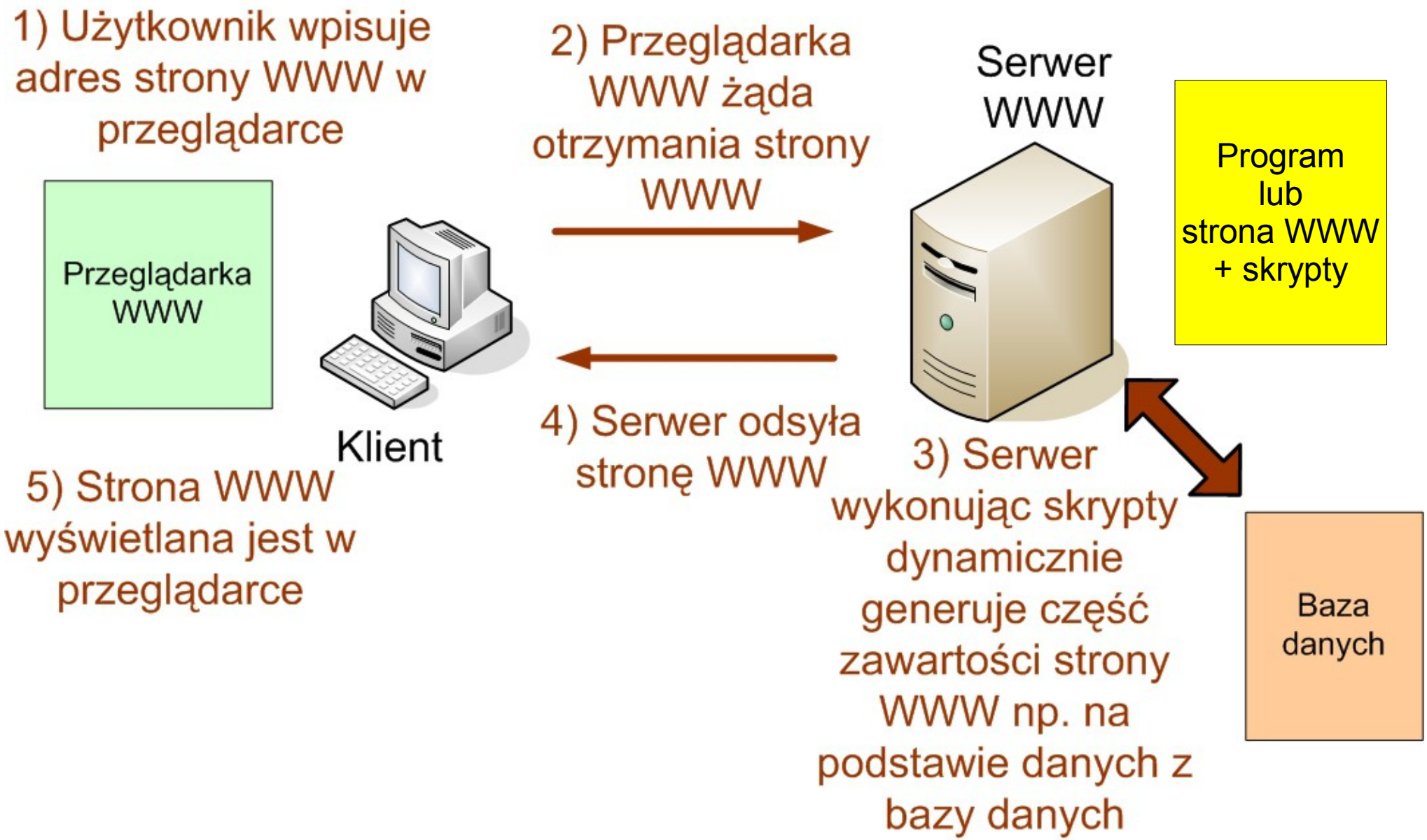
Programowanie współbieżne i rozproszone

WYKŁAD 12

dr inż. Krzysztof Pancerz

Technologie dynamicznych dokumentów WWW



Technologie dynamicznych dokumentów WWW

- Technologie dynamicznych dokumentów WWW nie wymagają bezpośredniego wsparcia przez przeglądarkę.
- Muszą zapewnić zgodność z klientem w zakresie poprawnego wsparcia serwerowej strony obsługi protokołu HTTP

