

Praca domowa 4 – gauss

Termin zwrotu : 7 kwietnia godz. 23.00

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

Należy stworzyć (zaimplementować) z wykorzystaniem technologii servletów komponent (servlet) o nazwie *Gauss*. Servlet otrzymuje jako dane wejściowe serię żądań typu GET z parametrem *n*. Parametr *n* przyjmuje wartości rzeczywiste. Otrzymanie żądania typu POST oznacza, że proces zbierania danych został zakończony.

Otrzymanie przez servlet żądania typu POST oznacza zakończenie fazy zbierania danych oraz polecenie przejścia do fazy analizy danych i wyznaczenia wyniku. Zadaniem servletu jest sprawdzenie czy rozkład zebranych danych jest rozkładem normalnym. Program winien w wyniku zwrócić najmniejszą wartość poziomu istotności (z dokładnością do dwóch miejsc dziesiętnych) na którym hipotezy o zgodności z rozkładem normalnym odrzucić nie można. Dla potrzeb analizy statystycznej można wykorzystać Apache Commons Math 3.4.1 API.

Proces kompilacji w środowisku serwera aplikacyjnego musi być możliwy z użyciem komendy

```
javac -extdirs <path-to-appserver>/lib -Xlint Gauss.java
```

Uruchomienie programu w środowisku serwera aplikacyjnego musi być możliwe wyłącznie z wykorzystaniem dwóch plików:

```
WEB-INF/classes/Gauss.class  
WEB-INF/web.xml
```

Zawartość pliku web.xml, który używany będzie w 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>Gauss</servlet-class>  
  </servlet>  
  <servlet-mapping>  
    <servlet-name>servletNNNNN</servlet-name>  
    <url-pattern>*/</url-pattern>  
  </servlet-mapping>  
</web-app>
```

gdzie **NNNNN** oznacza numer albumu studenta, którym sygnowana jest praca.

Program ma być zapisany wyłącznie w pojedynczym pliku `Gauss.java` zawierającym kod servletu implementującego mechanizm poszukiwania rozwiązania. Program nie może korzystać z bibliotek zewnętrznych innych niż Apache Commons Math 3.4.1 API (org.apache.commons.math3), która jest zainstalowana i dostępna dla serwera aplikacyjnego w otoczeniu którego testowany będzie servlet.

Wynik działania programu winien być poprawnie sformatowaną liczbą o wartości 1 lub 0 (nie należy formatować i zwracać kompletnej strony html).

Wymagania :

- Klasa implementująca problem winna zostać zdefiniowana w pliku `Gauss.java`
- W pliku `README.pdf` winien być zawarty szczegółowy opis organizacji struktur danych oraz szczegółowy opis zastosowanego mechanizmu (metody) poszukiwania rozwiązania z opisem mechanizmu zapewnienia osiągnięcia zakładanej dokładności wyznaczania wyniku.
- Proces wyznaczania 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 funkcje i zmienne posiadają samo-wyjaśniające nazwy ? Czy podział na funkcje ułatwia czytelność i zrozumiałość kodu ? Czy funkcje eliminują (redukuja) 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.