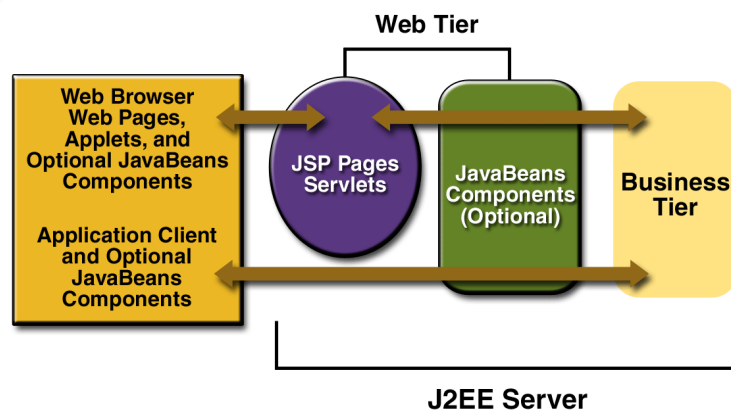
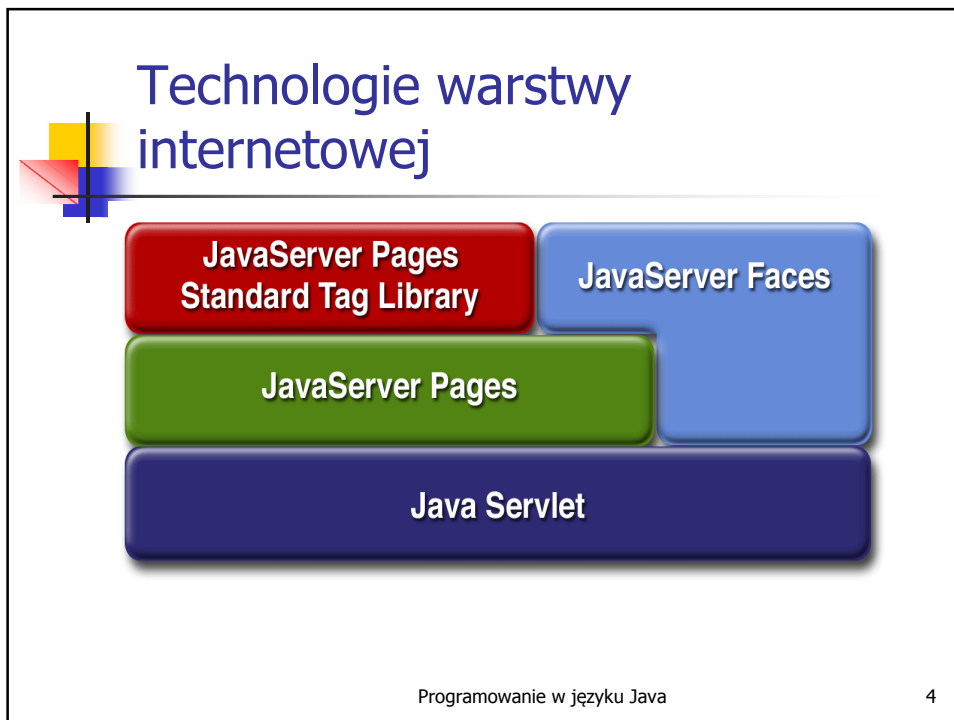
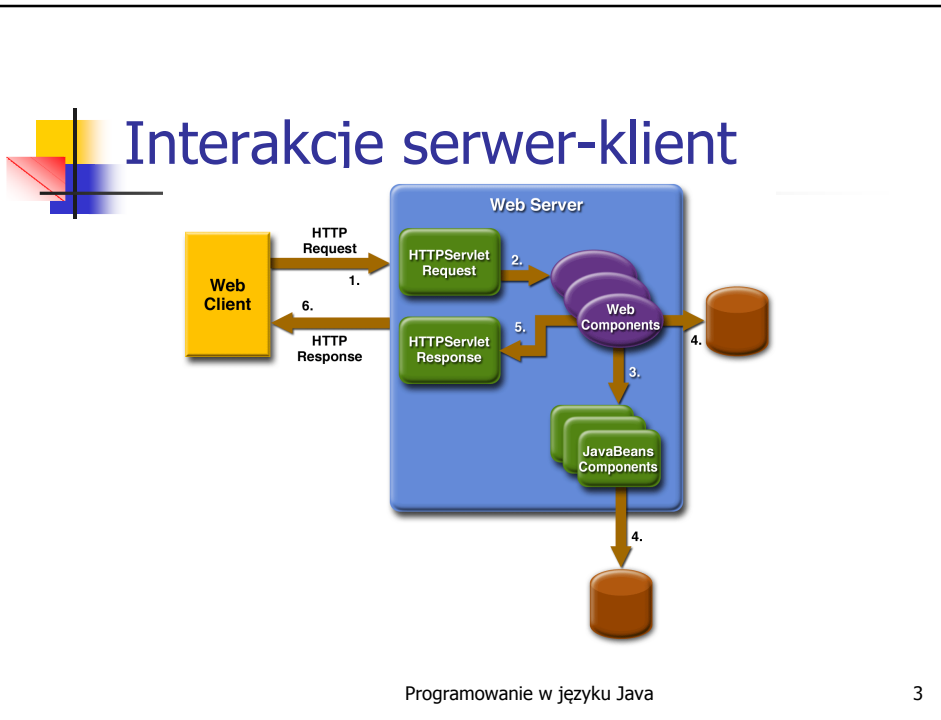


