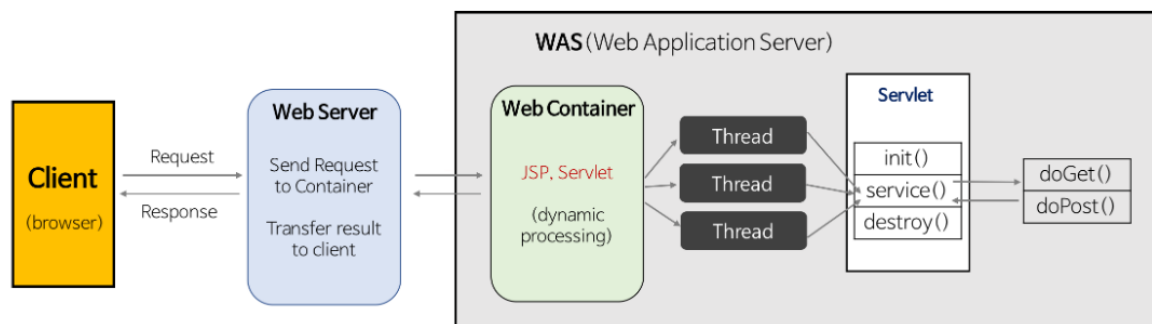


zaawansowane technologie Javy

servlety

Instytut Informatyki
Uniwersytetu Wrocławskiego

Paweł Rzechonek



Servlety to jedna z kluczowych technologii w środowisku Javy, wykorzystywana do tworzenia dynamicznych aplikacji internetowych. Servlet to klasa zdefiniowana w Javie, która implementuje interfejs `Servlet`, albo prościej dziedziczy po klasie `HttpServlet`. Obiekty servletów są tworzone i działają w kontenerze servletów, który jest częścią serwera aplikacyjnego. Kiedy klient wysyła żądanie do serwera (zwykle za pomocą przeglądarki internetowej), kontener servletów odbiera to żądanie i przekazuje je odpowiedniemu servletowi do realizacji; servlet przetwarza żądanie, generuje odpowiedź (często korzystając z bazy danych) i zwraca ją z powrotem do klienta. W ten sposób servlety umożliwiają dynamiczne generowanie treści na stronie internetowej.

Servlety są częścią specyfikacji Jakarta EE (kiedyś Java EE) i są wspierane przez większość serwerów aplikacji jadowych, takich jak Apache Tomcat.

Zadanie 1: kula

Mała Do bardzo lubi okrągłe kształty – nawiasem mówiąc sama też jest trochę okrągła. Teraz postanowiła zrobić z gliny nową kulistą formę artystyczną i potrzebuje obliczyć ile materiału będzie potrzebować na to dzieło sztuki oraz ile farby na jego pomalowanie. Pomóżmy jej!

Stwórz servlet, który na podstawie długości promienia kuli wyliczy jej objętość i pole powierzchni. W przypadku tego servletu promień należy przekazać przez adres URL (metoda GET w protokole HTTP). Przykładowo:

<http://localhost:8080/swank/lab6/kula?r=19.68>

Promień odczytaj za pomocą metody `request.getParameter("r")`. Otrzymana wartość jest typu `String`, więc należy ją najpierw przekształcić na typ `double` a dopiero potem wykonać obliczenia i generować wyniki.

Zabezpiecz aplikację webową przed błędnymi danymi. Jeśli metoda `getParameter()` zwróci wartość pustą `null` to znaczy, że użytkownik nie dopisał parametru `r` do adresu URL – wtedy należy wypisać komunikat z instrukcją obsługi aplikacji (należy wyjaśnić jaki parametr trzeba dopisać do adresu URL). Jeśli pobrany promień nie da się przekonwertować na wartość typu `double`, to nie wykonujemy żadnych obliczeń, za to wypisujemy komunikat z informacją o błędnym formacie danych. Wreszcie, gdy promień udało się przekonwertować ale jego wartość jest ujemna, to również nie wykonujemy obliczeń a tylko wypisujemy komunikat o niepoprawnej wartości.

Dodatkowo na końcu wygenerowanego przez servlet dokumentu umieść informację o autorze (Twoje imię i nazwisko) i jego skrzynce mailowej. Informacje te należy umieścić w pliku `web.xml` jako parametry inicjalizacji kontekstu w znaczniku `<context-param>` a odczytać w servlecie za pomocą funkcji `getServletContext().getInitParameter()`.

Zadanie 2: trójkąt

Mała Do nie znosi trójkątów – ma jakieś niedobre życiowe doświadczenia z trójkątami. Ostatnio jednak znalazła trzy ładnie wyprofilowane listewki, które idealnie nadają się do zrobienia ramy do jej nowego trójkątnego obrazu z alpakami. Potrzebuje jednak wyliczyć obwód obrazu i jego powierzchnię, bo musi zabezpieczyć odpowiednią ilość środków do jego zakonserwowania (werniks i bejca). Pomóżmy jej!

Stwórz servlet, który na podstawie długości boków trójkąta wyliczy jego obwód i pole powierzchni. W przypadku tego servletu długości boków należy przekazać przez formularz. Sam formularz umieść na osobnej stronie HTML zbierającej dane i wysyłającej je do servletu (metoda POST w protokole HTTP); w formularzu umieść trzy pola numeryczne dla boków trójkąta:

```
<input type="number" value="..." step="0.1" min="0.1" name="a" />
<input type="number" value="..." step="0.1" min="0.1" name="b" />
<input type="number" value="..." step="0.1" min="0.1" name="c" />
```

Zabezpiecz aplikację webową przed błędnymi danymi. Nie chodzi tylko o brak danych, błędny format danych, czy niedopuszczalne wartości ujemne, ale także o nierówność trójkąta (sprawdź, czy najdłuższy bok jest krótszy od sumy dwóch pozostałych).

Zainicjalizuj pola numeryczne w formularzu atrybutem `value` – wartość tego atrybutu ma być ustawiana parametrami `a`, `b` i `c` przekazanymi przez formularz, ale pierwsze uruchomienie servletu (jeszcze przy braku parametrów `a`, `b` i `c`) ma być ustawione za pomocą wartości przekazanych przez parametry inicjalizacji samego servletu w znaczniku `<init-param>` wewnątrz znacznika `<servlet>` w pliku `web.xml`. Odczyt tych wartości w servlecie można zrobić funkcją `getServletConfig().getInitParameter()`.

Dodatkowo na końcu wygenerowanego przez servlet dokumentu umieść informację o autorze (Twoje imię i nazwisko) i jego skrzynce mailowej. Informacje te należy umieścić w pliku `web.xml` jako parametry inicjalizacji kontekstu w znaczniku `<context-param>` a odczytać w servlecie za pomocą funkcji `getServletContext().getInitParameter()`.