

zaawansowane technologie Javy

programowanie za pomocą znaczników JSTL

Instytut Informatyki
Uniwersytetu Wrocławskiego

Paweł Rzechonek



Jakarta Standard Tag Library (JSTL, dawniej JavaServer Pages Standard Tag Library) jest składnikiem platformy Java EE Web. Rozszerza specyfikację JSP, dodając bibliotekę znaczników do JSP dla typowych zadań jak,

przetwarzanie strukturalne (wykonanie warunkowe, pętle), przetwarzanie danych zawartych w dokumentach XML, dostęp do danych przechowywanych w bazie danych, formatowanie danych zgodnie ze wskazaną lokalizacją (internacjonalizacja) itp.

Zadanie 1: rok przestępny

Stwórz aplikację webową w postaci strony JSP korzystającej ze standardowych znaczników JSTL i wyrażeń EL, która będzie określała, czy podany rok jest przestępny czy zwykły.

Aplikacja ma się składać z jednego dokumentu .jsp, który odbiera dane od użytkownika poprzez formularz, oblicza wynik i odsyła z powrotem. Formularz ma zawierać tylko jedno pole numeryczne `<input type="number" name="rok" ... />` do wprowadzenia roku w kalendarzu gregoriańskim (czterocyfrowa liczba całkowita z zakresu od 1582 do 9999).

Przy pierwszym uruchomieniu aplikacji należy wykonać obliczenia dla roku bieżącego, odczytanego z zegara systemowego w komputerze:

```
<jsp:scriptlet>
    java.util.Date d = new java.util.Date();
    String r = String.format("%4d", d.getYear() + 1900);
    request.setAttribute("r", r);
</jsp:scriptlet>
<c:set var="rok" scope="request" value="${r}" />
```

Przy kolejnych uruchomieniach aplikacji skorzystamy już z przekazanego parametru "rok":

```
<c:set var="rok" scope="request" value="${param.rok}" />
```

Dalsze obliczenia będą wykonywane na zmiennej o nazwie rok.

Całą logikę przetwarzania danych zaprogramuj, używając znaczników JSTL kategorii core (konstrukcje sterujące wykonaniem programu) oraz wyrażeń zapisanych w języku EL, bez używania skryptletów (jedyne wyjątek został pokazany w przykładowym fragmencie kodu, aby wprowadzić atrybut "r", który można potem wykorzystać w wyrażeniu EL).

Zadanie 2: godzina potocznie

Stwórz aplikację webową w postaci strony JSP korzystającą ze standardowych znaczników JSTL i wyrażeń EL, która będzie wypisywała zadaną godzinę tak jak się to mówi potocznie w języku polskim (z uwzględnieniem deklinacji przy liczebnikach). Przykłady:

7:00 – dokładnie 7, 7:15 – kwadrans po 7, 7:30 – wpół do 8, 7:45 – kwadrans do 8,

17:01 – 1 minuta po 17, 17:10 – 10 minut po 17, 17:23 – 23 minuty po 17,

19:37 – 23 minuty do 20, 19:43 – 17 minut do 20, 19:59 – 1 minuta do 20.

Aplikacja ma się składać z jednego dokumentu .jsp, który odbiera dane od użytkownika poprzez formularz, oblicza wynik i odsyła z powrotem. Formularz ma zawierać tylko jedno pole wejściowe `<input type="time" name="czas" ... />` do wprowadzenia godziny.

Przy pierwszym uruchomieniu aplikacji należy wykonać obliczenia dla bieżącej godziny, odczytanej z zegara systemowego w komputerze.

Całą logikę przetwarzania danych zaprogramuj, używając znaczników JSTL kategorii core (konstrukcje sterujące wykonaniem programu) i functions (operacje na łańcuchach znakowych) oraz wyrażeń zapisanych w języku EL, bez nadużywania skryptletów.