Technologie dynamicznych dokumentów WWW

- CGI (Common Gateway Interface)
  - historycznie pierwsza upowszechniona technologia dokumentów dynamicznych
  - standard specyfikuje protokół uruchamiania programów na serwerze WWW
  - pozwala na udostępnianie dynamicznej informacji
  - nie determinuje języka programowania (często wykorzystywany jest język Perl, poza tym m.in. C, PL/SQL, bash)
  - komunikacja z klientem odbywa się przez protokół HTTP

Technologie dynamicznych dokumentów WWW

- CGI (Common Gateway Interface)
  - standardowy scenariusz działania programu CGI:
    - odczytanie parametrów żądania (ze standardowego wejścia albo ze zmiennych środowiskowych)
    - przetworzenie danych
    - wyprowadzenie komunikatu HTTP na standardowe wyjście

Technologie dynamicznych dokumentów WWW

- SSI (Server Side Includes)
  - kod generujący dynamiczną informację osadzany jest w strukturze dokumentu HTML
  - kod generujący dynamiczną zawartość jest przed wysłaniem dokumentu do klienta interpretowany i zastępowany wynikową treścią

PHP

- Język PHP wykorzystywany jest do generowania dynamicznych dokumentów WWW w technologii SSI.
- Jest językiem skryptowym zagnieżdżanym w kodzie HTML.
- Pierwotna wersja języka stworzona została w 1994 roku przez Rasmusa Lerdorfa.
- Jest wzorowany na językach C/C++, Perl i Java.

ASP.NET

- ASP.NET jest rozwinięciem technologii ASP (Active Server Pages) opartej na koncepcji SSI.
- Dokumenty ASP.NET kompilowane są przez kompilator zgodny z CLR (Common Language Runtime).
- Dostępne są kompilatory dla VB, VC++, C#, Java i inne.

ASP.NET

- Wykorzystywany jest wyższy poziom abstrakcji niż w tradycyjnych technologiach skryptowych (mechanizmy komunikacyjne oraz przetwarzanie danych ukryte są w gotowych komponentach realizujących określoną funkcjonalność).
- Możliwa jest separacja kodu pomiędzy interfejs użytkownika i logikę przetwarzania.

Web Services

- Technologia Web Services stanowi połączenie:
  - oprogramowania pośredniczącego (middleware),
  - aplikacji WWW.
- Technologia Web Services pozwala przezwyciężyć bariery współdziałania rozproszonych aplikacji przez wykorzystanie niezależności od platform (ustandaryzowane protokoły komunikacyjne oraz oparte na tekście formaty zasobów).

