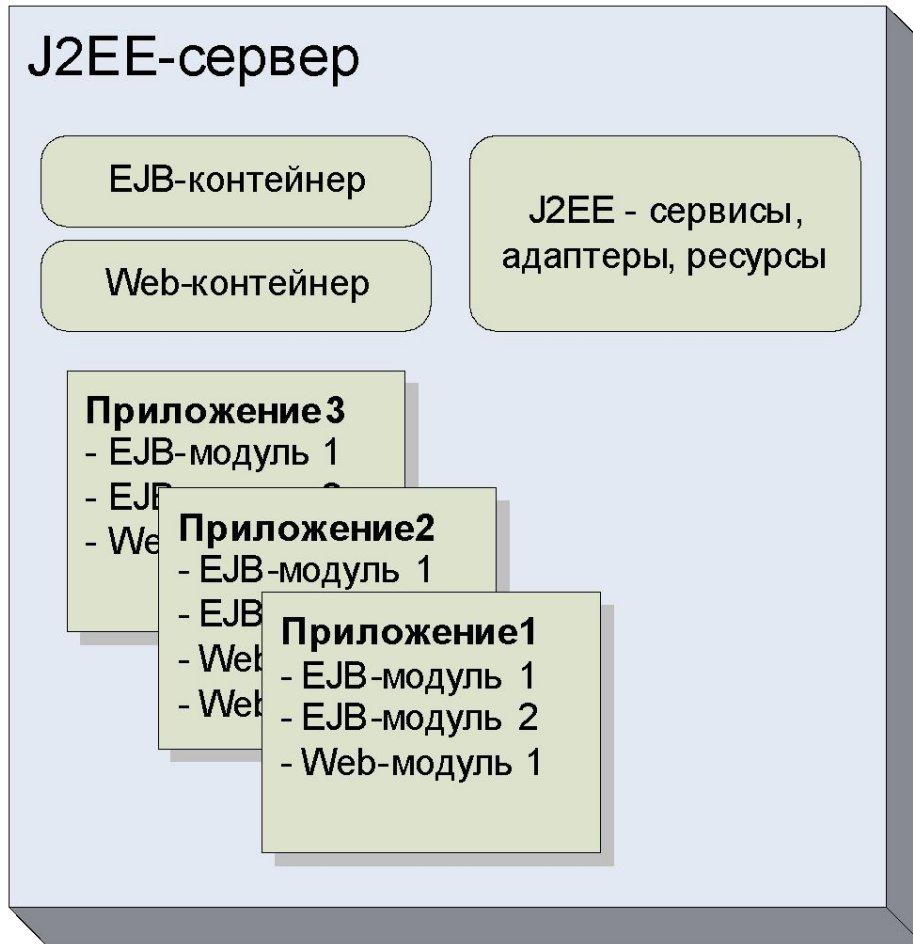


# Платформа J2EE



## Сервис:

Java Naming Directory (JNDI)

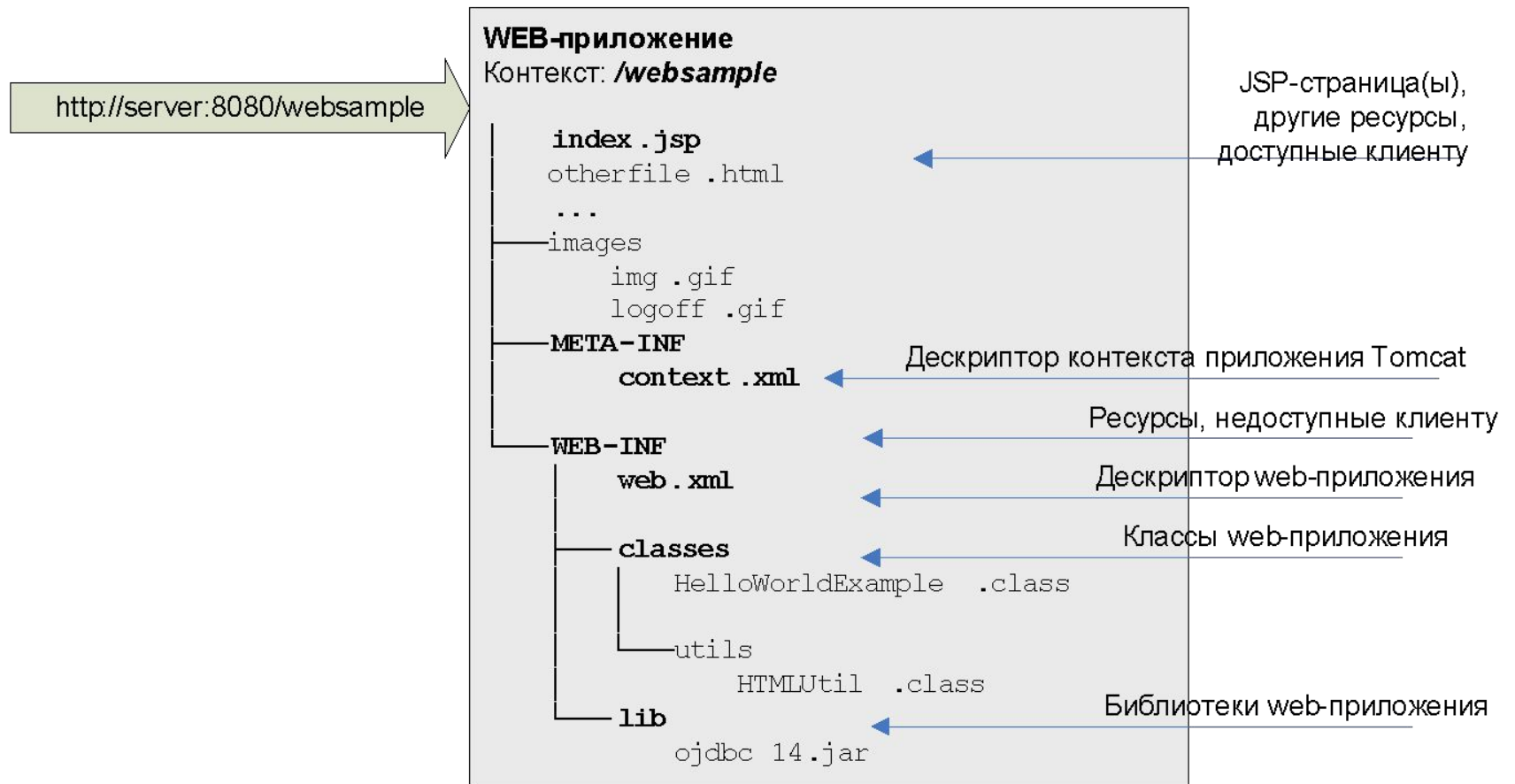
- универсальный сервис хранения объектов в иерархической структуре имен (аналогично файловой системе)