Zadanie 3: rozkład na czynniki pierwsze

Stwórz aplikację webową w postaci strony JSP korzystającą ze standardowych znaczników JSTL i wyrażeń EL, która będzie wypisywała rozkład zadanej liczby całkowitej na czynniki pierwsze. Rozkład liczby ma być wypisany w następujący sposób: najpierw liczba, potem znak = a potem czynniki pierwsze odseparowane znakiem *. Rozkład liczby ujemnej na czynniki pierwsze ma się rozpoczynać od wartości -1. Przykłady:

$72 = 2 * 2 * 2 * 3 * 3$

$0 = 0$

$-5 = -1 * 5$

$-147 = -1 * 3 * 7 * 7$

Aplikacja ma się składać z jednego dokumentu .jsp, który odbiera dane od użytkownika poprzez formularz, oblicza wynik i odsyła z powrotem. Formularz ma zawierać tylko jedno pole numeryczne `<input type="number" name="n" ... />` do wprowadzenia liczby (liczba całkowita typu int z zakresu od -2147483648 do 2147483647).

Przy pierwszym uruchomieniu aplikacji należy wykonać obliczenia dla wartości 0.

Całą logikę przetwarzania danych zaprogramuj, używając znaczników JSTL kategorii core (konstrukcje sterujące wykonaniem programu) oraz wyrażeń zapisanych w języku EL, bez nadużywania skryptletów.

Zapisz za pomocą znaczników algorytm podobny do poniższego kodu napisanego w Javie (zmienna n przechowuje wartość do przetworzenia):

```
if (n < -1) { // case for negative integer
    System.out.print("-1 * ");
    n = -n;
}
if (n < -1) { // special case for Integer.MIN_VALUE (-2147483648)
    System.out.print("2 * ");
    n /= 2;
    n = -n;
}
if (n < 2) // case for -1, 0, 1
    System.out.print(n);
else {
    for (int k = 2; k * k <= n; k++) {
        while (n % k == 0) {
            System.out.print(k);
            n /= k;
            if (n > 1) // it's not over yet
                System.out.print(" * ");
        }
    }
    if (n > 1) // the rest was left after calculations
        System.out.print(n);
}
```

Zadanie 4: miasta w krajach świata

Stwórz aplikację webową w postaci strony JSP korzystającej ze standardowych znaczników JSTL i wyrażeń EL, która będzie wypisywała informacje o wybranym mieście we wskazanym kraju na świecie.

Aplikacja ma się składać z ciągu dokumentów .jsp, które odbierają decyzję użytkownika i przenoszą go dalej albo wstecz w wyszukiwaniu informacji o miastach.

Dane o miastach znajdują się w bazie danych. Przed przystąpieniem do programowania uruchom system bazodanowy MySQL (można skorzystać z pakietu XAMPP i serwisu phpMyAdmin do zarządzania systemem), utwórz bazę o nazwie world (ustaw kodowanie znaków na utf8_general_ci) i zaimportuj do niej dane z dołączonego pliku world.sql.

Aplikacja powinna działać według następującego scenariusza. Na pierwszej stronie wyświetla się lista z kontynentami odczytanymi z bazy (tabela country); po wybraniu kontynentu użytkownik przenosi się do drugiej strony. Na drugiej stronie wyświetla się lista z krajami na

wybranych kontynencie odczytanymi z bazy (tabela `country`); po wybraniu kraju użytkownik przenosi się do trzeciej strony. Na trzeciej stronie wyświetla się lista z miastami w wybranym kraju na wybranym kontynencie odczytanymi z bazy (tabela `city`); po wybraniu miasta użytkownik przenosi się do czwartej strony. Na czwartej stronie wyświetlają się ważne informacje o wskazanym mieście (region kraju w którym jest położone miasto, liczba ludności oraz czy to jest stolica) w wybranym kraju na wybranym kontynencie odczytane z bazy (tabele `country` i `city`). Z każdej strony można się wycofać do poprzedniej.

Całą logikę przetwarzania danych zaprogramuj, używając znaczników JSTL kategorii `sql` (konstrukcje wspierające wysyłanie zapytań do bazy danych) oraz wyrażeń zapisanych w języku EL, bez nadużywania skryptletów.