



FELADATGYŰJTEMÉNY

az Internet alkalmazásfejlesztés tantárgy elsajátításához

Szerző: Fodor Szabina, PhD

2020

Budapesti Corvinus Egyetem

TARTALOMJEGYZÉK

1. Feladatgyűjteményről	3
2. Vastag kliens és vékony kliens	4
Elméleti kérdések	4
Gyakorlati feladatok	5
3. Web szerverek, JSP	6
Elméleti kérdések	6
Gyakorlati feladatok	6
4. Kliensről érkező adatok feldolgozása	7
Elméleti kérdések	7
Gyakorlati feladatok	8
5. JavaBean technológia	11
Elméleti kérdések	12
Gyakorlati feladatok	12
6. Felhasználó adatainak nyomkövetése	15
Elméleti kérdések	15
Gyakorlati feladatok	15
7. Egyedi tagek	17
Elméleti kérdések	17
Gyakorlati feladatok	17
8. XML	20
Elméleti kérdések	20
Érvényes XML dokumentumok (DTD, XSD)	21
Gyakorlati feladatok	21
XML adatok transformálása	21
Gyakorlati feladatok	21
9. Webszolgáltatások	25
Elméleti kérdések	25
Gyakorlati feladatok	26
10. Adatbáziskapcsolat	27
Elméleti kérdések	29
Gyakorlati feladatok	29
11. Komplex problémák	30

1. FELADATGYŰJTEMÉNYRŐL

Az Internet alkalmazásfejlesztés tantárgy oktatásának célja, hogy a hallgatók megismerkedjenek, és jártassági szinten elsajátítsák a Java alapú webes alkalmazások fejlesztésnek terminológiáját. Az ismertett témák között szerepel a HTTP kérések kezelése, HTTP válaszadatok létrehozása, cookie-k feldolgozása, a munkamenetek nyomon követése, az egyedi JSP tagkönyvtárak megtervezése és implementálása, valamint a tartalom elkülönítése JavaBean komponensek és az MVC keretrendszer használatával (2. modell). A kurzus sikeres elvégzése után a hallgatók képesek lesznek a trendeknek megfelelő szerver oldali webalkalmazások elkészítésére.

A szerző szándéka szerint ez a feladatgyűjtemény segíti a hallgatók szerveroldali programozáshoz szükséges gyakorlati tudásának elsajátítását. A problémák megoldása során a hallgatók elmélyítik az elméletben megszerzett tudásukat.

A feladatgyűjtemény követi a tantárgy keretében feldolgozott témaköröket, így a fejezetek a technológiához kapcsolódó fontos kérdésköröket mutatnak be úgy, mint a kliensről érkező adatok feldolgozását, egyes feladatokat megvalósító komponensek készítését, a felhasználók adatainak nyomonkövetését, egyedi tagek készítését, adatbázis kapcsolatok felépítését és az elosztott rendszerek felé mutató webszolgáltatások készítését és használatát.

A feladatok mindegyikének megoldása Netbeans projektbe csomagolva az egyetem sharepoint és moodle rendszerében elérhető. A feladatok kidolgozásához használt fejlesztési és futtatási környezet (Netbeans 11.0; Apache Tomcat 9.0; Glassfish 4.1.1) szintén innen elérhető a telepítéshez szükséges útmutatókkal.

Budapest, 2020. december 18.

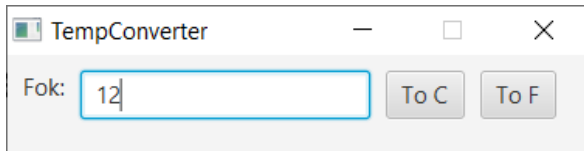
Fodor Szabina

2. VASTAG KLIENS ÉS VÉKONY KLIENS

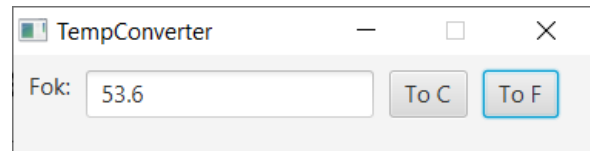
Hasonlítsa össze a minta projektben lévő Fahrenheit/Celsius hőmérséklet átváltó program vastag és vékony kliens alkalmazását.

Vastag kliens

Beolvasás

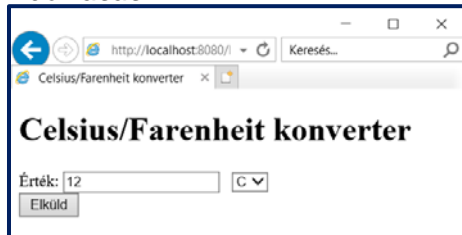


Eredmény:



Vékony kliens

Beolvasás



Eredmény:



ELMÉLETI KÉRDÉSEK

E2.1 Mire használjuk tipikusan a webservereket?

E2.2 A Microsoft Outlook asztali alkalmazás egy tipikus vastag vagy vékony kliens?

E2.3 Igazak vagy hamisak az alábbi állítások:

- A vastag kliensek nagymértékben függenek a kiszolgáló szervertől.
- Az OWASP Top 10 (<https://owasp.org/www-project-top-ten/>) kockázat általában nem alkalmazhatóak a vastag kliensekre.
- Az érzékeny adatok szöveges állományban való tárolásának legjobb megoldása a konfigurációs állományok.
- Egy tipikus vastag kliens programokat futtat és adatokat kér el az adatbázisból.
- Az egyetemi oktatásban használt Moodle alkalmazás egy tipikus web alkalmazás, amelyet vékony kliens segítségével használunk.
- A HTTP protokoll kérés/válasz információ átviteli megoldás.

E2.4 Egy on-line kereskedelmet folytató vállalkozás informatikai részlegén dolgozik. A cég elhatározza, hogy szolgáltatását mobil eszközökön is elérhetővé teszi. A cégnek döntenie

kell, hogy egy saját („hagyományos”, vastag) kliensalkalmazást készít vagy inkább mobilra optimalizált dinamikusan weboldallal („hajszálvékony kliens”) érhet-e el nagyobb sikert.

Hasonlítsa össze a két technológiai megoldás előnyeit és hátrányait és döntse el, hogy melyiket javasolná a vállalkozás vezetőjének.

GYAKORLATI FELADATOK

Gy2.1 Írjon egy egyszerű vastag klienst „HelloService” két szolgáltatásához.

Gy2.2 Írjon vastag klienst a NumberConversion SOAP szolgáltatáshoz, amely a következő linken található

<https://www.dataaccess.com/webserviceserver/numberconversion.wso>

3. WEB SZERVEREK, JSP

Nézze meg a minta projektben adott megoldást, amelyik megvizsgálja, hogy a felhasználó küldött-e „nev” nevű paraméterben valamilyen értéket. Ha igen, akkor a paraméterben küldött értéket használva a nevéen üdvözi a felhasználót, hanem akkor „Hello HTML!” üdvözetet ír ki!

