

Praca domowa 12 – mdb

Termin zwrotu : 19 czerwca godz. 23.00

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

Należy zaprojektować i wykonać komponent typu MDB (Message Driven Bean) o nazwie *MdbBean* nasłuchujący komunikatów tekstowych. Poprawny komunikat tekstowy obsługiwany przez komponent przyjmuje postać:

```
start
stop
counter
error
increment
increment/n
```

Komponent może znajdować się w jednym z dwóch stanów : *zliczania* lub *wstrzymania*. Początkowo komponent znajduje się w stanie *wstrzymania* a licznik komponentu jest wyzerowany. Przejście ze stanu *wstrzymania* do stanu *zliczania* następuje w efekcie wywołania polecenia *start*. Przejście ze stanu *zliczania* do stanu *wstrzymania* następuje w efekcie wywołania polecenia *stop*. Wykonanie polecenia *increment* powoduje zmianę wartości licznika o 1, polecenia *increment/n* powoduje zmianę wartości licznika o wartość *n*. Wykonanie polecenia zmieniającego stan licznika dopuszczalne jest wyłącznie w stanie *zliczania*. Wykonanie metody *increment/n* w stanie *wstrzymania* nie zmienia wartości licznika, powoduje natomiast zarejestrowanie faktu niepoprawnego żądania poprzez zwiększenie licznika błędów (error). Podobnie wywołanie metody *start* w stanie *zliczania* lub metody *stop* w stanie *wstrzymania* jest operacją niepoprawną, skutkującą wyłącznie zwiększeniem licznika błędów (error). Odebranie komunikatu o treści innej, niż obsługiwana przez komponent powoduje zwiększenie licznika błędów (error). Metoda *counter* zwraca aktualny stan licznika, metoda *error* zwraca informację o łącznej ilości nieprawidłowych wywołań (stan licznika błędów error).

Komunikaty dostarczane są z użyciem zdefiniowanego zasobu o nazwie *jms/MyQueue*. Wyniki działania komponentu (stan licznika) przesyłane są zwrótnie w postaci tekstowej poprzez umieszczenie w zasobie o nazwie *jms/MyTopic* komunikatu tekstowego o składni

```
<album>/n
```

gdzie *<album>* oznacza numer albumu a *n* jest zwracaną wartością. Oba zasoby (*jms/MyQueue* oraz *jms/MyTopic*) udostępniane są poprzez *jms/MyConnectionFactory*.

Program ma być zapisany w pliku *MdbBean.java*. Poszczególne elementy rozwiązania nie mogą korzystać z bibliotek zewnętrznych innych niż niezbędne moduły serwera (jak np. *gf-client.jar*, *javaee.jar* itp.).

Proces kompilacji musi być możliwy z użyciem komendy

```
javac -cp <app-server-modules> -Xlint MdbBean.java
```

Wymagania :

- Klasa implementująca komponent winna zostać zdefiniowana w pliku `MdbBean.java`.
- W pliku README.pdf winien być zawarty opis architektury proponowanego rozwiązania.
- 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.