

Praca domowa 08 – jaxrs

Termin zwrotu : 07 stycznia godz. 23.00

Zadanie uznaje się za zaliczone, gdy praca oceniona zostanie na co najmniej 6 pkt.

Na serwerze aplikacyjnym Glassfish 4 w kontenerze *ejb* zainstalowany jest pod nazwą *ejb-project* (deployment descriptor) komponent (stateful session bean) o nazwie *DbManager* wraz z interfejsem *IDbManager*, który zdefiniowany jest następująco :

```
package pl.jrj.db;
import javax.ejb.Remote;

@Remote
public interface IDbManager {
    public boolean register(int hwork, String album);    // hwork - numer zadania, album - numer albumu studenta
}
```

Metoda *register* dokonuje rejestracji użytkownika w systemie zwracając **true** jeżeli proces rejestracji zakończył się poprawnie. Jeżeli rejestracja zakończyła się niepowodzeniem, metoda *register* zwraca wartość **false**.

Należy stworzyć w oparciu o technologię EJB usługę sieciową pracującą w standardzie WebService's (JAX-RS). Usługa winna zwracać ilość jednostek podziału terytorialnego kraju na poszczególnych stopniach hierarchii, a więc np. ilość województw, na które podzielono terytorium Polski, ilość powiatów w ramach wskazanego województwa, ilość gmin w określonym województwie i powiecie. Ponadto usługa winna zapewnić możliwość otrzymania listy wskazanych obiektów w formacie JSON.

Aktualny wykaz podziału administracyjnego kraju dostępny jest na stronie <http://www.stat.gov.pl/broker/access/prefile/listPreFiles.jspa> w postaci przeznaczonej do automatycznego przetwarzania (link do spakowanego pliku w formacie xml).

Program ma być zapisany w plikach : *Country.java* zawierającym implementację komponentu, *IDbManager.java* zawierającym definicję interfejsu komponentu *DbManager* oraz ewentualnie pozostałych – wykonanych dla potrzeb zadania – elementów rozwiązania. Poszczególne elementy rozwiązania nie mogą korzystać z bibliotek zewnętrznych innych niż niezbędne moduły serwera (jak np. *javaee.jar* itp.). Na serwerze aplikacyjnym dostępna jest biblioteka *gson-1.7.2.jar* niezbędna do operowania obiektami JSON.

Proces kompilacji musi być możliwy z użyciem komendy

```
javac -cp <app-server-modules> -Xlint Country.java IDbManager.java *.java
```

Zawartość pliku web.xml, który używany będzie w trakcie uruchamiania i testowania usługi 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>Jersey Web Application</servlet-name>
    <servlet-class>org.glassfish.jersey.servlet.ServletContainer</servlet-class>
    <init-param>
      <param-name>jersey.config.servlet.provider.webapp</param-name>
      <param-value>>true</param-value>
    </init-param>
    <load-on-startup>1</load-on-startup>
  </servlet>
  <servlet-mapping>
    <servlet-name>Jersey Web Application</servlet-name>
    <url-pattern>/*</url-pattern>
  </servlet-mapping>
</web-app>
```

Proces uruchamiania usługi musi być możliwy z użyciem poleceń postaci :

http://<host:port>/<app-name>/country	- wyznacza ilość województw
http://<host:port>/<app-name>/country/XX	- wyznacza ilość powiatów w województwie
http://<host:port>/<app-name>/country/XX/YY	- wyznacza ilość gmin w obrębie wskazanego województwa i powiatu
http://<host:port>/<app-name>/country/XXYY	- wyznacza ilość gmin w obrębie wskazanego województwa i powiatu

gdzie **XX** wskazuje kod poszukiwanego województwa a **YY** kod powiatu w obrębie województwa. W przypadku poleceń postaci

```
http://<host:port>/<app-name>/country/L
http://<host:port>/<app-name>/country/XX/L
http://<host:port>/<app-name>/country/XX/YY/L
http://<host:port>/<app-name>/country/XXYY/L
```

usługa winna zwrócić tekstową reprezentację obiektu JSON prezentującą listę nazw żądanych obiektów (odpowiednio : listę właściwych województw, powiatów lub gmin).

Wymagania :

- Klasa implementująca aplikację winna zostać zdefiniowana w pliku `Country.java`.
- Interfejs umożliwiający poprawną rejestrację zadania winien zostać zdefiniowany w pliku `IDbManager.java`.
- W pliku `README.pdf` winien być zawarty opis mechanizm operowania danymi oraz algorytm wyznaczania wyniku.
- 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 sposób zestawiania połączenia
- 1 pkt – **Styl kodowania** : czy funkcje 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.