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
|
| <html>
|   <head>
|     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
|     <title>JSP Oldal</title>
|   </head>
|   <body>
|     <% if (request.getParameter("nev")== null) { %>
|       <h1>Hello HTML!</h1>
|     <% } else { %>
|       <h1>Hello <%=request.getParameter("nev")%>! </h1>
|     <% } %>
|   </body>
| </html>
```

ELMÉLETI KÉRDÉSEK

E3.1 Hasonlítsa össze Javában a JSP és a szervlet technológiát. Mikor melyiket érdemes használni?

E3.2 Milyen típusú fájlok a JSP-k? Hogyan működnek?

E3.3 Mi a HTTP-n történő kommunikáció lényege? Írjon példát mikor használjuk!

E3.4 Hasonlítsa össze Javában a JSP és a szervlet technológiát? Mikor melyiket érdemes használni?

GYAKORLATI FELADATOK

Gy3.1 Készítsen egy JSP oldalt, amelyik háttérszínt vár, ha az átadott paraméter nem üres, akkor beállítja az oldalnak a megadott színt, egyébként pedig fehérre állítja az oldal színét.

Gy3.2 Készítsen jósló JSP programot, amely véletlenszerűen jósolja azt, hogy gyönyörű, illetve borzalmas napunk lesz.

Gy3.3 Készítsen egy JSP alkalmazást, amelyik a kapott szöveget megfordítja.

Gy3.4 Készítsen egy JSP programot, amely egy négy-alapműveletet tudó számológépet valósít meg.

4. KLIENSRŐL ÉRTEKEZŐ ADATOK FELDOLGOZÁSA

Nézze meg a minta projektben adott megoldást, ahol a standard JSP tagek segítségével ellenőrizzük a kliens IP címét és csak az 1-sel kezdődő IP című gépeket szolgálja ki.

```
...
<c:set var='ip' value='${pageContext.request.remoteAddr}' />
<c:choose>
  <c:when test="${fn:startsWith(ip, '1')}">
    <h1 style="color:gray">Engedélyezett hozzáférés.</h1>
  </c:when>
  <c:otherwise>
    <h1 style="color:red">Nem engedélyezett hozzáférés.</h1>
  </c:otherwise>
</c:choose>
...
```

ELMÉLETI KÉRDÉSEK

E4.1 Melyik állítás igaz?

- a) Ha az űrlapot POST metódussal küldjük, akkor szervlet/JSP nem látja a küldött paramétereket.
- b) Ha az Internet böngészőben egy URL címre kattintunk, akkor az információ POST metódussal küldjük el.
- c) GET metódus esetén a küldhető adat nagysága korlátozva van.
- d) HTTP protokoll nyolcféle metódust definiál.

E4.2 Válassza ki a helyes állításokat

Ha egy weblapon több űrlap is szerepel, akkor

- a) elég egy „elküld” gombot írni hozzájuk.
- b) minden űrlaphoz külön „elküld” gomb szükséges.
- c) Internet böngészőtől függ, hogy egy vagy több submit/elküld gombra van szükség.

E4.3 Mire használható a web.xml állomány?

E.4.4 Hol tárolhatunk könnyen inicializáló paramétereket egy Java szervlet számára?

E.4.5 Java JSP alkalmazásban miből tudjuk lekérdezni a kliens IP címét?

- a) request objektumból
- b) response objektumból
- c) pageContext objektumból, annak servletContext tulajdonságát kell lekérdezni
- d) web.xml állományból.

GYAKORLATI FELADATOK

Gy4.1 Készítsünk el egy egyszerű pizza-rendelést felvevő oldalt.

Paraméterek beállítása:

The screenshot shows a web form titled "Internet Pizza rendelés" in Microsoft Internet Explorer. The form contains the following elements:

- Input fields for "Név:" (Name) and "Cím:" (Address).
- Radio buttons for "Kiszállítás" (Delivery) and "Elvitel" (Takeaway).
- Checkboxes for pizza types: "Margherita", "Hawaiii", and "Pepperoni".
- A dropdown menu for "Pizza mérete:" (Pizza size) with "Közepes" (Medium) selected.
- A "Rendelés" (Order) button.

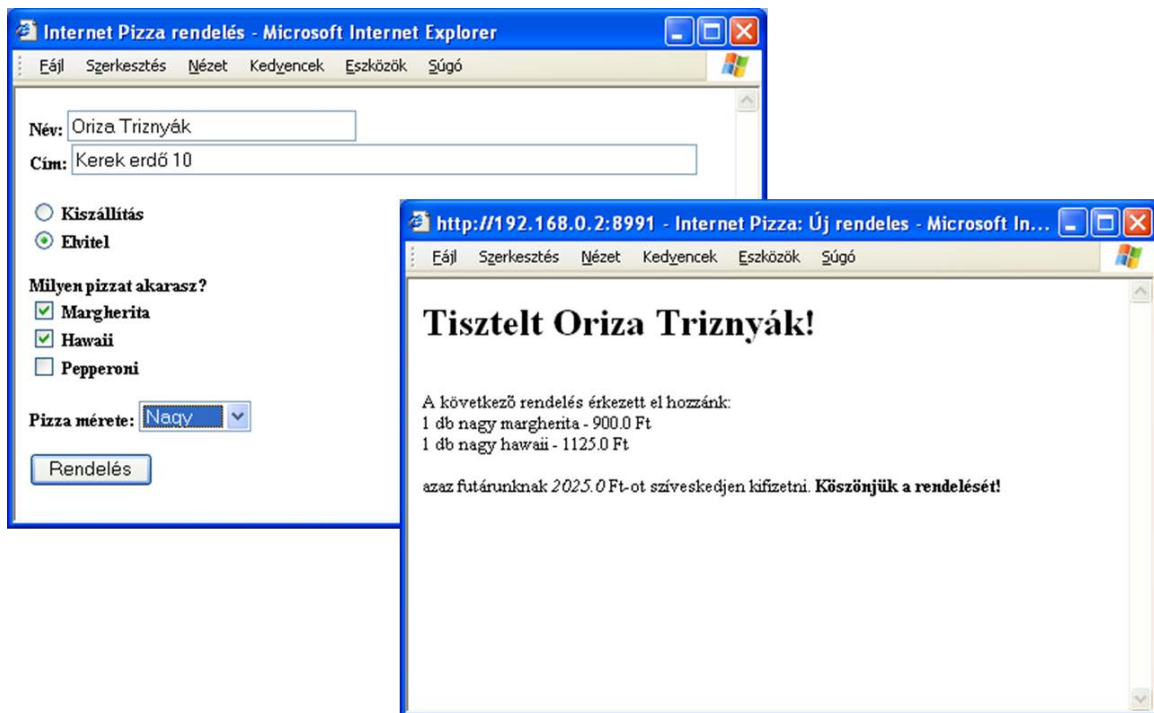
Red arrows point from text boxes to these elements, indicating the HTML parameters to be used:

- name="name" (points to the Name input field)
- name="address" (points to the Address input field)
- name="purchaseType" value="Kiszallitas" value="Elvitel" (points to the radio buttons)
- name="type" value="margherita" name="type" value="hawaiian" name="type" value="pepperoni" (points to the checkboxes)
- name="size" option="nagy" option="kozepes" option="kicsi" (points to the size dropdown menu)

A program kiírja a megkapott paraméterek nevét és értékét.

Gy4.2 Módosítsuk a pizza-rendelés programját úgy, hogy a következők szerint ki is számítsa a rendelés értékét.

- Név és cím megadása nélkül ne vegyük fel a rendelést.
- Számítsuk ki a fizetendő árat, ha
 - Margherita alapára 600 Ft,
 - Hawaiian alapára 750 Ft,
 - Pepperoni alapára 500 Ft
- a kis méretű pizza ára az alapár a közepes ennek 1.2-szerese, míg a nagy az alapár 1.5 szorososa.



Gy4.3 Módosítsuk a pizza-rendelés programját úgy, hogy a JSP GET metódus esetén írja ki a pizza-rendelés űrlapját és post metódus esetén értékelje ki azt.

Gy4.4 Készítsen egy egyszerű számológépet, amely a négy alapműveletet ismeri.

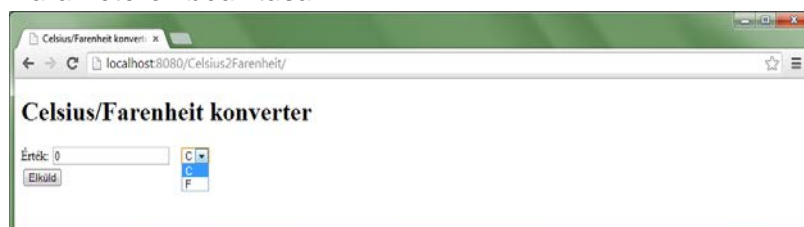
Gy4.5 Az előző feladatot egészítse ki, azzal, hogy a számológépnek két nézete van egy alap (ez az előző feladat) és egy tudományos. A tudományos nézetben lehet gyököt vonni és hatványozni is.

Gy4.6 Készítsen egy JSP alkalmazást, amely az aktuális hónapnak "megfelelő" oldallal kezdődik az 1.jpg, ...,12.jpg képek felhasználásával.

Figyeljen arra, hogy a felhasználónak nem kell az aktuális dátumot megadnia.

Gy4.7 Készítsen egy JSP 2.x programot, amely bekér egy Celsius vagy Fahrenheit fokot és átkonvertálja a másik mértékegységre.

Paraméterek beállítása:



Az eredmény:



5. JAVABEAN TECHNOLÓGIA

Nézze meg a minta projektet, ahol gépjárművek kezelésére alkalmas komponenst hozunk létre. Az autó tulajdonságai a márka, szín és a fogyasztói ár. Az AutoBean komponens ezen tulajdonságok lekérdezésére, módosítására ad lehetőséget.

AutoBean komponens váza

