp stranici, ona komunicira sa bean-ovima ivraca odgovor. U svemu ovome fale servleti, i zbog toga je jako otezano testiranje. MVC model 2 razdvaja model, prezentaciju i kontroler. Servleti imaju ulogu kontrolera, jsp stranice imaju ulogu prezentacije, a bean klase modeluju podatke. Browser se obraca servletu, koji zatim komunicira sa bean-ovima, zatim jsp stranice koriste bean-ove da bi formirale konacni response.  **37. Ako browser zatrazi stranicu pera.jsp sta ce se dogoditi u sevletskom kontejneru (JSP)?** | | |
|  |
|  | | |