Programowanie w języku Java

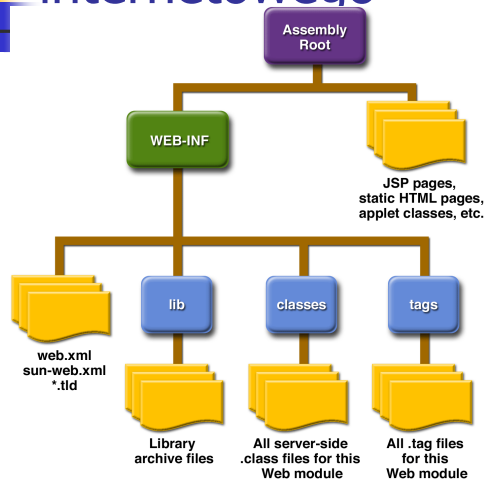
Wykład 6: Programowanie rozproszone: Servlety, JSP

JEE – warstwa WWW





Konfiguracja serwera internetowego



Pliki .war

- automatyczna instalacja (deploy)

Programowanie w języku Java

5

Struktura serwletów



```
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;

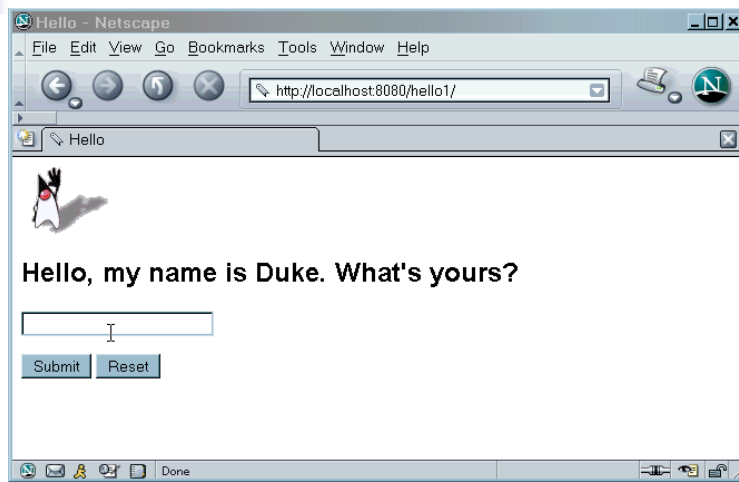
public class ThreadServlet extends HttpServlet {
    public void init(ServletConfig config) throws ServletException {}
    public ServletConfig getServletConfig();
    public void service(HttpServletRequest req, HttpServletResponse res)
        throws IOException {}
    public String getServletInfo() {}
    public void destroy() {}
}
```

Zamiast service() może być doGet() i doPost()

Programowanie w języku Java

6

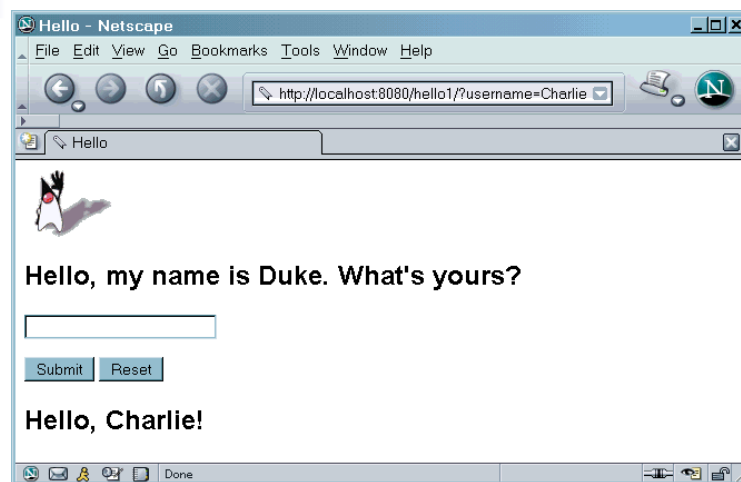
Przykład



Programowanie w języku Java

7

Przykład cd.



