

Praca domowa 5 – solid

Przemysław Kleszcz

Zadanie

Cel pracy domowej: Obliczyć powierzchnię wypukłej otoczki rzutu punktów przechowywanych w bazie danych na płaszczyznę XZ.

Organizacja strukturalna

Projekt zawiera cztery pliki: **Solver.java**, **Solid.java**, **ISolidRemote.java**, **IDSManagerRemote.java**

Solver.java:

```
public class Solver
    protected void doGet(HttpServletRequest req, HttpServletResponse resp)
    protected void doPost(HttpServletRequest req, HttpServletResponse resp)
    private void process(HttpServletRequest req, HttpServletResponse resp)
    private Connection getConnection()
    private List<Solid.Point> getData(Connection conn, String table)
```

Solver.java:

```
public class Solid
    public double calculate(List<Point> points)
    private double calculateArea(List<Solid.Point> points)
    private static Point getSmallestPoint(List<Point> points)
    private static Set<Point> getSortedSet(List<Point> points)
    private static Turn getDirection(Point a, Point b, Point c)
```

ISolidRemote.java:

```
public interface ISolidRemote
```

IDSManagerRemote.java:

```
public interface IDSManagerRemote
```

doGet(HttpServletRequest req, HttpServletResponse resp) - Przyjmuje jako parametry obiekt żądania oraz obiekt odpowiedzi. Metoda przetwarzająca żądania GET servletu.

doPost(HttpServletRequest req, HttpServletResponse resp) - Przyjmuje jako parametry obiekt żądania oraz obiekt odpowiedzi. Metoda przetwarzająca żądania POST servletu.

process(HttpServletRequest req, HttpServletResponse resp) – Przyjmuje jako parametry obiekt żądania oraz obiekt odpowiedzi. Wspólna metoda dla żądań GET i POST realizująca następny etap przetwarzania.

getConnection() – Metoda zwraca połączenie do źródła danych.

getData(Connection conn, String table) – Metoda przyjmuje jako parametry połączenie do źródła danych oraz nazwę tabeli. Zwraca wyodrębnione dane jako lista punktów.

calculate(List<Point> points) – Metoda przyjmuje jako parametr listę punktów. Metoda realizująca podstawowe założenie programu. Zwraca pole powierzchni otoczki.

calculateArea(List<Solid.Point> points) – Metoda przyjmuje jako parametr listę punktów. Metoda oblicza i zwraca pole powierzchni otoczki.

getSmallestPoint(List<Point> points) – Metoda przyjmuje jako parametr listę punktów. Zwraca punkt o najmniejszych współrzędnych z i x.

getSortedSet(List<Point> points) – Metoda przyjmuje jako parametr listę punktów. Zwraca posortowany rosnąco zbiór punktów wg kąta jaki tworzy połączenie danego punktu i punktu najmniejszego do prostej x.

getDirection(Point a, Point b, Point c) – Metoda przyjmuje jako parametr struktury opisujące trzy punkty układu współrzędnych. Sprawdza czy punkt zawiera się w otoczce wypukłej.

Opis mechanizmu

Program jest zrealizowany w konwencji servletu + EJB. Zadaniem metody *process*, która jest wspólną funkcjonalnością przetwarzania metod żądań jest pobranie parametru *t*, który stanowi nazwę tabeli przechowującej dane. W następnym etapie wykorzystywany jest komponent *DSManager*, który zwraca informację o nazwie źródła danych. W tym celu wykorzystywany jest interfejs JNDI. Dostęp do beana realizowany jest poprzez następującą konwencję:

java:global[/application-name]/<module-name>/<bean-name>!<fully-qualifiedinterface-name>

Fragment kodu obrazujący wykorzystanie:

```
Context cont = new InitialContext();
String n = "java:global/ejb-project/DSManager!pl.jrj.dsm.IDSManagerRemote";
IDSManagerRemote man = (IDSManagerRemote) cont.lookup(n);
```

Wykorzystując połączenie do źródła danych oraz nazwę tabeli, punkty określające położenie na układzie współrzędnych są odczytywane z źródła danych i umieszczane w liście, która następnie jest przekazywana do metody realizującej obliczenia.

Do rozwiązania zadania wykorzystany został algorytm Grahama, którego implementacja umieszczony została w osobnym komponencie o nazwie *Solid*. Algorytm wykorzystuje się do odnajdowania wypukłej otoczki punktów. Złożoność algorytmu to $O(n \log n)$. Algorytm polega na obliczeniu wyznacznika dla trzech punktów. Jeśli wyznacznik < 0 – punkt należy do otoczki wypukłej. Jeśli wyznacznik > 0 – punkt zawiera się wewnątrz otoczki. Wynikiem jest zbiór punktów składających się na otoczkę. Posiadając zbiór punktów składających się na otoczkę obliczane jest pole powierzchni wielokąta, który tworzy. Obliczane jest wg wzoru:

$$area = \left| \frac{(x_1 z_2 - z_1 x_2) + (x_2 z_3 - z_2 x_3) \dots + (x_n z_1 - z_n x_1)}{2} \right|$$

Obliczona wartość jest wynikiem działania programu i umieszczana jest w odpowiedzi servletu.