```
package auto;
import java.io.Serializable;
public class AutoBean implements Serializable
{
    private String marka="Volkswagen Golf";
    .....
    public AutoBean() { }
    .....
    public void setMarka(String marka) {
        this.marka= marka;
    }
    public String getMarka() {
        return this.marka;
    }
    .....
}
```

osztályváltozó

konstruktor

tulajdonság beállítása

tulajdonság lekérdezése

JSP oldal váza

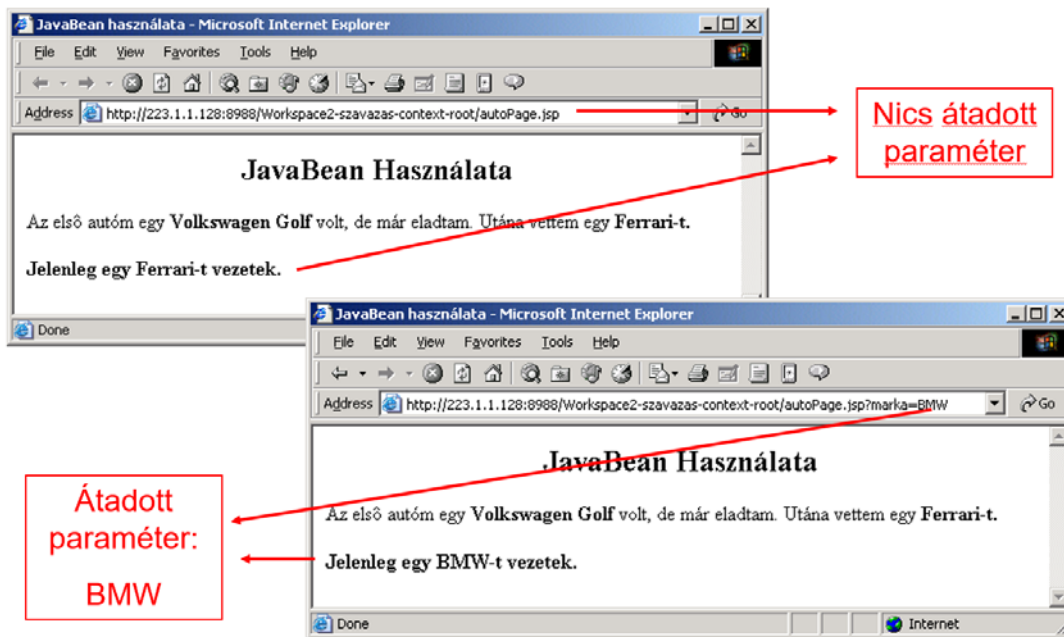
```
<html>
....
<jsp:useBean id= "enAutom"
    class="auto.AutoBean"/>
Az első autóm egy
    <jsp:getProperty name="enAutom"
        property="marka"/>, de eladtam.<br />
<jsp:setProperty name="enAutom"
    property="marka" value="Ferrari" />
Utána vettem egy <jsp:getProperty
    name="enAutom" property="marka"/>
....
</html>
```

getMarka()

setMarka(String marka)

getMarka()

A program futása



ELMÉLETI KÉRDÉSEK

E5.1 Mi az előnye JavaBeanek használatának?

E5.2 Milyen alternatívája létezik annak, hogy a Java kódot közvetlenül a HTML formázással együtt a JSP oldalba írjuk bele?

E5.3 Melyik kötelező attribútum a `<jsp:getProperty/>` tagben?

GYAKORLATI FELADATOK

Gy5.1 Módosítsuk az AutoBean.java osztályt, hogy

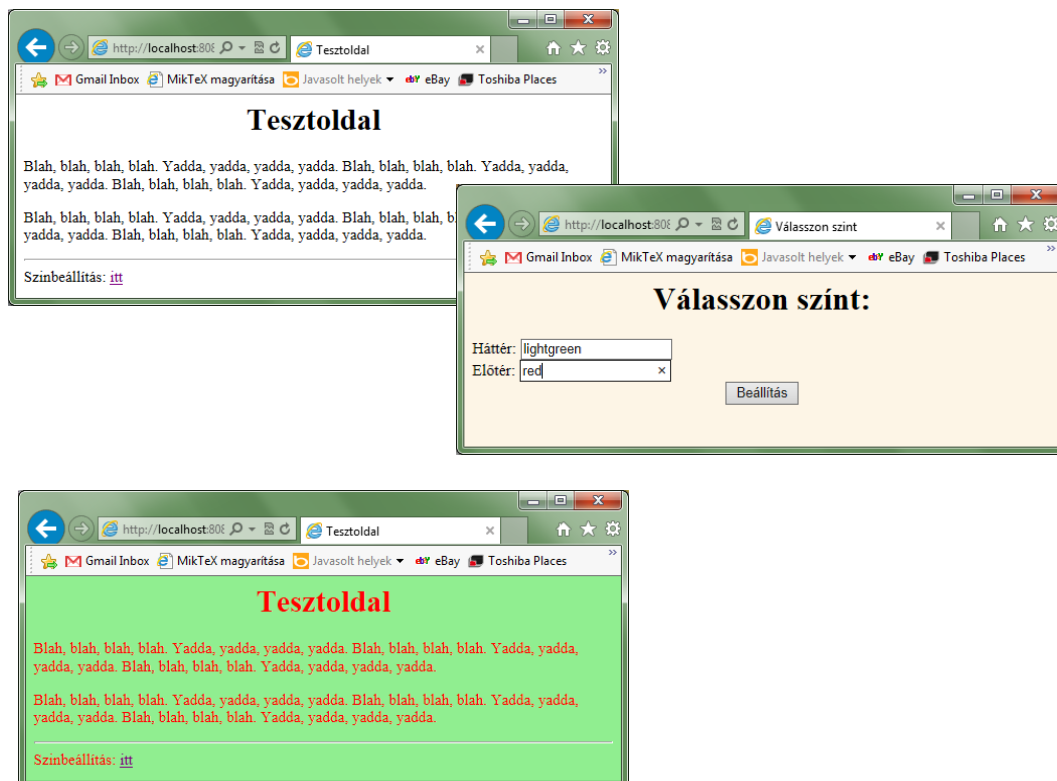
- az autó árát forintban illetve dollárban is megtudjuk adni,
- kezelni tudjuk az autó évjárátát,
- kezelni tudja az autó tulajdonosának nevét,
- ellenőrizze, hogy az ár nem lehet negatív
- ellenőrizze, hogy az évjártat 1990 és 2004 közé esik.

Gy5.2 Készítsen egy on-line gépjármű hirdetés feladására alkalmas programot. JSP oldalon keresztül lehessen új hirdetést (gépjárművet) feladni, illetve már meglévő hirdetést eltávolítani. A megoldásában használja az előző feladatban készített JavaBean komponenst.

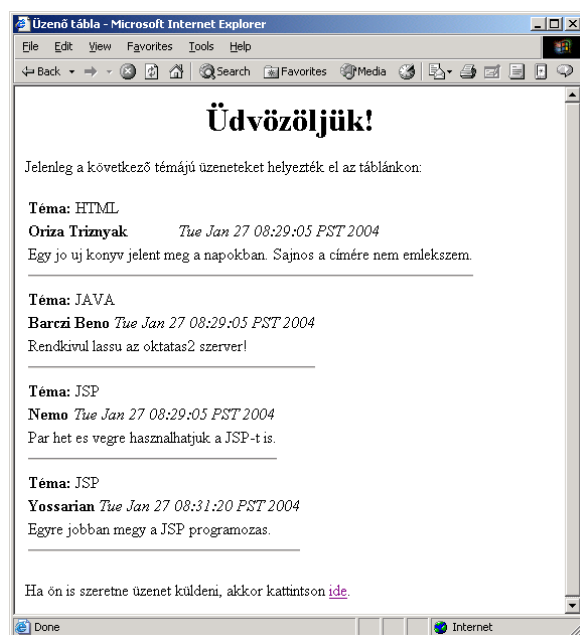
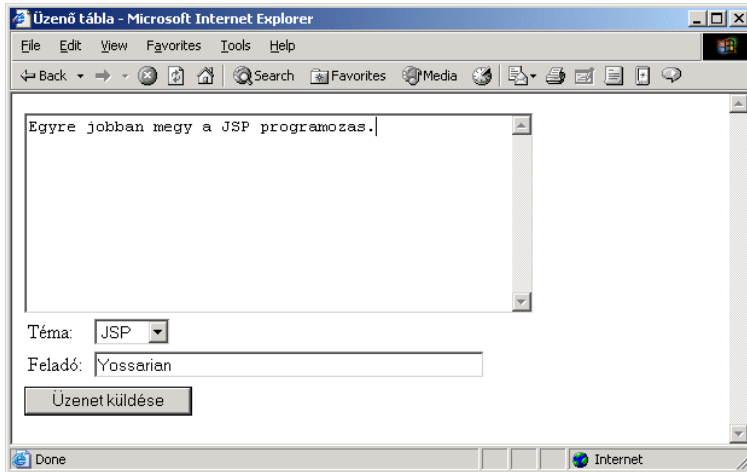
Gy5.3 Készítsen egy JavaBean komponenst és az azt használó JSP oldalt. A bab tartalmazza azt az üzleti logikát, amely megállapítja a felhasználó nevét. (Ha nincs megadva, akkor alapértelmezésként a "Világ"-t használja.) A JSP oldal pedig csak keretet adja hozzá.

Gy5.4 Készítse el az előző fejezetben elkészített számológép programot JavaBean komponens segítségével.

Gy5.5 Készítsen egy JSP alkalmazást, amely a háttérszín, előtérszín átadott paramétereket saját beállításként elmenti és alkalmazza. Ha a felhasználó a paraméterben átadja az előtér színét és a háttérszínét, akkor azt elmenti, majd egy tesztdalra irányítja a felhasználót, ahol már a beállításokat használja.



Gy5.6 Készítsen egy üzenő táblát kezelő programot, ahol üzeneteket három témában lehet elhelyezni (HTML, Java, JSP) és az elküldés után az üzenet rögtön megjelenik a táblán az elküldés idejével együtt.



6. FELHASZNÁLÓ ADATAINAK NYOMONKÖVETÉSE

Nézze meg a minta projektet, amelyben egy a bevásárló kocsit különböző módszerekkel (rejtett űrlap elemek, URL újraírás, cookie) valósítottunk meg.

ELMÉLETI KÉRDÉSEK

E6.1 Egy vállalkozás (termékcsoporthoz tetszőleges) elhatározza, hogy nyit az on-line kereskedelem felé és a tevékenységet támogató célszoftver elkészítésének koordinálásával bízza meg Önt. Az informatikai egyeztetés során dönteni kell, hogy a programban az elektronikus bevásárló kosarat milyen módon implementálják.

Tegyen javaslatot a megvalósításra és érvekkel támassza alá, hogy a meglévő egyéb technológiai lehetőségek közül Ön miért éppen az adott megoldást tartaná a legmegfelelőbbnek.