## Ресурс:

DataSource

- объект, позволяющий приложению получить доступ к соединению к БД

# Структура J2EE Web-приложения



# Настройка DataSource в Apache Tomcat

Настройка DataSource:

Файл /META-INF/context.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
  <!-- контекст web-приложения - префикс всех URL для данного приложения -->
  <Context path="/websample">
    <!-- Объявление ресурса-источника данных -->
    <Resource auth="Container" type="javax.sql.DataSource"
      driverClassName="oracle.jdbc.driver.OracleDriver"
      maxActive="20" maxIdle="10" maxWait="-1"
      name="jdbc/sample"
      url="jdbc:oracle:thin:@:1521:spm" username="o50"
      password="o50" />
  </Context>
```

Использование DataSource:

```
// Создаем начальный контекст JNDI (Java Naming Directory)
```

```
InitialContext ctx = new InitialContext();
```

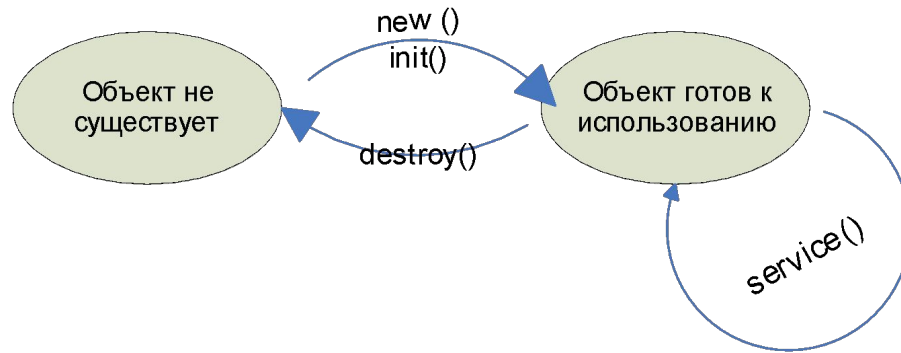
```
// Достаем из контекста источник данных
```

```
DataSource ds = (DataSource)ctx.lookup("java:comp/env/jdbc/sample");
```

```
// Получаем соединение с БД из источника данных
```

```
return ds.getConnection();
```

# J2EE: Сервлеты



Пример:

```
public class MyServlet extends javax.servlet.http.HttpServlet {
    protected void service(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException
    {}
    public void destroy() {
        this.log("Servlet destroyed");
    }
    public void init(ServletConfig cfg) throws ServletException {
        this.log("Servlet inited");
    }
}
```

# Основные классы Servlet API

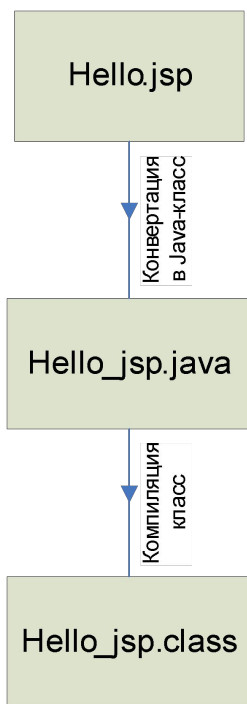
(javax.servlet.http.\*)

- **HttpServletRequest** – класс, экземпляры кот. представляют запрос от браузера
  - **String getContextPath()** – возвращает путь к контексту приложения
  - **String getServletPath()** – URL вызванного сервлета (JSP)
  - **HttpSession getSession()** – Сессия пользователя
  - **Object getAttribute() / void setAttribute(String name, Object value)** – Хранение пользовательских атрибутов, связанных с запросом
  - **String getParameter(String value)** – Параметр запроса (и для GET и для POST)
  - **void setCharacterEncoding(String enc)** – Кодировка значений параметров запроса (windows-1251, UTF-8)
- **HttpServletResponse** – класс, экземпляры кот. представляют ответ браузеру
  - **void setContentType(String contentType)** – MIME-тип ответа браузеру
  - **java.io.PrintWriter getWriter()** – поток вывода для ответа браузеру
  - **void sendRedirect(String location)** – перенаправление на другую страницу
- **HttpSession** – класс, экземпляры кот. хранят состояние сессии клиента
  - **Object getAttribute() / void setAttribute(String key, Object value)** – Атрибуты сессии (сохраняются между запросами одного клиента)
- **ServletContext** – класс, экземпляры кот. представляют все web-приложение
  - **Object getAttribute() / void setAttribute(String key, Object value)** – Атрибуты контекста (общие для всех пользователей и запросов к web-приложению)

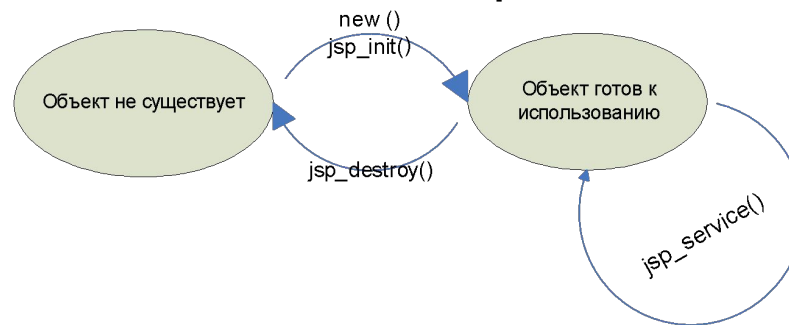
# J2EE: Java Server Pages

## Жизненный цикл

### 1. Жизненный цикл класса страницы



### 2. Жизненный цикл объекта страницы



Элемент JSP	Представление в JSP-файле	Преобразуется в java-класс как
Импорт пакета	<code>&lt;%@ page import="java.util.*" %&gt;</code>	<code>import java.util.*;</code>
Скриптлет:	<code>&lt;% List items=new ArrayList(); // любой java-код %&gt;</code>	<code>jsp_service() { ... List items=new ArrayList(); //любой java-код ... }</code>
Вывод на страницу	<code>&lt;%= new Date() %&gt;</code>	<code>out.write(new Date());</code>

# J2EE: Java Server Pages (JSP)

```
<%@ page language="java"
    contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8" %>
<%@ page import="java.util.*" %>
<%!
// Объявляется поле в классе страницы
int my_integer_field = 777;
// Объявляется метод в классе страницы
private String make_greeting(String name) {
    return "Hello, "+name + "!";
}
%>
<html>
<head>
    <title>Sample Hello world page</title>
</head>
<body>
<% // Скриптлет 1
    for (int i=0; i<10; i++) {
%>
<h1>
<%=
    /* Вывод в поток out */
    make_greeting("World "+i)
%>
</h1>
<% // Скриптлет 2
    } // Конец цикла
%>
</body>
</html>
```



```
import java.util.*;
public final class hello_jsp extends
org.apache.jasper.runtime.HttpJspBase
{
// Объявляется поле в классе страницы
int my_integer_field = 777;
// Объявляется метод в классе страницы
private String make_greeting(String name) {
    return "Hello, "+name + "!";
}
public void _jspService(HttpServletRequest request,
    HttpServletResponse response)
    throws java.io.IOException, ServletException {
    PageContext pageContext = null;
    HttpSession session = null;
    JspWriter out = null;
    Object page = this;
    response.setContentType("text/html; charset=UTF-8");
    session = pageContext.getSession();
    out = pageContext.getOut();
    out.write("<html>\n");
    out.write("<head>\n");
    out.write("\t<title>Sample Hello world page</title>\n");
    out.write("</head>\n");
    out.write("<body>\r\n");
// Скриптлет 1
    for (int i=0; i<10; i++) {

        out.write("\r\n");
        out.write("<h1> ");
        out.print( /* Вывод в поток out */ make_greeting("World "+i) );
        out.write("</h1>\r\n");
// Скриптлет 2
    } // Конец цикла
        out.write("\n");
        out.write("</body>\n");
        out.write("</html>");
    }
}
```