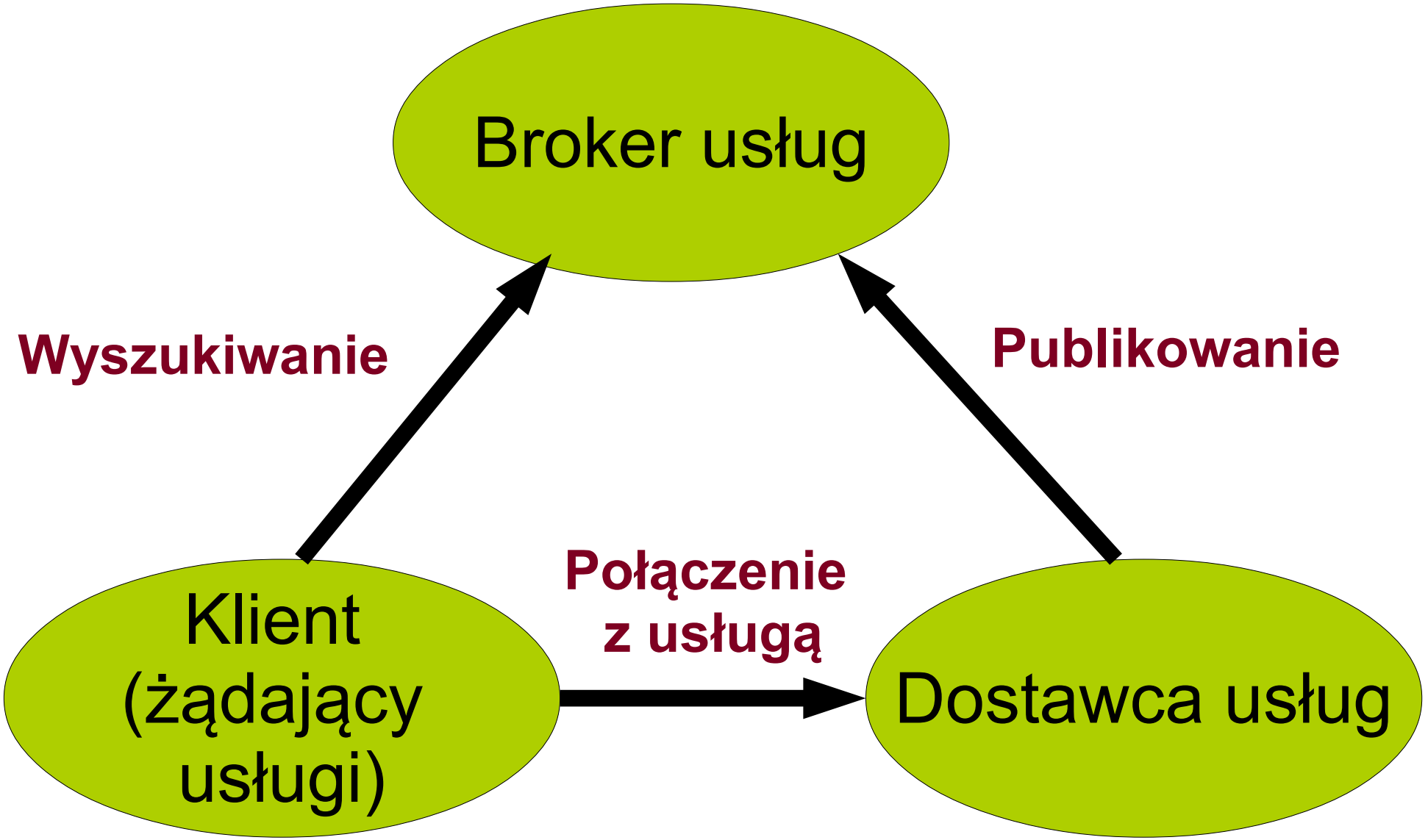
Web Services

- Web Service jest dostępnym poprzez sieć komponentem aplikacyjnym przeznaczonym do wykorzystania przez inne aplikacje, zwane aplikacjami klienckimi.
- Web Services - **hermetyzowane** (implementacja nie jest widoczna na zewnątrz), **luźno skojarzone** (modyfikacja danej implementacji nie powoduje problemu propagacji zmiany), **kontraktowane** (opis działania funkcji oraz specyfikacja ich interfejsów jest publicznie dostępna) funkcje oferowane przez standardowe protokoły.

Web Services

- Elementy w modelu działania Web Services:
  - klienci,
  - dostawcy usług,
  - brokerzy usług (serwisy umożliwiające klientom zlokalizowanie dostawców usług, a dostawcom usług publikowanie ich opisów).

Web Services



Web Services

- Infrastruktura WWW pozwala lokalizować usługi w oparciu o adresy URL ich żądań.
- Rejestr usług dostarczany przez brokera nie jest niezbędny jeśli programista aplikacji pozyskuje specyfikację interfejsu inną drogą.
- Protokół Web Services oparty jest zasadniczo o protokół HTTP.

WSDL

- WSDL (Web Services Description Language) jest specyfikacją W3C maszynowego opisu usług Web Services opartego na XML.
- Struktura pliku WSDL jest zdefiniowana w specyfikacji w postaci schematów XML Scheme.