E6.2 Mi(/Mik) a piték használatának hátránya(/i)?

E6.3 Mi igaz a pitékre (cookie-ra)?

- a) Az információ név/érték párban van tárolva.
- b) Minden pitének van érvényességi ideje.
- c) Egyszerű szöveg fájlok.

E6.4 Mi igaz a Java session objektumára?

- a) Pitéket (cookie-kat) használ.
- b) URL újraírással dolgozik.
- c) Rejtett HTML űrlapmezőkkel dolgozik.
- d) Attól függ, hogy milyen Java implementációt használunk. Tomcat webszerver például pitéket használ, ha ez megengedett egyébként pedig URL újraírással dolgozik.

GYAKORLATI FELADATOK

Gy6.1 Készítsen egy JSP alkalmazást, amelyik letesz egy sütit és utána felolvassa azt.

Gy6.2 Készítsen egy JSP alkalmazást, amelyik a session objektumról kiírja a létrehozás dátumát, az utolsó hozzáférés idejét és a hozzáférések számát.

Gy6.3 Készítsen olyan JSP alkalmazást, ahol a felhasználónak be kell jelentkeznie (felhasználói név, jelszó). Sikeres bejelentkezés után a felhasználói nevet eltároljuk egy menet/session változóban és következő oldalon már személyre szóló üdvözlést küldünk (pl. Üdvözlöm <felhasználói név>!).

Gy6.4 Készítsen egy JSP alkalmazást, amely név és e-mail cím tárolására alkalmas. Ha a felhasználó bejelöli a "Megjegyezze" lehetőséget, akkor az alkalmazás 30 napig megőrzi az adatokat, amelyeket egy teszt oldalon lehet lekérdezni. (Ha nincsenek tárolt adat, akkor a Teszt oldalon a név és e-mail cím után is az "ismeretlen" szöveg jelenjen meg.

Név:	<input type="text" value="bubuka"/>
E-mail:	<input type="text" value="bubuka@valami.hu"/>
<input checked="" type="checkbox"/> Megjegyezze	
<input type="button" value="Elküld"/>	

Teszt

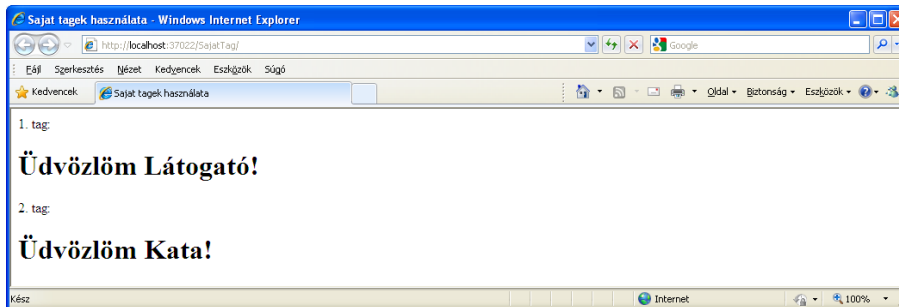
- Név: bubuka
- E-mail: bubuka@valami.hu

7. EGYEDI TAGEK

Nézze meg a minta projektet, amelyben egyedi tagek segítségével üdvözlí a felhasználót az alkalmazás.

1. tag: `<sajattagek:udvozol1/>`

2. tag: `<sajattagek:udvozol2 nev="Kata"/>`



ELMÉLETI KÉRDÉSEK

E7.1 Mi a tag bővítés előnye a JavaBean technológiával szemben?

E7.2 Mi igaz a standard JSP tag-re?

- a) Minden Java konténer támogatja
- b) A jsp: előtaggal kezdődnek
- c) A szkriptelt kód csökkentésére, illetve eltüntetésére szolgál

E7.3 Igaz-e, hogy az egyedi (felhasználó által létrehozott) tag tartalmazhat attribútumot?

GYAKORLATI FELADATOK

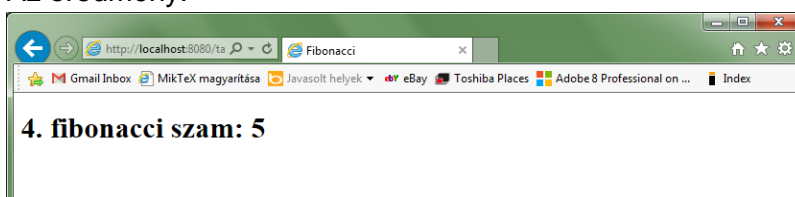
Gy7.1 Készítsen egy saját taget, amely egy attribútumban megadott n természetes számhoz kiírja az n-dik Fibonacci számot.

(Fibonacci(0)=1, Fibonacci(1)=1, Fibonacci(n-2)+Fibonacci(n-1)=Fibonacci(n))

Pl. Ha a következő módon hívjuk meg a taget:

`<sajat:fibonacci n="4"/>`

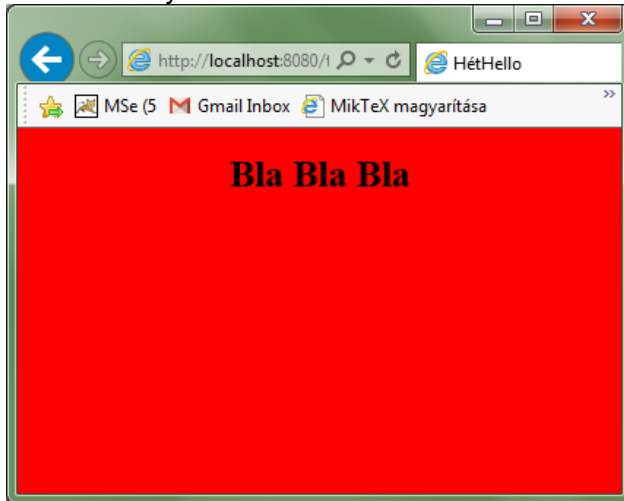
Az eredmény:



Gy7.2 Készítsen egy saját taget, amely egy személyre szabott oldallal jelentkezik be. A hét napjától függően különböző oldallal jelentkezik be. Hétfőn: kék, kedden: sárga, szerda: piros, csütörtök: zöld, péntek: fehér, szombat: pink, vasárnap: lila legyen az oldal háttere.

Pl. Ha következő módon hívjuk meg a taget (ma, szerdán):
<sajat:Hethello><h1 align="center">Bla Bla Bla</h1></sajat:Hethello>

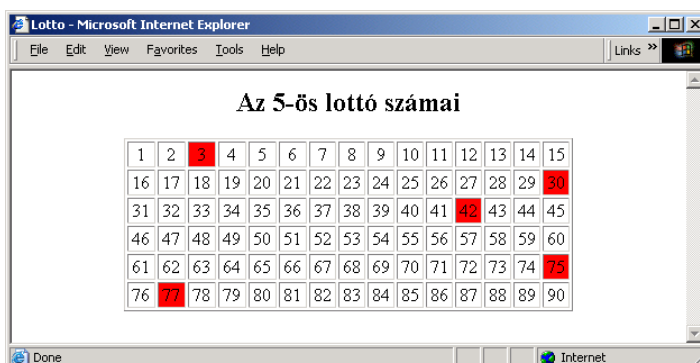
Az eredmény:



Gy7.3 Készítsen egy lottózszámokat sorsoló JSP alkalmazást, amelynek paraméterként megadhatjuk, hogy hány számból hány darabot sorsoljon ki. (Kiindulásnak használja az ötös lottó számait sorsoló lotto5.jsp.)

A tag használata ötös lottó esetén:

