



Exercise 8: Programming Geometrical Problems & Open GeoSpatial Consortium (OGC) Specification

Part I: Programming geometrical problems using Java / JTS

Aim

The aim of this exercise is to write Java program for solving geometrical problems and learn how to use a Java class library JTS.

Introduction

The first task is to learn the basic of JTS Topology Suite (JTS). JTS is an API providing spatial object model and fundamental geometric functions; and it implements the geometry model defined in the OGC *Simple Features Specification for SQL*. Functions provided include:

- topological relationships,
- overlay functions (difference, union, symmetric difference),
- buffer,
- convex hull,
- area and distance functions, and
- topological validity checking.

To be able to use JTS classes in the Java program the classes must be imported to your Java program. In this exercise you will add the Java classes in your home directory. Do the following:

1. Make a new directory for this exercise in your own place, for example z:\E10. You will need the following file: *PointTest.java*. It is available here:

<http://www.infra.kth.se/courses/AG2411/PointTest.java>

Save it to your directory Z:\E10.

2. The file *PointTest.java* is a simple example of using JTS. Try to understand what the program does by reading the code and studying the documentation of the JTS object classes. JTS is installed on your computer and its documentation can be found at:

C:\jts17\doc

3. Compile and run the file *PointTest.java* (and later on your own programs) using the following commands from the command prompt:

```
z:\E10> javac PointTest.java
z:\E10> java PointTest 1 2 3 4 5 6 7 8
```

4. If you would like to refresh your Java knowledge or you need help while programming, you can check these pages:

- Java tutorial: <http://java.sun.com/docs/books/tutorial/> ,
- API specification for Java 1.5.0: <http://java.sun.com/j2se/1.5.0/docs/api/> ,
- JTS: <http://www.vividsolutions.com/jts/jtshome.htm> .

Tasks

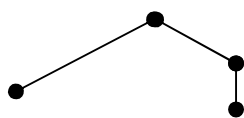
Do the following tasks:

- 1) Compile and run the program *PointTest.java*. Does the program do what you expect it to do?
- 2) Write a Java program. The program should create three lines (LineString; *a*, *b*, *c*) with the coordinates (x,y):
 - a) 12, 2; 12, 13; 12, 19
 - b) 2, 10; 10, 10; 21, 11
 - c) 1,1; 9,9; 20, 20

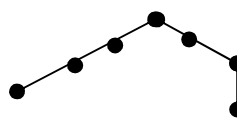
Then the program should tell which lines that intersect line *a*. Write the result in the command window.

- 3) Write a Java program that creates a line with the following coordinates (x,y):
12, 2; 12, 13; 12, 19; 19,19; 20, 20; 25, 25; 27, 18

Then the program should add coordinates on the line so that no line segment on the line is longer than 5 units. This is done by following procedure. Create a new line object. Copy each of the points from the original line to the new line. If the line segment is longer than 5 units, one (or more) points have to be added on the line segment in the new line (see figure below).



Original line



New line

Finally, the program should print all the coordinates of the new line object.

Report

A written report (of 2 pages) should be handed in to the exercise supervisor. The report should include your major findings in the programming as well as some personal comments of the use of JTS (e.g. if you think it is good or not). The code for your programs (2 and 3 above) should be appended to the report.

Part II: OGC Simple Feature Specification for SQL

Aim

The major aim of this exercise is to receive knowledge about the OGC (Open Geospatial Consortium) standard *Simple Feature Specification for SQL*. A secondary aim is general familiarity to read standards in the field of GIS.

Tasks

Read the following two documents:

- *Simple Feature Specification for SQL*
(available at <http://www.opengeospatial.org/docs/99-049.pdf>)
- *PostGIS manual*
(available at <http://postgis.refractory.net/documentation/>)

PostGIS was also used in the second exercise: Extended SQL. You can get some information by looking at this exercise and also test the web application used in this exercise.

Write a report (of about 2 pages) of the major findings in these two documents. The report should consider the following issues:

- What is the main reason for the standard *Simple Feature Specification for SQL*?
- What kind of content is *Simple Feature Specification for SQL* containing?
- Give example of methods that *Simple Feature Specification for SQL* defines for different geometries. How are these methods defined?
- What is the relationship between *PostGIS* and *Simple Feature Specification for SQL*?

Report

A written report (of about 4-5 pages) should be sent to the TA.