**Praca domowa 05**

**Jarosław Górski**

**Opis zadania**

Zadanie polega na zaimplementowaniu algorytmów w postaci komponentu EJB o nazwie **IBlockRemote.**java wraz z interfejsem **IDSManagerRemotep** potrzebnych do obliczenia objętości bryły ze zbioru punktów umieszczonych w SQL-owym repozytorium danych w tabeli, oraz z połączeniem do bazy danych przy użyciu usługi JNDI. Obliczenia przeprowadzane za pomocą servletu Solver.

**Założenia**

Wartości punktów pobierane z SQL-owego repozytorium danych w postaci (x,y,z).

Podstawa graniastosłupa to otoczka wypukła zbioru punktów na płaszczyźnie XY.

Objętość bryły powinna być zaokrąglona do 5 miejsc po przecinku.

Wysokość graniastosłupa to średnia wartość funkcji z=f(x,y) określonej nad zadanym zbiorem punktów

**Struktury ogólna**

Klasa Solver rozszerza klase **HttpServlet**. Na początku programu używamy metody *doGet()* która realizuje algorytm olibczajacy pole powierzchni bocznej bryły. Wymaga ona podanie parametru ,,t’’ w żądaniu, dzięki temu przeniesiona zostaje nazwa tabeli bazy danych, z której pobierane są informacje.

Metoda *doGet()* wywołuje również metode *solve()* inferfejsu **IBockRemote,** która oblicza objętość graniastosłupa.

Klasa **Block** implementuje interfejs **IBlockRemote** i zajmuje a się obliczaniem pola powierzchni bocznej zadanej bryły. Zawiera również metode *solve()*, która zajmuje się obliczeniem objętości graniastosłupa prostego dla podanych danych. Używa do tego algorytmu Grahama w celu wyszukania wypukłej otoczki oraz sposobu analitycznego liczenia pól wraz z obliczaniem wysokości graniastosłupa,

Klasa **punkt** użyta w pliku Block.java, reprezentująca punkt w przestrzeni. Posiada ona dodatkową metodę *getAlfa()* która oblicza kierunek odchylenia podanego parametrem.

**IBlockRemote** zawiera zdalny interfejs EJB, posiada on metodę *solve(),* z dwoma argumentami: nazwę źródła danych oraz nazwę tabeli, z której będą pobrane informacje o punktach w przestrzeni.

**Opis działania**

Na serwerze Glassfish znajduje się komponent EJB o nazwie DSManager. Komponent ten jest wyszukiwany poprzez przenośne JNDI (ang. Java Naming and Directory ). Interfejs JNDI pozwala odkrywać i wyszukiwać obiekty za pomocą nazw. Po uruchomieniu obiektu dsManager inicjuje się metoda *solve()* należąca do interfejsu IBlockRemote, pobiera dwa argumenty jeden pograny metodą getDS() a drugi przekazywany do metody *solve()* jako parametr ,,t’’ który w URL przekazuje żądania zawierające nazwę tabeli , z której będą pobierane dane punktów *setPoints*, która w celu uzyskania połączenia z bazą danych wywołuje metodę *getDatabaseConnection*() przekazując jej argument pobrany za pomocą interfejsu IDSManagerRemote.W celu z nalezienia obiektu należy podać: nazwę kontenera EJB w jakiej jest komponent EJB, nazwę klasy oraz Interfejs implementowany przez daną klasę wraz z pełną ścieżką pakietu. Pełna implementacja obliczająca objętość graniastosłupa prostego nad zadanym zbiorem punktów znajduje się w klasie Block. Do odnalezienia wypukłej otoczki zastosowano algorytm Grahama, który potrzebuje posortowanych wierzchołków. Kolejno zaczynając od drugiego wierzchołka z posortowanej listy, sprawdzało się jego odchylenie względem poprzedniego, jeśli odchylenie implikowało położenie wierzchołka na zewnątrz to wtedy ten wierzchołek należał do otoczki. Po wyznaczeniu otoczki zastosowano analityczną metodę liczenia pól do obliczenia pola podstawy graniastosłupa. Wysokość została wyznaczona zgodnie z założeniami zadania, a następnie w celu wyznaczenia objętości pomnożyliśmy otrzymane pola podstawy i wysokości i zaokrągliliśmy do 5 miejsca po przecinku .