```
<h2 align=center> Az 5-ös lottó számai </h2>  
<p><tartalom:lottoTag tipus="90" darab="5" />
```



A tag használata hatos lottó esetén:

```
<h2 align=center> A 6-os lottó számai </h2>  
<tartalom:lottoTag tipus="45" darab="6"/>
```



8. XML

Nézze meg a minta projektet, amelyben leltar.xml állomány adatait érvényesítő leltar.dtd és leltar.xsd állományok mellett a leltar.xsl állomány található, ami az ábrán látható formára alakítja az xml adatokat.



The screenshot shows a web browser window titled 'Könyvleltár - Microsoft Internet Explorer'. The browser's address bar and menu bar are visible. The main content area displays a table with the following data:

Cím	Szerző	Kötés	Ár
Huckleberry Finn kalandjai	Mark Twain	olcsó kiadás papírkötés	549 Ft
Fűszálak	Walt Whitman	keménykötés	775 Ft
Az Álmos völgy legendája	Washington Irving	olcsó kiadás papírkötés	295 Ft
Derüvölgy románca	Nathaniel Hawthorne	cérmakötés	1095 Ft
Moby-Dick	Herman Melville	keménykötés	995 Ft
Egy hölgy arcképe	Henry James	olcsó kiadás papírkötés	495 Ft
A skarlát betű	Nathaniel Hawthorne	cérmakötés	425 Ft
A csavar fordul egyet	Henry James	cérmakötés	335 Ft

ELMÉLETI KÉRDÉSEK

E8.1 Igaz-e, hogy egy XSLT stylesheet-tek több template-t tartalmazhatnak?

E8.2 A DTD az XML dokumentum elején lehet deklarálni, vagy beilleszthető egy _____

- a) lokális hivatkozásként
- b) külső hivatkozásként
- c) lokális vagy külső hivatkozásként
- d) az állítás nem igaz, mert a DTD-t az XML fájl elején kell megadni.

E8.3 Mi az alapértelmezett sorrendje az `xsl:for-each` utasítással feldolgozott csomópontoknak?

E8.4 Írja le mire használhatóak a DTD (Document Type Definition) illetve az XSD (XML Schema) formátumú állományok. Hasonlítsa össze őket, és írjon le 3 különbséget.

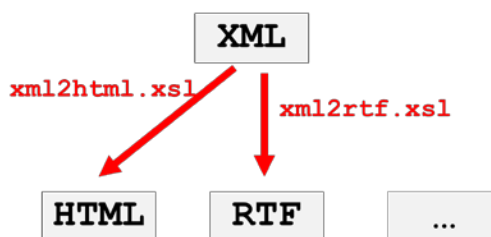
GYAKORLATI FELADATOK

Gy8.1 A note.xml állományhoz készítsen az adatokat érvényesítő dtd és xsd állományt.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
<note>
  <to>Julia</to>
  <from>Romeo</from>
  <heading>Emlekezteto</heading>
  <body>Ne felelts el a hetvegen!</body>
</note>
```

