

Rapport Mini-Projet 2

Introduction

Notre projet consiste à développer une API java, qui permet , à partir de géométries ponctuelles de créer un polygone contenant ces points, en utilisant différents types des enveloppes comme l'enveloppe convexe, le minimal bounding box et Oriented Minimal Bounding box et d'exporter les résultats dans fichier shapefile. Dans ce rapport, on commencera en premier temps ,par présenter , les api utilitaires et les outils technologiques qu'on a utilisé, en deuxième temps , on décrira la conception du projet par un diagramme de classe, ensuite on présentera les algorithmes implémentés et les résultats générés. On finit notre rapport par une conclusion.

API utilitaires et outils

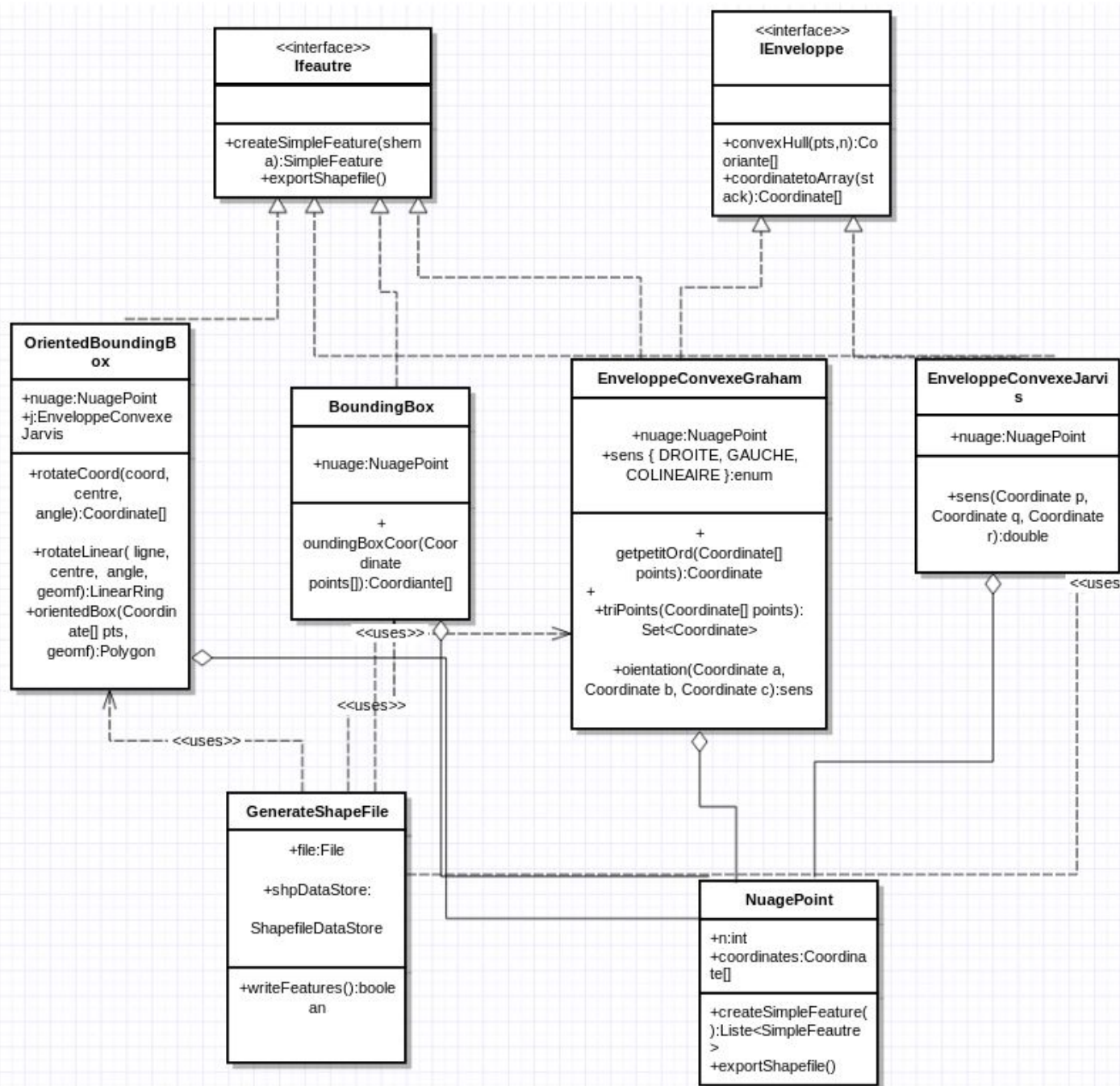
GeoTools : On a utilisé la librairie **GeoTools** parce qu'elle est plus compatible au langage java que d'autres librairies (**GDAL** par exemple) et elle supporte **JTS topology** suite qui est une librairie en java offrant des fonctionnalités pour la manipulation des géométries.