WSDL

- Struktura pliku WSDL:

```
<definitions>
<import/>
<types> <xs:element> ... </xs:element> <types>
<interface> <operation> ... </operation> <interface>
<binding> ... </binding>
<service> ... </service>
</definitions>
```

WSDL

- Przykład pliku WSDL:

`http://www.w3.org/2001/04/wsws-proceedings/uche/wsdl.html`

UDDI

- UDDI (Universal Description, Discovery and Integration) - repozytorium do przechowywania definicji Web Services zapisanych w języku WSDL.
- Projekt UDDI dostępny jest pod adresem:

http://www.uddi.org

Czynności przypisane do ról w architekturze Web Services

- Dostawcy usług rejestrują się u brokerów (UDDI)
- Klienci pytają brokerów o usługi (SOAP)
- Brokerzy lokalizują usługi i podają ich opisy (WSDL)
- Klienci podłączają się do usług i korzystają z nich (SOAP)

Protokół SOAP

- SOAP - Simple Object Access Protocol (Prosty protokół dostępu do obiektów) - protokół wykorzystywany do wywoływania metod serwerów, serwisów, komponentów oraz obiektów).
- Dane w postaci XML przesyłane są zasadniczo za pomocą protokołu HTTP.

Protokół SOAP

- Specyfikacja SOAP określa rodzaje nagłówków HTTP oraz definiuje słownik znaczników XML wykorzystywanych do reprezentacji parametrów, zwracanych wartości oraz generowanych przez metody wyjątków.

Protokół SOAP

- Struktura komunikatu SOAP:
  - **Envelope** - opakowanie, element bazowy (korzeń dokumentu).
  - **Header** - nagłówek (opcjonalny). Bloki nagłówka (podelementy elementu Header) zawierają informacje precyzujące sposób traktowania wiadomości (np. kodowanie).
  - **Body** - ciało zawierające parametry wywołania / odpowiedź / informacje o błędach.

Protokół SOAP

- Cechy komunikatu SOAP:
  - musi być zapisany w formacie XML
  - musi zawierać element Envelope
  - może zawierać element Header
  - musi zawierać element Body
  - musi używać odpowiednich przestrzeni nazw
  - nie może zawierać odwołania do DTD
  - nie może zawierać instrukcji przetwarzania XML

Protokół SOAP

- Przykład struktury komunikatu SOAP:

```
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

<soap:Header>
...
</soap:Header>

<soap:Body>
...
  <soap:Fault>
  ...
  </soap:Fault>
</soap:Body>

</soap:Envelope>
```

Źródło: http://www.w3schools.com/

Protokół SOAP

- Przykład komunikatu żądania SOAP:

```
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

<soap:Body>
  <m:GetPrice xmlns:m="http://www.w3schools.com/prices">
    <m:Item>Apples</m:Item>
  </m:GetPrice>
</soap:Body>

</soap:Envelope>
```

Źródło: http://www.w3schools.com/

Protokół SOAP

- Przykład komunikatu odpowiedzi SOAP:

```
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

<soap:Body>
  <m:GetPriceResponse xmlns:m="http://www.w3schools.com/prices">
    <m:Price>1.90</m:Price>
  </m:GetPriceResponse>
</soap:Body>

</soap:Envelope>
```

Źródło: http://www.w3schools.com/

Platforma Java Enterprise Edition (Java EE)

- Platforma Java Enterprise Edition (Java EE), to platforma umożliwiająca tworzenie aplikacji webowych i korporacyjnych.
- Aplikacje te są zazwyczaj wielowarstwowe.
- Aktualna wersja: Java EE 7, trwają prace nad Java EE 8.

Warstwy technologii J2EE

- Warstwa kliencka (przeglądarka, aplikacja napisana w języku Java).
- Warstwa webowa (serwlety i strony JSP).
- Warstwa biznesowa (komponenty Enterprise JavaBeans).
- Warstw danych (np. bazy danych).

Serwlety

- Serwlety są technologią należącą do Java Enterprise Edition (JEE).
- Serwlet jest elementem oprogramowania odpowiadającym na żądania, zwykle HTTP.
- Serwlety służą do generowania dynamicznych dokumentów WWW.
- Model działania nawiązuje do technologii CGI

Serwlety

- Serwlet jest obiektem klasy implementującej interfejs `javax.servlet.Servlet` wykonywanym w środowisku maszyny wirtualnej przez usługę zwaną silnikiem serwletów (serwlet engine).
- Serwlet może korzystać ze standardowych klas Java oraz klas wchodzących w skład Servlet API.
- Serwlet nie posiada interfejsu użytkownika, komunikuje się z klientem (przeglądarką WWW) za pomocą protokołu HTTP.

