SOA – laboratorium nr 2

Temat: Servlety w JavaEE

W nawiązaniu do poznanej na zajęciach nr 1 koncepcji servletu wykonaj poniższe zadania.

Zadanie 1.

Napisz aplikacje internetowa wykorzystując koncepcje servleta, pozwalającą sprawdzić czy podane na stronie internetowej dane (imie oraz wiek) należą do pełnoletniej kobiety czy też nie. Imię kobiece rozpoznajemy po końcówce ‘a’.

Zadanie 2.

Utwórz aplikację w modelu MVC korzystając z serwletów będącą grą w papier-nożyczk i kamień.

Zadanie 3.

- Napisz servlet, który będzie przyjmował 5 liczb całkowitych metodą GET, wyliczy ich średnią i zwróci wynik.
- Napisz servlet, który będzie przyjmował dowolną ilość parametrów metodą POST i sprawdzał, czy parametry te są liczbami. Jeśli są, niech je wyświetli w kolejności od najmniejszej do największej. Jeśli parametry nie są liczbami, niech servlet zwróci informacje o błędnych danych.

  podpowiedź– sprawdź co robi funkcja „getParameterNames()” i czy można ją wykorzystać ją do pobierania dowolnej ilości parametrów z request’a.

Zadanie 4.

a. Stworz formularz zawierający dwa pola: lista wyboru Rodzaj samochodu (sportowy, miejski, luksusowy) oraz pole tekstowe Przedział cenowy.
b. Przetwarzanie formularza ma być przekazane do servletu CarChoiceServlet.
c. Napisz servlet CarChoiceServlet wypisujący parametry otrzymane z przetwarzania formularza.
d. Zaimplementuj klasę pomocniczą CarChoiceHelper (klasa umieszczona w pakiecie Car) wspomagającą wybor samochodu na podstawie parametrow opisujących rodzaj samochodu oraz przedział cenowy. Klasa powinna zwracać listę odpowiednich marek samochodow.
e. Zmodyfikuj kod servletu tak, aby wykorzystywał klasę pomocniczą.

Zadanie 5 - Księga gości

wykorzystanie koncepcji sesji i cookies.

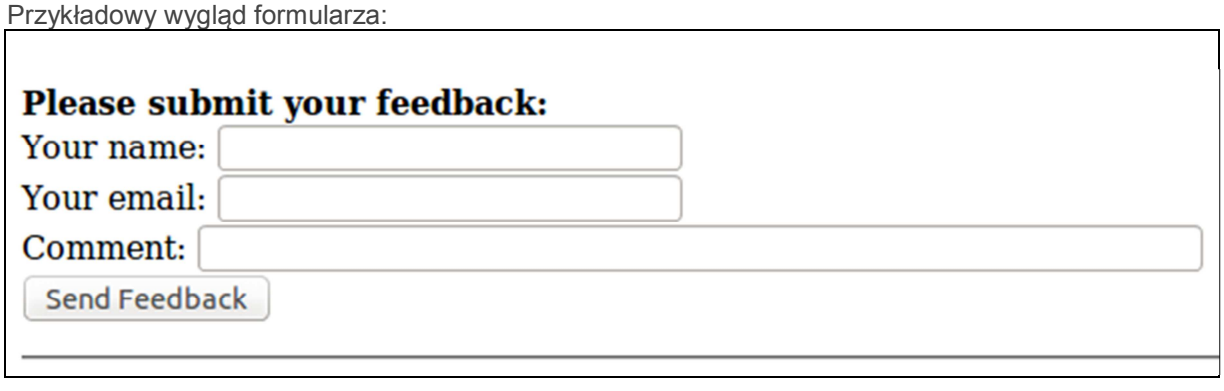
Przygotuj stronę logowania. Założenia:

- Formularz wyświetla 2 pola (login i hasło) oraz przycisk "Zaloguj".
- Wewnątrz skryptu w sposób jawny jest zadeklarowany wektor przechowujący elementy typu *DaneOsobowe* (składowe: login, hasło, imie, nazwisko). Wektor ten powinien zostać wypełniony przykładowymi danymi.
- Scenariusze użycia:
  - nie podano loginu - komunikat "Podaj login" i ponowne wyświetlenie formularza,
  - nie podano hasła - komunikat "Podaj hasło" i ponowne wyświetlenie formularza,
  - błędne dane logowania - komunikat o błędzie (tekst dowolny) i ponowne wyświetlenie formularza,
  - poprawne dane logowania – przejście do strony obsługującej księgę gości. (patrz poniżej).