Programowanie w języku Java

8

Kod przykładowego servletu

```
import java.io.*;
import java.util.*;
import java.sql.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class GreetingServlet extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html"); response.setBufferSize(8192);
        PrintWriter out = response.getWriter();
        out.println("<html>" + "<head><title>Hello</title></head>");
        out.println("<body bgcolor=\\"#ffffff\\">" + "<img src=\\"duke.waving.gif\\" alt=\\"Duke
        waving\\">" +
            "<h2>Hello, my name is Duke. What's yours?</h2>" + "<form method=\\"get\\">" +
            "<input type=\\"text\\" name=\\"username\\" size=\\"25\\">" + "<p></p>" +
            "<input type=\\"submit\\" value=\\"Submit\\">" + "<input type=\\"reset\\" value=\\"Reset\\">" +
            "</form>");
        String username = request.getParameter("username");
        if ((username != null) && (username.length() > 0)) {
            RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/response");
            if (dispatcher != null) { dispatcher.include(request, response); }
        }
        out.println("</body></html>"); out.close();
    }
}
```

Programowanie w języku Java

9

Kod servletu generującego odpowiedź

```
import java.io.*;
import java.util.*;
import java.sql.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class ResponseServlet extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        PrintWriter out = response.getWriter();
        String username = request.getParameter("username");
        if ((username != null) && (username.length() > 0)) {
            out.println("<h2>Hello, " + username + "!</h2>");
        }
    }
    public String getServletInfo() {
        return "The Response servlet says hello.";
    }
}
```

Programowanie w języku Java

10



Konfiguracja serwera

- Instalacja serwletów:
 - wgranie skompilowanych serwletów do kartoteki classes
 - rejestracja serwletów w pliku web.xml
- Wywołanie:
 - `http://sewer:8080/hello2/greeting`



Plik web.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<web-app xmlns="http://java.sun.com/xml/ns/javaee" version="2.5"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
    http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd">
  <display-name>hello2</display-name>
  <servlet>
    <display-name>GreetingServlet</display-name>
    <servlet-name>GreetingServlet</servlet-name>
    <servlet-class>servlets.GreetingServlet</servlet-class>
  </servlet>
  <servlet>
    <display-name>ResponseServlet</display-name>
    <servlet-name>ResponseServlet</servlet-name>
    <servlet-class>servlets.ResponseServlet</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>GreetingServlet</servlet-name>
    <url-pattern>/greeting</url-pattern>
  </servlet-mapping>
  <servlet-mapping>
    <servlet-name>ResponseServlet</servlet-name>
    <url-pattern>/response</url-pattern>
  </servlet-mapping>
</web-app>
```

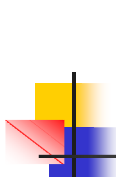


Formularze HTML

- Struktura znacznika:
`<FORM ACTION=„...” ... </FORM>`
- Akcja:
`ACTION=„servlets/MojServlet”` - obsługa przez wskazany servlet
Brak akcji: przesłanie danych pod aktualny adres URL
- Pola formularzy:
Pole1: `<INPUT TYPE=„TEXT” NAME=„par1” VALUE=„x”>`
`<INPUT TYPE=„SUBMIT” VALUE=„Wyślij”>`
Inne typy: PASSWORD, RADIO, itp.

Programowanie w języku Java

13



Odczyt parametrów formularza

- `request.getParameter(„p1”)`
 - zwraca String
- `request.getParameterValues(„p1”)`
 - zwraca tablicę stringów
- `request.getParameterNames()`
 - obiekt typu Enumeration

Programowanie w języku Java

14



Przykład: aplet komunikujący się z servletem (1)

```
import java.io.*;
import javax.servlet.ServletException;
import javax.servlet.http.*;

public class EchoServlet extends HttpServlet {
    public void doPost(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException {
        try {
            response.setContentType("application/x-java-serialized-object");
            InputStream in = request.getInputStream();
            ObjectInputStream inputFromApplet = new ObjectInputStream(in);
            String echo = (String) inputFromApplet.readObject();
            OutputStream outstr = response.getOutputStream();
            ObjectOutputStream oos = new ObjectOutputStream(outstr);
            oos.writeObject(echo);
            oos.flush();
            oos.close();
        } catch (Exception e) { e.printStackTrace(); }
    }
}
```

Programowanie w języku Java

15

Aplet komunikujący się z servletem (2)

