

Praca domowa 05 – integral

Termin zwrotu : 27 listopada 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 (stateless session bean) o nazwie *FuncMonitor* implementujący interfejs *IFuncMonitor*, który zdefiniowany jest następująco :

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

@Remote
public interface IFuncMonitor {
    public double f( double x, double y );
}
```

Metoda $f(x,y)$ zwraca wartość pewnej funkcji dwóch zmiennych. Należy stworzyć (zaimplementować) z wykorzystaniem technologii EJB komponent o nazwie *SIntegral*. Udostępniona poprzez interfejs o nazwie *ISIntegralRemote* metoda *solve* komponentu *SIntegral* otrzymuje jako parametry liczbę naturalną n oraz stałe a i b . Metoda $solve(a,b,n)$ zwraca wyznaczoną z dokładnością nie mniejszą niż 10^{-n} wartość całki

$$I = \int_0^a \int_0^b f(x,y) dx dy$$

gdzie stałe a oraz b przyjmują wartość dodatnią (format stałoprzecinkowy).

Należy stworzyć (zaimplementować) z wykorzystaniem technologii servletów komponent o nazwie *Solver*. Servlet otrzymuje jako dane wejściowe parametry o nazwie n , a oraz b (kolejność parametrów dowolna) przekazywane w żądaniu (url). Odpowiedź zawiera wyznaczoną przez metodą $solve(a,b,n)$ komponentu EJB *SIntegral* wartość całki.

Proces kompilacji musi być możliwy z użyciem komendy

```
javac -extdirs <path-to-appserver>/lib -Xlint SIntegral.java ISIntegralRemote.java Solver.java IFuncMonitor.java
```

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>servletNNNN</servlet-name>
  <servlet-class>MGame</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>servletNNNN</servlet-name>
  <url-pattern>/*</url-pattern>
</servlet-mapping>
</web-app>

```

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

Wymagania :

- Klasa implementująca komponent całkujący winna zostać zdefiniowana w pliku `SIntegral.java`.
- Interfejs udostępniający metodę realizującą algorytm w komponencie `SIntegral.java` winien zostać zdefiniowany w pliku `ISIntegralRemote.java`.
- Servlet nadzorujący proces obliczeń zapisać należy w pliku `Solver.java`
- W pliku `README.pdf` winien być zawarty opis mechanizmu wyszukiwania (lookup) i zestawiania połączenia oraz uzasadnienie zastosowanego rozwiązania zapewniającego osiągnięcie zakładanej dokładności obliczeń.
- Proces obliczenia rozwiązania winien się kończyć w czasie nie przekraczającym 3 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 algorytm poszukiwania rozwiązania.
- 1 pkt – **Styl kodowania** : czy funkcje i zmienne posiadają samo-wyjaśniające nazwy ? Czy podział na funkcje ułatwia czytelność i zrozumiałość kodu ? Czy funkcje eliminują (redukuje) 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.