XML ADATOK TRANSFORMÁLÁSA

Nézze meg a minta projektet, amelyben data.xml állományt a xml2html.xsl állomány HTML formátumra a xml2rtf.xsl állomány pedig RTF formátumra konvertál.



GYAKORLATI FELADATOK

Gy8.2 Készítsen egy XSL állományt konyvek.xsl néven, ami a megadott konyvek.xml állományt a következő módra alakítja át HTML formátumúra.

Cím	Szerző	Kötés	Terjedelem (oldal)	Ár
Huckleberry Finn kalandjai	Mark Twain (született: 1835)	olcsó kiadás papírkötés	298	549 Ft.
Derűvölgy Románca	Nathaniel Hawthorne (született: 1804)	cérnakötés	473	1095 Ft.
Moby-Dick	Herman Melville (született: 1819)	keménykötés	724	995 Ft.

Gy8.3 Készítsen egy XSL állományt konyvek.xml néven, ami a megadott konyvek.xml állományt a következő módra alakítja át HTML formátumúra.

Cím	Szerző	Kötés	Terjedelem (oldal)	Ár
Fűszálak	Walt Whitman (született: 1819)	keménykötés	462	775 Ft
Egy jenki Arthur király udvarában	Mark Twain (született: 1835)	olcsó kiadás papírkötés	385	549 Ft

Gy8.4 Készítsen egy XSL állományt gyujtemeny.xml néven, ami a megadott gyujtemeny.xml állományt a következő módra alakítja át HTML formátumúra.

Gyűjtemények

CD-k

Beethoven D-dúr hegedűverseny
1495 Ft

Mozart Hegedűversenyek 1, 2, 3
1649 Ft

Könyvek

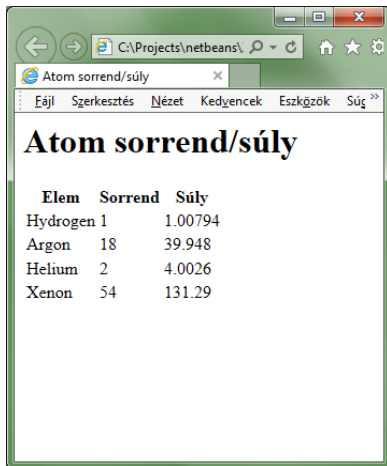
Mark Twain Huckleberry Finn kalandjai
549 Ft
bent

Walt Whitman Fűszálak
775 Ft
kint

Washington Irving Az Álmos völgy legendája
295 Ft
kint

Nathaniel Hawthorne Derűvölgy románcja
1095 Ft
bent

Gy8.5 Készítsen egy XSL állományt, ami a megadott periodic.xml állományt a következő módra alakítja át HTML formátumúra. A gázok a periódusos rendszer szerint rendezve jelenjenek meg.