```
import java.applet.Applet;
import java.awt.*;
import java.awt.event.*;
import java.io.*;
import java.net.*;

public class EchoApplet extends Applet {
    private TextField inputField = new TextField();
    private TextField outputField = new TextField();
    private TextArea exceptionArea = new TextArea();
    public void init() {
        setLayout(new GridBagLayout());
        Label title=new Label("Echo Applet", Label.CENTER);
        title.setFont(new Font("SansSerif", Font.BOLD, 14));
        GridBagConstraints c = new GridBagConstraints();
        c.gridwidth = GridBagConstraints.REMAINDER;
        c.weightx = 1.0;
        c.fill = GridBagConstraints.HORIZONTAL;
        c.insets = new Insets(5, 5, 5, 5);
        add(title, c);
        c = new GridBagConstraints();
        c.anchor = GridBagConstraints.EAST;
        add(new Label("Input:", Label.RIGHT), c);
        c = new GridBagConstraints();
        c.fill = GridBagConstraints.HORIZONTAL;
        c.weightx = 1.0;
        add(inputField, c);

        Button sendButton = new Button("Send");
        c = new GridBagConstraints();
        c.gridwidth = GridBagConstraints.REMAINDER;
        add(sendButton, c);
        sendButton.addActionListener(new ActionListener()
        { public void actionPerformed(ActionEvent e)
            { onSendData();}
        });
        c = new GridBagConstraints();
        c.anchor = GridBagConstraints.EAST;
        add(new Label("Output:", Label.RIGHT), c);
        c = new GridBagConstraints();
        c.gridwidth = GridBagConstraints.REMAINDER;
        c.fill = GridBagConstraints.HORIZONTAL;
        c.weightx = 1.0;
        add(outputField, c);
        outputField.setEditable(false);
        c = new GridBagConstraints();
        c.anchor = GridBagConstraints.EAST;
        add(new Label("Exception:", Label.RIGHT), c);
        c = new GridBagConstraints();
        c.gridwidth = GridBagConstraints.REMAINDER;
        c.weighty = 1;
        c.fill = GridBagConstraints.BOTH;
        add(exceptionArea, c);
        exceptionArea.setEditable(false);
    }
}
```

Programowanie w języku Java

16



Aplet komunikujący się z servletem (3)

```
private URLConnection getServletConnection()
throws MalformedURLException, IOException {
    URL urlServlet = new URL(getCodeBase(), "echo");
    URLConnection con = urlServlet.openConnection();
    con.setDoInput(true);
    con.setDoOutput(true);
    con.setUseCaches(false);
    con.setRequestProperty("Content-Type",
        "application/x-java-serialized-object");
    return con;
}
```

```
private void onSendData() {
    try {
        String input = inputField.getText();
        URLConnection con = getServletConnection();
        OutputStream outstream =
            con.getOutputStream();
        ObjectOutputStream oos = new
            ObjectOutputStream(outstream);
        oos.writeObject(input);
        oos.flush();
        oos.close();
        InputStream instr = con.getInputStream();
        ObjectInputStream inputFromServlet = new
            ObjectInputStream(instr);
        String result = (String)
            inputFromServlet.readObject();
        inputFromServlet.close();
        instr.close();
        outputField.setText(result);
    } catch (Exception ex) {
        ex.printStackTrace();
        exceptionArea.setText(ex.toString());
    }
}
```

Programowanie w języku Java

17



Inne możliwości servletów

- Obsługa „cookies”
- Śledzenie sesji
- Analiza nagłówek żądań
- Generowanie niestandardowych odpowiedzi
- Przesyłanie plików w różnych formatach

Programowanie w języku Java

18



Podstawowe konstrukcje JSP

- `<H1>` Kod w HTMLu `</H1>`
- `<!--` komentarz HTML `-->`
- `<% --` komentarz JSP `-->`
- `<%=` wyrażenie w Javie `%>`
- `<%` Skryptlet (blok instrukcji w Javie) `%>`
- `<%!` definicja pola lub metody `%>`
- `<%@` dyrektywa atrybut="wartość" `%>`
- `<jsp:akcja>` ... `</jsp:akcja>` - format XML



Domyślne zmienne

- `HttpServletRequest request`
- `HttpServletResponse response`
- `Writer out`
- `HttpSession session`
- `ServletContext application`
- `ServletConfig config`
- `PageContext pageContext`
- `page = this`



Koniec

Programowanie w języku Java

21