Eclipse et maven Notre environnement de développement est Eclipse. C'est un IDE conçu pour java et très simple à utiliser . Notre projet se présente sous forme de projet Maven. Maven permet d'automatiser la gestion de projet et ajouter les dépendances qu'on a besoin.

JUNIT 4 : Pour les tests unitaires, nous avons utilisé la framework de test conçu pour java.

QGIS : Pour visualiser les résultats en shapefile.

Conception : Diagramme de classe



Implémentation

Enveloppe convexe :

Une enveloppe convexe d'un nuage de points est le plus petit ensemble convexe qui contient ces points.

Pour créer cette enveloppe, nous avons utilisé deux approches algorithmiques :

Graham Scan:

On commence par :

- 1) Identifier le point le plus bas P , c'est-à-dire le point avec le plus petit y . S'il existe deux points avec le même y , on choisit le point avec le plus petit x .
- 2) Les points sont alors triés en ordre croissant d'angle par rapport à l'axe des abscisses. S'il y a plusieurs points avec le même angle, on choisit celui qui est le plus près de P .
- 3) Pour chaque séquence de trois points, si ces points forment un sens à gauche alors on passe à la séquence de trois points suivantes. S'ils constituent un sens à droite ou s'ils sont colinéaires, alors le deuxième point n'appartient pas à l'enveloppe convexe. On conserve les deux autres points dans la séquence suivante.

Marche de Jarvis :

L'approche de la marche de Jarvis est plus simple et facile à implémenter que celle de Graham. L'approche consiste à :

- 1) Identifier un point P qui en général le point le plus bas (le point avec le plus petit y)
- 2) A partir de ce point P , on cherche un deuxième point C tel que tous les autres points soient à gauche, a partir de ce point C , on cherche un autre point tel que tout les autres points soient à gauche. On répète la même méthode jusqu'à revenir au point de départ P .

Minimal Bounding Box

Le minimal Bounding Box d'un nuage de points est le rectangle minimal qui enveloppe tous les points.

Le principe est très simple : On cherche la valeur maximale d'abscisse x X_{max} la valeur minimale X_{min} , de même, la valeur maximale de l'ordonnée y : Y_{max} et la valeur minimale : Y_{min}

Les coordonnées de quatre sommets de rectangle sont respectivement : (X_{min}, Y_{max}) , (X_{max}, Y_{min}) , (X_{max}, Y_{max}) , (X_{min}, Y_{min}) .

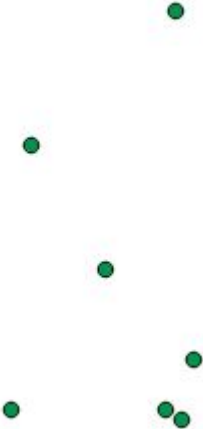
Oriented Minimal Bounding Box :

A partir des sommets l'enveloppe convexe :

- 1) On identifie le centre c de l'enveloppe en utilisant la fonction **getcentroid()**.
- 2) On calcule l'angle θ qui fait chaque sommet de l'enveloppe avec l'axe des abscisses en utilisant la fonction **Math.atan2()**
- 3) On fait une rotation de centre c et d'angle θ de l'enveloppe , et on calcule l'aire de la surface générée.
- 4) On refait les étapes précédentes jusqu'à avoir la surface minimale du rectangle qui englobe tout les points.