Gy8.6 Készítsen egy XSL állományt, ami a megadott etlap.xml állományt a következő HTML formátumra alakítja. (Az étlap angol nyelvű legyen.)



Gy8.7 Készítsen egy XSL állományt, ami a megadott etlap.xml állományt a következő HTM formátumra alakítja. (Az étlap magyar nyelvű legyen.)



The screenshot shows a web browser window titled "A'la Carte Corvinus" displaying a menu. The menu is organized into three sections: "Előételek" (Appetizers), "Levesek" (Soups), and "Tészták" (Pasta). Each item is listed with its name and price in Hungarian Forint (Ft).

Category	Item Name	Price (Ft)
Előételek	Citromleveses tonhal saláta	1450 Ft
	Cuékés csirkesaláta	1350 Ft
	Pácolt feta sajg fűsz salátájgon	1150 Ft
	Friso kevert saláta vöröshalásó öntetel	700 Ft
Levesek	Újász trükkösleves	650 Ft
	Osztóleves csipetkésel csirkesaláta	750 Ft
	Osztóleves csipetkésel bográcsleves	850 Ft
	Hálsólt pórtkésel, belsőséggel	850 Ft
	Hálsólt pórtkésel, belsőséggel	1150 Ft
	Túrósleves pórtkéses leves	750 Ft
	Hálsólt málsó kérésleves	750 Ft
Tészták	Jálsólt sztrápasóka kolozsvári öntetelrel	1450 Ft
	Spagetti bácsórel, vöröshalásóval	1150 Ft

9. WEBSZOLGÁLTATÁSOK

Nézze meg a minta projektet, amely egy SOAP Celsius/Fahrenheit hőmérséklet konvertálás végző webszolgáltatást használó JSP alkalmazás.

```
<c:otherwise>
  A ${param.fok} ${param.tipus} =
  <% double valasz=0.0;
  try {
    intalk.TempConverterEndpointService service = new intalk.TempConverterEndpointService();
    intalk.TempConverterEndpoint port = service.getTempConverterEndpointPort();

    if (request.getParameter("tipus")!=null &&
        request.getParameter("tipus").equals("C")) {
      intalk.CelsiusToFahrenheitRequest celsiusToFahrenheitRequest =
        new intalk.CelsiusToFahrenheitRequest();
      celsiusToFahrenheitRequest.setTemperatureInCelsius(Double.parseDouble(request.getParameter("fok")));
      intalk.CelsiusToFahrenheitResponse result = port.celsiusToFahrenheit(celsiusToFahrenheitRequest);
      valasz = result.getTemperatureInFahrenheit();
    } else {
      intalk.FahrenheitToCelsiusRequest fahrenheitToCelsiusRequest = new intalk.FahrenheitToCelsiusRequest();
      fahrenheitToCelsiusRequest.setTemperatureInFahrenheit(Double.parseDouble(request.getParameter("fok")));
      intalk.FahrenheitToCelsiusResponse result = port.fahrenheitToCelsius(fahrenheitToCelsiusRequest);
      valasz=result.getTemperatureInCelsius();
    }
  } catch (Exception e) {}
  %>
  <%= valasz%>
</c:otherwise>
```

ELMÉLETI KÉRDÉSEK

E9.1 Melyik protokollt használják a RESTful webszolgáltatások az ügyfél és a kiszolgáló közötti kommunikációhoz?

E9.2 Igaz-e, hogy a webes szolgáltatások egy adott programozási nyelvhez kötöttek?

E9.3 A WSDL az alábbiak közül melyikből származik?

- a) REST és SOAP
- b) REST és NASSL
- c) SOAP és NASSL
- d) SOAP és HTML

E9.4 A webes szolgáltatások az alábbiak közül melyikre épülnek?

- a) program és program közötti interakció
- b) ember és program közötti interakció

Gy9.1 Készítsen egy szervlet, vagy JSP alkalmazást, ami a <http://www.learnwebservices.com/services/hello?WSDL> linken elérhető webszolgáltatást használva.

Gy9.2 Készítsen egy Java nyelvű webszolgáltatást és az azt használó klienst. A szolgáltatás egy elküldött értékhez számítsa ki a négyzet gyökét. Ha negatív értéket adnak meg, akkor a megfelelő komplex számot adja vissza. ($i^2 = -1$)

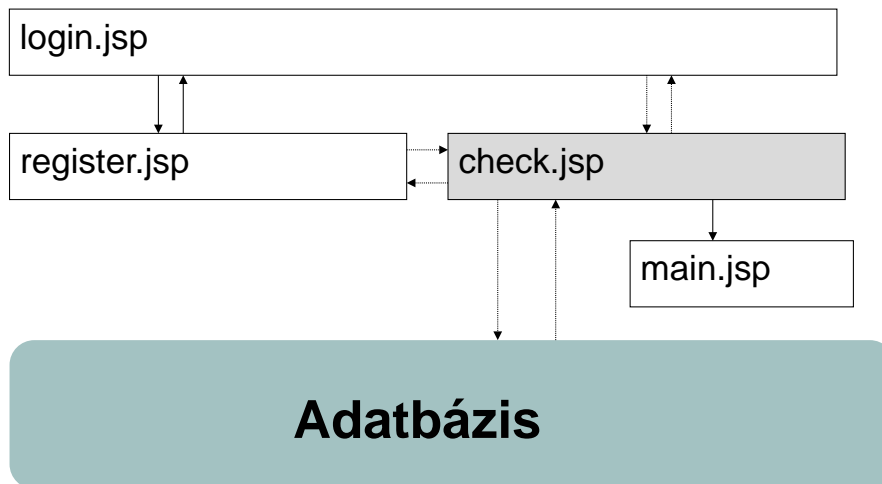
Gy9.3 Készítsen egy Java nyelvű webszolgáltatást és az azt használó klienst. A szolgáltatás két egész számot vár egy értéket és egy számrendszert. Adott értéket az adott számrendszerbeli számként fogja fel és 10-es számrendszerbelivé átváltja *Pl. 11 (érték) és 7 esetén a válasz 8.*

Gy9.4 Készítsen egy Java nyelvű webszolgáltatást és az azt használó klienst, amelyik egy küldött szöveget megfordít.

10. ADATBÁZISKAPCSOLAT

Nézze meg a minta projektet, amely egy egyszerű beléptető rendszer, amelyben regisztrált felhasználók jelszó megadása után egy védett oldalra kerülnek. Az alkalmazás lehetőséget ad új felhasználóknak regisztrációjára is.

