

Java technológiák - 1. előadás

ANTAL Margit

Sapientia - EMTE

2010

- Antal Margit, Java alapú webtechnológiák, Scientia, 2009.
- [Sun] Sun Microsystems, Web Component Development with Servlet and JSP Technologies. SL-314, 2007.
- [Hunter] Jason Hunter, Java szervletek programozása, Kossuth kiadó, 2002.
- [Bergsten] Hans Bergsten, Java Server Pages, Kossuth kiadó, 2001.
- [Crawford] William Crawford, Jonathan Kaplan, J2EE Design Patterns, O'Reilly, 2003.

- Java webtechnológiák: JavaServlet (szervletek), JSP lapok, JSTL elemkönyvtárak, Struts keretrendszer.
- Java technológiák:
 - Java Naming and Directory Interface (JNDI) - névszolgáltatás
 - Java Authentication and Authorization Service (JAAS) - biztonság
 - Webszolgáltatások: SOAP, REST
 - Java Persistence API (JPA) - perzisztencia
 - Tesztelés - JUnit keretrendszer
 - Spring keretrendszer
 - Build eszközök: Ant, Maven

1. előadás célja

- Webalkalmazások bemutatása.
- A Java EE 5 platform bemutatása.
- Webalkalmazások működése.
- A Java szervletek rövid bemutatása.
- JSP technológia rövid bemutatása.
- A háromrétegű webalkalmazások bemutatása.
- MVC architektúra bemutatása.

Java:

- programozási nyelv - JDK - Java Development Kit
- futtatási környezet - JRE - Java Runtime Environment

Java platformok:

- J2SE - alap Java,
- J2EE - nagyvállalati alkalmazások,
- J2ME - mobilalkalmazások.

- C++: Make (Makefile: függőségek, parancsok, ...)
- Java:
 - Ant: XML konfigurációs állományok
 - Maven: Project Object Model - pom.xml; repository a jar csomagoknak; függőségek automatikus letöltése

- alkalmazások komponens alapú
 - tervezése – design,
 - fejlesztése – development,
 - összeállítása – assembly,
 - telepítése – deployment.
- többretegű osztott alkalmazásmodell,
- újrafelhasználható komponensek,
- egységesített biztonsági modell,
- tranzakcióvezérlés,
- webszolgáltatások,
- platformfüggetlenség,
- szerverfüggetlenség.

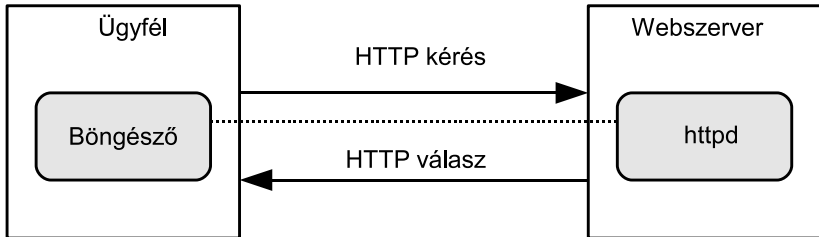
Hypertext Transfer Protocol

Egy kapcsolat (Connection) \Rightarrow Egy kérés (Request)

- állapotmentes protokoll,
- hatékony,
- bármilyen fájl (MIME) átvitele megvalósítható.

Hypertext Markup Language

Olyan speciális nyelv, amely parancs-, illetve a szöveg megjelenítésére vonatkozó formázási tagokat tartalmaz.



- a böngésző egy erőforrást kér a szervertől - általában sima szöveg,
- a szerver elküldi a kért erőforrást - részben szöveg, részben bináris (pl. képek).

Webszerverek és kliensek

Cég	Szerver	Kliens
Apache	httpd	
Netscape	Netscape Server	Netscape Navigator
Microsoft	IIS	Internet Explorer
Mozilla		Firefox

Webhely

HTML lapok és más médiafájlok gyűjteménye, amelyek esetenként egy komplex könyvtárszerkezetet alkotnak.