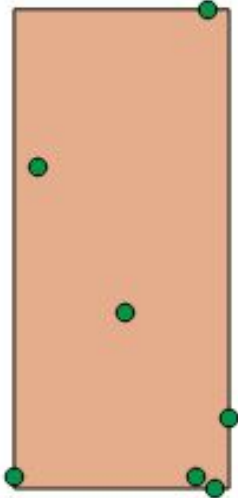
Résultats

Nuage de Points

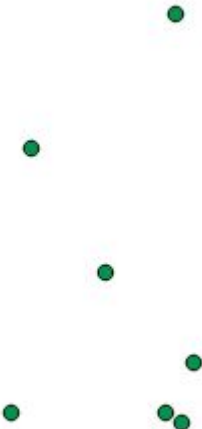


====>

BoundingBox

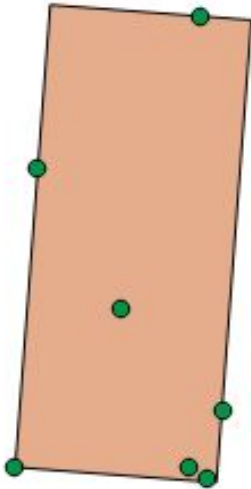


Nuage de Points
Bounding Box



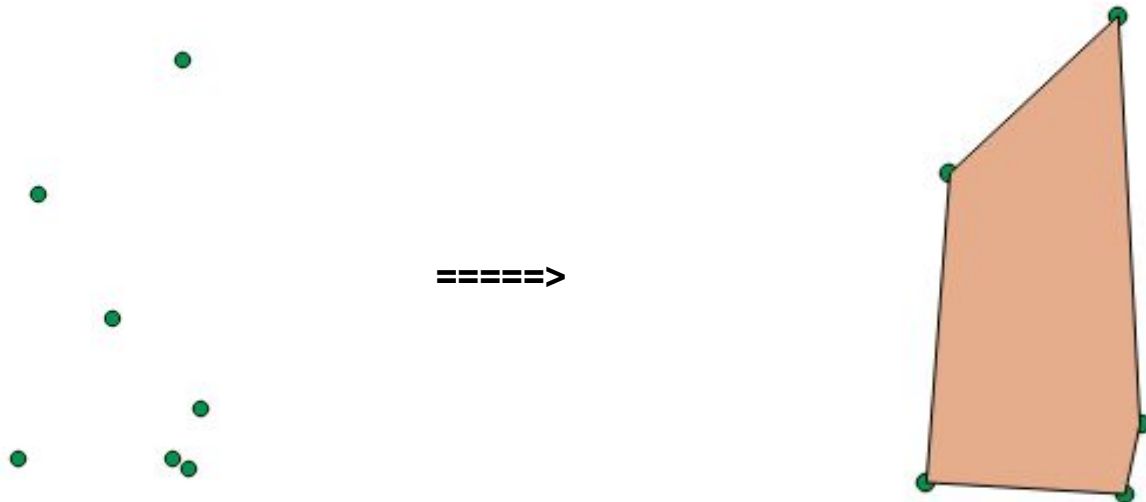
====>

Minimal Oriented



Nuage de Points

Enveloppe Convexe



Conclusion

Dans ce projet , nous avons crée une API java qui permet de faire les fonctionnalités de la création d'une enveloppe convexe avec deux algorithmes différents(Jarvis et Graham) et le minimal bounding box et le oriented bounding box et les exporter en Shapefile.

Comme améliorations, des autres approches algorithmiques peuvent être ajoutés pour l'enveloppe convexe (algorithme de chan, quick hull..) ou le oriented minimal box(rotation des axes..).

La difficulté rencontrée est le calcul l'enveloppe concave, les approches algorithmiques sont très complexes.

Références :

<https://sourceforge.net/p/opencarto/code/HEAD/tree/trunk/server/src/main/java/org/opencarto/algo/base/SmallestSurroundingRectangle.java?SetFreeDomCookie>

<http://www.sanfoundry.com/java-program-to-implementation-jarvis-algorithm/>

<https://www.geeksforgeeks.org/convex-hull-set-2-graham-scan/>

<https://github.com/ianturton/geotools-cookbook/blob/master/modules/output/src/main/java/org/ianturton/cookbook/output/WriteShapefile.java>

<http://docs.geotools.org/stable/tutorials/feature/csv2shp.html>

<http://docs.geotools.org/latest/userguide/tutorial/feature/csv2shp.html>