Java – technologie zaawansowane

Wykład 1
Java EE - wstęp

Instytut Fizyki i Informatyki Stosowanej
Uniwersytetu Łódzkiego 16.02.2008
Łódź

Rozkład przedmiotu

- Wykład
  - 15 godzin
    - Ogólne założenia i podstawy
- Laboratorium
  - 45 godzin
    - Praktyczne
      - Narzędzia
        - » Ecllipse
        - » Tomcat, GlassFish, JBoss

O czym te zajęcia

- Java EE
  - Tylko (znamy już SE i ME ☺)
    - Serwerowe zastosowania Javy
      - » Strony webowe (jsp, jsf i inne frameworki)
      - » Web Serwisy (Usługi Webowe)
    - Aplikacje klienckie do wykorzystania Web Serwisów
    - Współpraca z bazami danych
      - » JDBC
      - » Hibernate
    - Ciekawe frameworki np. Spring
    - Enetrprise Java Beans (EJB)

Literatura

- Java EE 5.0 Specification (strona Sun'a)
  - Referencje z tego dokumentu
- Marty Hall, Larry Brown, Java Servlet i JavaServer Pages.
- Robert Bruner, Java w komercyjnych usługach sieciowych. Księga eksperta
- Dave Minter, Jeff Linwood, Hibernate od Nowicjusza do Profesjonalisty
- Cay Horstmann, Gary Cornell, Core Java 2, Techniki zaawansowane

Aplikacje JEE

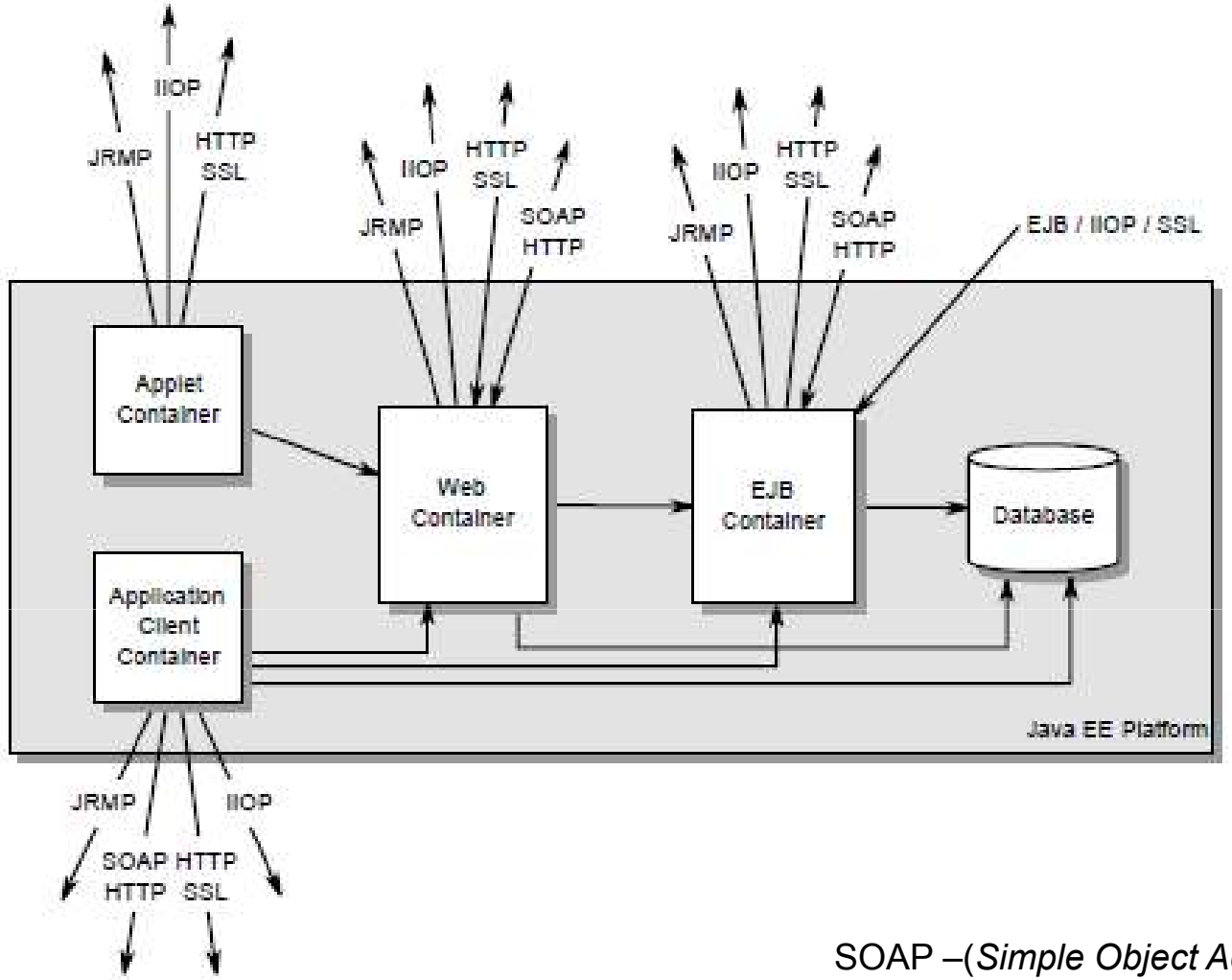
- Programy uruchamiane na:
  - Kontenerach Aplikacji
    - Tomcat
  - Serwerach Aplikacji
    - GlassFish (Sun)
    - JBoss (RedHat)
    - WebSphere (IBM)
    - WebLogic (BEA Systems)
    - i wiele innych