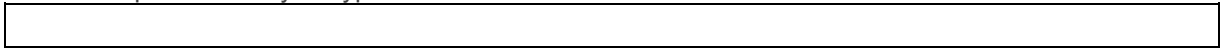
Przygotuj **sevlet** obsługujący prostą księgę gości. Założenia:

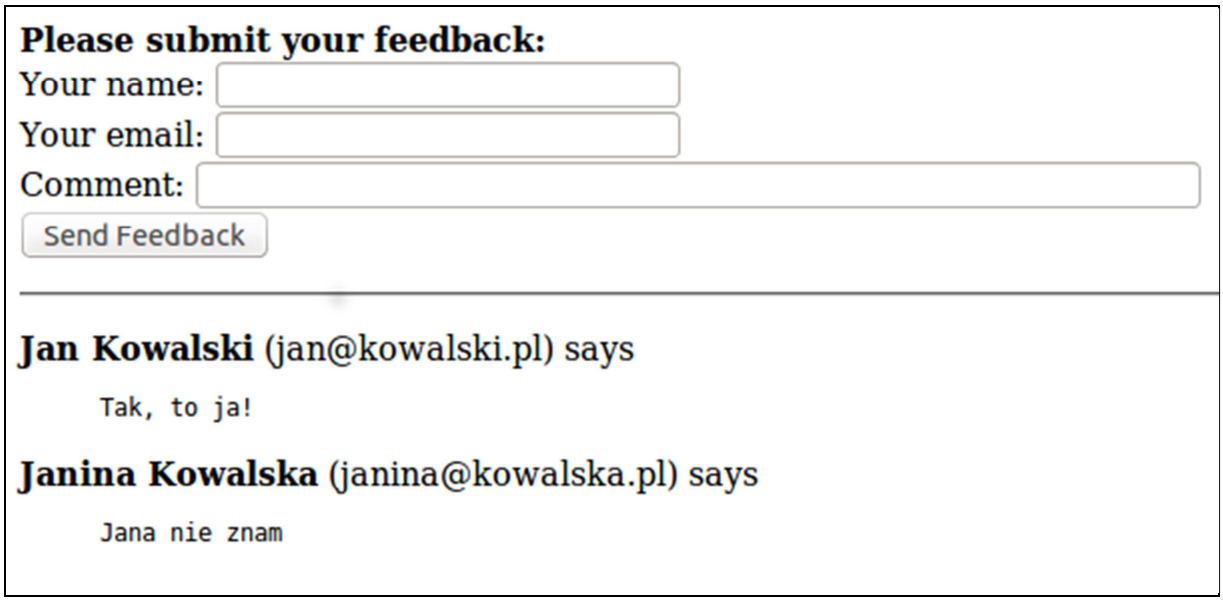
- Cały program jest obsługiwany w technologii Java Servlet. Zaleca się przygotowanie całości w jednym pliku źródłowym.
- Dane podane przez użytkownika powinny być utrwalane w pamięci serwera - zalecana struktura to *Vector*.
  - Dane nie muszą być utrwalane na stałe. Po restarcie serwera mogą być zerowane.
  - Dane wpisane w formularzu powinny być widoczne również z innej sesji przeglądarki (proszę przeprowadzić testy w przeglądarce trybie "Prywatnym").\

Przykładowy wygląd formularza:



Formularz po dwukrotnym wypełnieniu:





Informacje pomocnicze:

Parametry inicjalizacyjne servletu

Dodanie parametrów inicjalizacyjnych servletu polega na modyfikacji pliku web.xml i dodaniu wpisów określających parametry:

```
<servlet>
<servlet-name>StateSaverServlet</servlet-name>
<servlet-class>Servlets.StateSaverServlet</servlet-class>
<init-param>
<param-name>login</param-name>
<param-value>koper</param-value>
</init-param>
</servlet>
```

Dostęp do parametrów inicjalizacyjnych

Odczytanie parametrów konfiguracyjnych servletu wymaga pobrania obiektu klasy ServletConfig poprzez wywołanie metody getServletConfig() w obiekcie servletu, a następnie wywołaniu metody getInitParameter():

```
getServletConfig().getInitParameter()
```

Dostęp do sesji

Obiekt klasy HttpSession reprezentujący sesję jest dostępny jako poprzez wywołanie request.getSession() w przypadku servletu oraz zmiennej lokalnej session w przypadku JSP.

Przydatne metody:

- session.setAttribute() - dodanie obiektu do sesji,
- session.getAttribute() - pobranie obiektu z sesji,
- session.removeAttribute() - usuniecie obiektu z sesji,
- session.invalidate() - zniszczenie sesji.

Dostęp do kontekstu servletu (aplikacji!)

W przypadku servletu należy pobrać referencję do obiektu klasy ServletContext poprzez metodę getServletContext(), zaś w przypadku strony JSP poprzez zmienną lokalną application.

Przydatne metody:

- servletContext.setAttribute() - dodanie obiektu do kontekstu servletu,

• servletContext.getAttribute() - pobranie obiektu z kontekstu servletu.

Obsługa ciasteczek

Dostęp do ciasteczek realizowany jest poprzez obiekty request oraz response:

• response.addCookie() - metoda dodająca ciasteczko do przeglądarki klienta,

• request.getCookies() - metoda zwracając tablicę wszystkich ciasteczek pobranych od klienta.

Ustawianie czasu życia ciasteczka poprzez metodę cookie.setMaxAge(). Natychmiastowe usunięcie ciasteczka: cookie.setMaxAge(0) – jako wysłanie ciasteczka o tej samej nazwie do klienta poprzez obiekt response!!!

Przekierowanie obsługi żądania (w obrębie kontenera!)

Najpierw należy pobrać obiekt klasy RequestDispatcher za pomocą metody:
request.getRequestDispatcher()
Następnie trzeba wywołać metodę forward() zwróconego obiektu klasy RequestDispatcher

Przekierowanie obsługi żądania (poprzez przeglądarkę klienta!)

Przekierowanie przeglądarki do innego adresu URL: response.sendRedirect()