URL-Uniform Resource Locator

protocol://host:port/path/file

URL példa

<http://www.soccer.org:80/league/Spring2001.html>

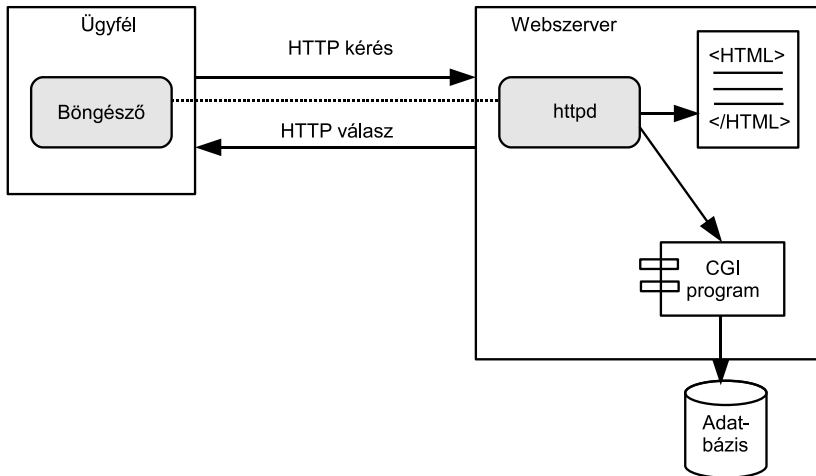
- webhely: statikus állományok gyűjteménye
- webalkalmazás: webhely+dinamikus funkcionalitás
- webszolgáltatás: olyan szolgáltatás, amelyet alkalmazások egymásnak kínálnak a világhálón

- szerveroldalon futtatott alkalmazás: szkript vagy végrehajtható állomány,
- a szerver külön folyamatként futtatja.

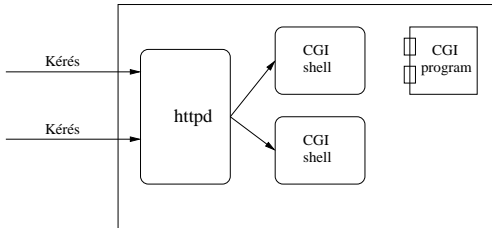
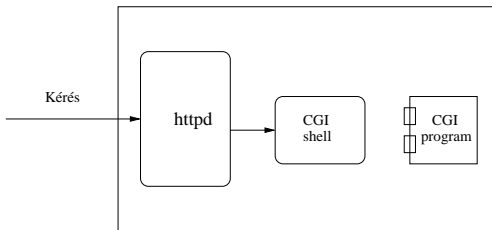
Webalkalmazás

webalkalmazás = webhely + CGI program

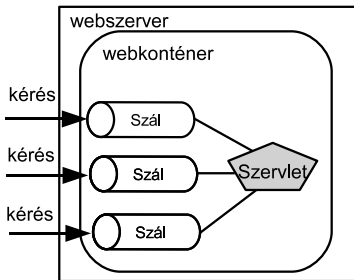
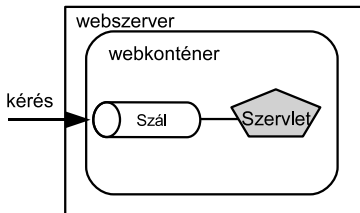
CGI program



CGI programok



- Előnyök:
 - bármilyen programozási nyelvben elkészíthető (Perl a legelterjedtebb),
 - hibás CGI program nem befolyásolja a webservert működését,
 - kényelmes a hívása - egy sor a weboldalon,
 - nincs konkurencia probléma - minden CGI program saját burokból fut,
 - minden szolgáltató támogatja.
- Hátrányok:
 - időigényes – a burok létrehozása miatt,
 - nem skálázható,
 - a CGI programnyelvek nem mindig biztonságosak,
 - a szkriptnyelvek néha platformfüggők is lehetnek.



- Előnyök:
 - gyorsabb mint a CGI,
 - skálázható,
 - hibatűrő \Leftarrow Java nyelvben íródott,
 - platformfüggetlen \Leftarrow Java nyelvben íródott,
 - naplózási lehetőség.
- Hátrányok:
 - keveredik a Java kód (logika) HTML kóddal (megjelenítés) \Rightarrow a weboldal módosítása, maga után vonja a szervlet újraforítását és telepítését,
 - konkurencia problémákat kell kezelni.

- a szervletek a szervletkonténerben futnak (szervletmotornak is nevezik)
- webkonténer = szervletkonténer = JVM + Servlet API

Java szervletek

Java kódba beágyazott HTML kód

JavaServer Pages – JSP

HTML oldalba beágyazott Java kód

- a JSP oldal szervletként fut,
- ami szervletként megvalósítható az megvalósítható JSP-ben is,
- a JSP a prezentációs részre fókuszol.