Beléptető rendszer váza



A login.jsp oldal

```
<html>
...
<body>
  <sql:setDataSource
    var="felhasznalok"
    driver="org.apache.derby.jdbc.ClientDriver"
    url="jdbc:derby://localhost:1527/MINTA"
    scope="session"
    user="scott"
    password="password"
  />
  <h1>Üdvözöljük! Kérjük jelentkezzen be!</h1>
  <c:if test="${!empty param.errormsg}">
    <font color="red">${errorMsg}</font>
  </c:if>

  <form action="check.jsp" method="POST">
    Felhasználó név: <input type="text" name="userid" value="" size="15" /><p/>
    Jelszó: <input type="password" name="password" value="" size="15" /><p/>
    <input type="submit" value="Bejelentkezés" /><p/>
  </form>
  <p/>
  <a href="registration.jsp">Regisztráció</a>
  <c:if test="${!empty felhasznalok}">
    <font color="red">${felhasznalok}</font>
  </c:if>
</body>
</html>
```

A check.jsp oldal az adatbázis-kapcsolatot használó része

```
<@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<@taglib prefix="sql" uri="http://java.sun.com/jsp/jstl/sql"%>
<!--...-->

<@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <body>
    <sql:query var="lekerdezes" dataSource="${felhasznalok}">
      SELECT * FROM users where userid='${param.userid}' and password='${param.password}'
    </sql:query>

    <c:choose>
      <c:when test="${lekerdezes.rowCount ne 0}">
        <jsp:forward page="main.jsp"/>
      </c:when>
      <c:otherwise>
        <jsp:forward page="login.jsp">
          <jsp:param name="errorMsg" value="Hibás jelszó/vagy felhasználói név"/>
        </jsp:forward>
      </c:otherwise>
    </c:choose>
  </body>
</html>
```

A register.jsp oldal az adatbázis-kapcsolatot használó része

```
<@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<@taglib prefix="sql" uri="http://java.sun.com/jsp/jstl/sql"%>
<!--...-->

<@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <body>
    <c:choose>
      <c:when test="${request.method eq \"POST\"}">
        <sql:update var="eredmeny" dataSource="${felhasznalok}">
          INSERT INTO users (userid, password, name) VALUES ('${param.userid}', '${param.password}', '${param.name}')
        </sql:update>

        Sikeres regisztrált. Bejelentkezéshez kattintson <a href="login.jsp">ide</a>.
      </c:when>
      <c:otherwise>
        <h1>Regisztráció</h1>
        <form action="registration.jsp" method="POST">
          Felhasználó név: <input type="text" name="userid" value="" size="15" /><p/>
          Jelszó: <input type="password" name="password" value="" size="15" /><p/>
          Név: <input type="text" name="name" value="" size="15" /><p/>
          <input type="submit" value="Regisztráció" /><p/>
        </form>
      </c:otherwise>
    </c:choose>
  </body>
</html>
```

ELMÉLETI KÉRDÉSEK

E10.1 Mi történik a táblázatban lévő adatokkal, ha egy helyes SQL parancsot használunk egy tábla törlésére?

E10.2 Hogyan lehet JDBC segítségével egy adatbázis létrehozni?

E10.3 Mi(k) kell(nek) JDBC kapcsolat kiépítésére egy JSP oldalon? (Gondoljon az órán használt megoldásra)

E10.4 Mit jelent, ha "No Suitable Driver" hibaüzenetet kapunk a program használatakor?

GYAKORLATI FELADATOK

Induljon ki a kezdő JDBC-starter projektből, ez már rendelkezik a derby JAR fájljal és néhány inicializáló kóddal, amely létrehozza az „alkalmazottak” adatbázist.

```
(private String tableName = "employees";  
private String password = "somepassword";  
private String dbName = "myDatabase";  
private String url = "jdbc:derby:" + dbName;  
private String username = "someuser";)
```

Employee tábla:

<i>int employeeID</i>	<i>String firstName</i>	<i>String lastName</i>	<i>String position</i>	<i>int salary</i>
-----------------------	-------------------------	------------------------	------------------------	-------------------

Gy10.1 Készítsen egy JSP alkalmazást, amely az összes alkalmazott vezetéknevének felsorolt listáját jeleníti meg.

Gy10.2 Készítsen egy JSP alkalmazást, amely kiszámítja az összes alkalmazott fizetésének összegét.

Gy10.3 Készítsen egy HTML űrlapot, amely összegyűjti a vezetéknevet. Küldje el a nevet egy JSP-nek oldalnak. Ha van ilyen vezetéknevvvel rendelkező alkalmazott, mutassa meg vele a teljes részleteket (csak mutassa meg az első alkalmazottat, ha több ember van azonos néven). Ha egyetlen ilyen vezetéknevvvel rendelkező alkalmazott sincs, akkor írja ki azt.

E11.1 Egy egyetem a pályázati folyamatait szeretné automatizálni egy alkalmazás segítségével. Az elképzelések szerint az új alkalmazás egy könnyen kezelhető, grafikus felületen keresztül lehetővé tenné új pályázatok indítását, folyamatban lévő pályázatok nyomon-követését. Az alkalmazást különböző szerepkörökkel lehetne használni, mint pályázó, a pályázatot elbíráló, illetve adminisztrátor. Elvárásként fogalmazódik meg, hogy a pályázó csak a saját pályázatait láthassa és módosíthassa; a bíráló a hozzárendelt pályázatok anyagait láthassa és véleményezhesse, az adminisztrátor pedig a pályázatokról összefoglaló riportokat is készíthessen (pl. éves riportot arról, hogy melyik tanszék hány pályázatot összesen milyen költséggel nyújtott be). Az alkalmazástól elvárás még, hogy az adatok és riportok táblázatkezelőbe (pl. MS Excel) és szövegszerkesztőbe (pl. MS Word) exportálhatók legyenek.

Önt ennek a szoftverfejlesztési projektnek a koordinálásával bízták meg. Tegyen javaslatot, hogy milyen hálózati megoldást javasolna a vezetőknek és indoklásában térjen ki olyan kérdésekre, mint

- a) adatbiztonság,
- b) különböző nyelvű felhasználói felületek készítése,
- c) az alkalmazás asztali és mobil eszközökről való használhatósága.

E11.2 Egy multinacionális élelmiszer áruházlánc új funkcióval szeretné kiegészíteni a jelenleg is működő fenntartási munkákat segítő Java alapú, webes alkalmazását. Lehetőséget szeretnének kapni arra, hogy folyamatosan monitorozzák a hűtők állapotát továbbá a rendszer automatikusan sürgősségi értesítést küldjön, ha hűtők hőmérséklete bizonyos határérték fölé nő. A hűtőket jelenleg egy Raspberry Pi-n futó alkalmazással lehet vezérelni, amely a hűtők hőmérsékletét lekérdezi és a mért adatokat egy adatbázisban eltárolja.

Önt azzal bízzák meg, hogy a két rendszer integrálására javaslatot tegyen, illetve a javasolt fejlesztési munkát irányítsa.

Milyen megoldást javasolna a megvalósításra? Miért? Válaszában gondoljon arra, hogy a már meglévő (és jól működő) folyamatokat minél kevésbé kelljen módosítani.