Serwlety

- Zalety serwletów:
  - przystosowanie do wykonywania wielowątkowego,
  - zazwyczaj brak potrzeby wywoływania poleceń systemu operacyjnego, dzięki API języka Java, co zwiększa bezpieczeństwo serwletów,
  - możliwość określania precyzyjnej kontroli dostępu do zasobów dzięki ACL (Java Access Control List),
  - mechanizm obsługi wyjątków.

Serwlety

- Serwlet jest ładowany przy pierwszym odwołaniu do niego lub przy starcie silnika i pozostaje w pamięci do momentu jawnego usunięcia lub zatrzymania serwera.
- Metody serwletu wyznaczające jego cykl życiowy:
  - `init`
  - `service`
  - `destroy`

Serwlety

- Metoda `service` zawiera funkcjonalność wywoływaną każdorazowo przy nadejściu żądania od klienta.
- Jej implementacja musi uwzględniać wielowątkowość.

Serwlety

- Dane wejściowe od klienta (przeglądarki WWW) dostępne są w obiekcie klasy `HttpServletRequest`. Podstawowe metody:
  - `getParameter`
  - `getCookies`
- Dane wyjściowe dla klienta (przeglądarki WWW) wysyłane są poprzez obiekt klasy `HttpServletResponse`. Podstawowe metody:
  - `getWriter` (dla danych tekstowych)
  - `getOutputStream` (dla danych binarnych)

JSP

- JSP (Java Server Pages) stanowią element platformy Java Enterprise Edition (JEE).
- Działają w oparciu o technologię SSI.
- Kod źródłowy Java osadzany jest w kodzie HTML wewnątrz znaczników <% %>.
- Osadzony kod Java przetwarzany jest przez silnik JSP do postaci serwletu i kompilowany.

Enterprise JavaBeans

- Komponenty Enterprise JavaBeans (EJB) umożliwiają tworzenie aplikacji rozproszonych na bazie komponentów oferujących:
  - skalowalność,
  - przetwarzanie transakcyjne,
  - bezpieczeństwo.

Enterprise JavaBeans

- Rodzaje komponentów EJB:
  - komponenty sesyjne (*Session Beans*),
  - komponenty sterowane komunikatami (*Message-diven Beans*),
  - komponenty encyjne (*Entity Beans*) – zastąpione przez Java Persistence API.

Enterprise JavaBeans

- Komponenty EJB działają w kontenerze EJB, który zajmuje się ich tworzeniem, usuwaniem i przypisywaniem do żądań klientów.

Komponenty sesyjne EJB

- Komponenty sesyjne EJB:
  - mają na celu obsługiwanie żądań klientów,
  - działają w ten sposób, że w danym momencie co najwyżej jeden klient jest przypisany do danego egzemplarza komponentu,
  - są dostępne przez protokół IIOP, przez co mogą być wykorzystywane przez klientów implementujących mechanizm CORBA.

Komponenty sesyjne EJB

- Komponenty sesyjne stanowe:
  - wykorzystywane są do konwersacji z określonym klientem,
  - stan zapisywany jest w polach instancji komponentu stanowego, obiektów powiązanych oraz obiektów osiągalnych z komponentu,
  - oznaczane są adnotacją **@Stateful**

Komponenty sesyjne EJB

- Komponenty sesyjne bezstanowe:
  - nie zawierają żadnych informacji o konwersacji z określonym klientem,
  - aby być dostępnymi zdalnie muszą implementować interfejsy biznesowe oznaczone adnotacją @Remote
  - kontener wybiera instancję komponentu bezstanowego, która obsłuży wywołanie metody,
  - oznaczane są adnotacją **@Stateless**

Komponenty EJB sterowane komunikatami

- Komponenty EJB sterowane komunikatami:
  - obsługują zgłaszane komunikaty pozwalające klientom na jednostronną komunikację z serwerem,
  - działają wg następującego mechanizmu:
    - klient wysyła komunikat i kontynuuje przetwarzanie, nie czekając na odpowiedź,
    - kontener po stronie serwera przekazuje komunikat do komponentów oczekujących na wiadomości.