Egyszerű JSP lap

```
<table>
<% for( int i=0; i<10;++i){ %>
  <tr><td>
    <%=i%><%=i*i%><br>
  </td></tr>
<% } %>
</table>
```

- JSP – Java Server Pages
- PHP – HyperText Preprocessor
- ASP – Active Server Pages
- Ruby on Rails

- Előnyök:
 - hatékonyság (szálak),
 - platformfüggetlenség,
 - kényelmesség (a Java API használatára épül).
- Hátrányok:
 - nehézkes hibakeresés,
 - az alkalmazási logika és prezentáció nehezen választható szét,
 - konkurencia-kezelés.

Martin Fowler

Egy darab szoftver, amelyet változtatás nélkül használnak az alkalmazások.

- önálló logikai funkciót töltenek be,
- kényelmes interfészen keresztül csatlakoztathatók,
- újrafelhasználhatók.

Jellemzők:

- Java nyelven íródtak,
 - eleget kell tenni az illető komponensre vonatkozó J2EE specifikációnak.
-
- appletek – kliensoldalon futó komponensek,
 - szervletek és JSP – szerveroldalon, webkonténerben futó komponensek,
 - EJB (Enterprise JavaBeans) - üzleti logika komponensek, J2EE szerver EJB-konténerében futnak.

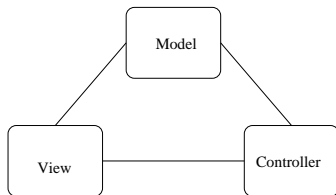
Cél: az alkalmazás rugalmasságának növelése

- kliens
- üzleti logika
- adat

- Model – Modell - Az adatok, illetve az ezeknek megfelelő adatstruktúrák, objektumok.
- View – Megjelenítés - A modell megjelenítése.
- Controller – Vezérlés - Események, amelyek hatására tevékenységet kell végezni a modellben.

Megjegyzés

A három rétegű architektúra \neq MVC



- 1. Modell: A kérést egy JSP lap vagy egy szervlet feldolgozza, elvégzi az adatok érvényesítését, kezeli az alkalmazás logikáját és előállítja a kérésre a választ
- 2. Modell: Célja a feladatok jobb szétválasztása;
Megvalósítás: MVC architektúra;
 - **Model**: Java osztályok, megvalósítják az alkalmazás logikáját.
 - **View**: JSP lap, előállítja a választ HTML formátumban; szükség esetén kommunikál a modellel adatok elérése céljából; az előállított űrlap megteremti az interaktivitás lehetőségét.
 - **Controller**: A vezérlést egy szervlet végzi. Feladata a beérkező adatok ellenőrzése, a modell és a megjelenítés frissítése.

Modell 2. típusú létező keretrendszerek

- Struts – Jakarta
- JavaServer Faces – Sun
- Velocity – Apache

- EJB konténer – nagyvállalati babok futtatása – a Java EE szerveren fut.
- webkonténer – JSP lapok és szervletek futtatása – a Java EE szerveren fut.
- kliens alkalmazás-konténer – kliens komponensek futtatása – kliensoldalon fut.
- applet-konténer – appletek végrehajtása – Java kompatibilis böngésző kliensoldalon.

- Java Servlet 2.5,
- JavaServer Pages 2.1,
- JavaServer Pages Standard Tag Library,
- JavaServer Faces 1.2.

- webdesigner: megjelenítés rész: HTML + JSP,
- webkomponens fejlesztő: szervletek + JSP lapokhoz szükséges osztályok,
- üzleti komponensfejlesztő:
 - web szint: Java osztályok vagy Java babok (JavaBeans),
 - EJB szint: EJB komponensek.
- adatelérési komponensfejlesztő: perzisztenciát biztosítanak a modellelemeknek.

- webcentrikus:
 - statikus HTML,
 - HTML alapvető szervletekkel és JSP lapokkal,
 - JSP JavaBeans komponensekkel.
- Java EE centrikus: Strukturált webalkalmazások moduláris komponensekkel és nagyvállalati babokkal (Enterprise Bean).

- CGI,
- Java szervletek,
- JSP lapok,
- J2EE architektúra.