Praca domowa 04 – ejb

Termin zwrotu : 14 listopada godz. 23.00
Zadanie uznaje się za zaliczone, gdy praca oceniona zostanie na co najmniej 6 pkt.

Należy stworzyć (zaimplementować) z wykorzystaniem technologii EJB komponent o nazwie *NprimeBean.* Udostępniona poprzez interfejs o nazwie *NprimeRemote* metoda *prime* komponentu *NprimeBean* otrzymuje jako parametr liczbę naturalną *n*. Metoda zwraca wartość liczby pierwszej o postaci *4k+3*.

Należy stworzyć (zaimplementować) z wykorzystaniem technologii servletów komponent o nazwie *Nprime.* Servlet otrzymuje jako dane wejściowe parametr o nazwie *n*, który przekazywany jest w żądaniu (url). Odpowiedź zawiera wyznaczoną przez metodę *prime(n)* komponentu EJB *NprimeBean* wartość.

Proces kompilacji musi być możliwy z użyciem komendy

```
javac –extdirs <path-to-appserver>/lib –Xlint NprimeBean.java NprimeRemote.java Nprime.java
```

Żądanie z wykorzystaniem metody POST protokołu http winno zwrócić wyznaczoną przez komponent liczbę pierwszą (o określonych wyżej właściwościach), najmniejszą z liczb większych od *n* (gdzie *n* jest wartością parametru żądania). Żądanie z użyciem metody GET winno zwrócić wartość liczby pierwszej (o określonych wyżej właściwościach), największą z liczb mniejszych lub równych od *n.* (można w pełni wykorzystać odpowiednio uproszczony servlet utworzony uprzednio w ramach zadania 06).

Np. dla n = 100 żądanie POST winno zwrócić wartość 103 (bo 4 * 25 + 3 = 103), natomiast żądanie GET wartość 83 (bo 4 * 20 + 3 = 83).

Rozwiązanie testowane będzie w środowisku serwera aplikacyjnego GlassFish 4. Zawartość pliku web.xml, który używany będzie trakcie uruchamiania i testowania komponentu podano niżej :

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns="http://java.sun.com/xml/ns/javaee"
       xmlns:web="http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
       xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
       id="WebApp_ID" version="3.0">
  <servlet>
    <servlet-name>servletNNNNN</servlet-name>
    <servlet-class>Highway</servlet-class>
  </servlet>
```

```
  <servlet-mapping>
    <servlet-name>servletNNNNN</servlet-name>
    <url-pattern>/*</url-pattern>
  </servlet-mapping>
</web-app>
```

Wymagania :

- Klasa implementująca servlet winna zostać zdefiniowane w pliku `Nprime.java`, klasa implementująca komponent EJB w pliku `NprimeBean.java`
- Należy zwrócić uwagę, że parametry url'a zawierać mogą napisy złożone z liter cyfr oraz znaków specjalnych, które kodowane będą z wykorzystaniem UTF-8.
- W pliku README.pdf winien być zawarty szczegółowy opis organizacji struktur danych oraz szczegółowy opis zastosowanego algorytmu obliczeniowego.
- Proces obliczenia rozwiązania winien się kończyć w czasie nie przekraczającym 1 min (orientacyjnie dla typowego notebooka). Po przekroczeniu limitu czasu zadanie będzie przerywane, i traktowane podobnie jak w sytuacji błędów wykonania (czyli nie podlega dalszej ocenie).

Sposób oceny **:**

- 1 pkt – **Weryfikacja** : czy program jest skompletowany i spakowany zgodnie z ogólnymi zasadami przesyłania zadań.
- 1 pkt – **Kompilacja** : każdy z plików winien być kompilowany bez jakichkolwiek błędów lub ostrzeżeń (w sposób omówiony wyżej)
- 1 pkt – **Wykonanie** : program powinien wykonywać się bez jakichkolwiek błędów i ostrzeżeń (dla pliku danych wejściowych zgodnych z wyżej zamieszczoną specyfikacją) z wykorzystaniem omówionych wyżej parametrów linii komend
- 2 pkt – **README** : plik README.pdf dokumentuje w sposób kompletny i właściwy struktury danych, oraz opis przyjętej koncepcji algorytmu
- 1 pkt – **Styl kodowania** : czy funkcji i zmienne posiadają samo-wyjaśniające nazwy ? Czy podział na funkcje ułatwia czytelność i zrozumiałość kodu ? Czy funkcje eliminują (redukują) powtarzające się bloki kodu ? Czy wcięcia, odstępy, wykorzystanie nawiasów itp. (formatowanie kodu) są spójne i sensowne ?
- 4 pkt – **Poprawność algorytmu** : czy algorytm został zaimplementowany poprawnie a wynik odpowiada prawidłowej (określonej zbiorem danych testowej) wartości.

Przykład prostego rozwiązania (EJB + servlet)**:**

http://javahowto.blogspot.com/2007/06/simple-ejb-3-application-hand-made.html
http://javahowto.blogspot.com/2007/07/simple-ejb-3-servlet-application.html