Kontenery - Serwery

- Kontenery Aplikacji (Web Containers)
  - Środowisko uruchomieniowe
  - Serwer WWW
  - Środowisko zarządzające
- Serwery Aplikacji
  - Kontener
  - Elementy pomocnicze
- Serwery EJB

Podstawowe usługi JEE

- XML Processing (SAX, JAXP; StAX)
- HTTP, HTTPS
- RMI (Remote Method Invocation)
    - CORBA (Common Object Requesting Broker Architecture)
    - IIORB (General Inter-ORB Protocol)
- Java Persistence
- Java Transactions
- JMS (Java Messaging System)
- JNDI (Java Naming and Directory Interface)
- Java Mail
- Web Services (JAX-WS, JAX-RPC)
- Security Services
- Management
- Deployment



SOAP –(*Simple Object Access Protocol)*
IIOP

Figure EE.2-2 Java EE Interoperability

Podstawowe Aplikacje

- JSP (JavaServer Pages)
  - Dynamiczne strony www
    - „Podobieństwo” do PHP
    - Dużo rozszerzeń
      - JSF, JSTL
  - Unified EL
  - Język
    - Mieszanka jsp i Javy
      - tradycyjne jsp
      - xml

Obsługa połączeń

- Servlety
  - javax.servlet
  - „podobieństwo” do CGI
  - Własna obsługa zawołań do serwera
    - GET, POST
    - AJAX
    - Transakcje, sesje, ciasteczka …
  - Często generowane automatycznie

Fazy w JSP

- Tworzenie
- Deploy
  - Walidacja składni
- Translation
  - Wykonywane raz dla strony
  - Tłumaczenie jsp na Javę i kompilacja
    - Generowanie odpowiednich Servletów
- Obsługa połączeń
  - events -> request -> reply

Powstawanie Aplikacji

- Tworzymy aplikację
  - ręcznie
  - Narzędzia Ecllipse, Netbeans …
- Generujemy plik war
  - Nic innego jak spakowana aplikacja (jar)
- Zawartość
  - META-INF
    - Katalog z metadatą
  - WEB-INF
    - Katalog z właściwościami aplikacji
  - pliki .jsp

WEB-INF

- classes
  - Katalog z skompilowanymi klasami java
- lib
  - Wykorzystane biblioteki (jar)
- web.xml
  - Plik opisujący aplikację

Przykład

pierwsza.jsp

```
<%@ page language="java"
    contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<html>
<head>
<meta http-equiv="Content-Type"
      content="text/html; charset=UTF-8">
<title>Ala ma Kotka</title>
</head>
<body>
<b>też mi nowość :-)</b>
</body>
</html>
```

WEB-INF/web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app id="WebApp_ID" version="2.4"
xmlns="http://java.sun.com/xml/ns/j2ee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">
<display-name>test</display-name>
    <welcome-file-list>
            <welcome-file>index.html</welcome-file>
            <welcome-file>index.htm</welcome-file>
            <welcome-file>index.jsp</welcome-file>
            <welcome-file>default.html</welcome-file>
            <welcome-file>default.htm</welcome-file>
            <welcome-file>default.jsp</welcome-file>
    </welcome-file-list>
</web-app>
```

Składnia jsp

tradycyjny jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
                                        pageEncoding="UTF-8"%>
```

xml jsp

```
<?xml version="1.0" encoding="UTF-8" ?>
<jsp:root xmlns:jsp="http://java.sun.com/JSP/Page" version="2.0">
   <jsp:directive.page language="java"  contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8" />
```

tradycyjny jsp

<%--’ JSPCommentBody

<%@’ DirectiveBody

<%!’ DeclarationBody

<%=’ ExpressionBody

<%’ ScriptletBody

${‘ EL ExpressionBody

xml jsp

<jsp:text’ XMLTemplateText

<jsp:directive.’ XMLDirectiveBody

<jsp:declaration’ XMLDeclarationBody

<jsp:expression’ XMLExpressionBody

<jsp:scriptlet’ XMLScriptletBody

#{‘ EL ExpressionBody

Zmienne standardowe

- request
  - Parametry zapytania
- session
  - Zmienna sesji
- response
  - Zmienna odpowiedzi
    - przekierowanie

Servlety

- Klasa implementuje po
  - javax.servlet.Servlet
    - Np. dziedziczymy po
      - **javax.servlet.http.HttpServlet**
- Dla HttpServlet
  - Mamy metody
    - doGet(…), doPost(…), i wiele innych
  - Przeładowywujemy

Aktywowanie Servletu

- Zmieniamy plik web.xml
  - Wstawiamy wewnątrz <web-app>

```
<servlet>
        <servlet-name>Nazwa</servlet-name>
        <servlet-class>pakiet.NazwaKlasy</servlet-class>
</servlet>
<servlet-mapping>
        <servlet-name>Nazwa</servlet-name>
        <url-pattern>path</url-pattern>
</servlet-mapping>
```