

## Feuille TP n° 4 : Manipulation des patrons et de la STL

Chaque TP est divisé en deux phases qui doivent, dans la mesure du possible, être traitées simultanément. La première phase consiste en l'implémentation pure de la classe et de ses méthodes. La seconde phase s'attachera à l'analyse et à l'amélioration du code.

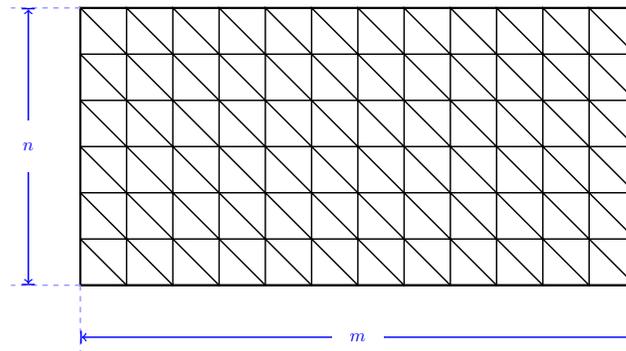
L'objectif de ce TP est de travailler sur la notion de patron et de manipuler différents éléments de la STL.

Vous devrez en particulier prêter attention à ne pas rendre votre code trop dépendant de la nature du conteneur utilisé.

**Une attention toute particulière sera prêter à la complétude des classes : les classes ne doivent contenir que les opérateurs et fonctions qui ont un sens de le contexte étudié.**

### Phase 1 : Implémentation

1. Implémenter une classe `Point` bidimensionnel dont le type sera soit le `double`, soit le `float`.
2. Implémenter une classe `Triangle` constituée de trois points.
3. Implémenter une classe `Maillage` constituée de triangles. En particulier, cette classe devra contenir une méthode permettant de générer un maillage à partir des nombres  $m$  et  $n$  de segments selon les directions  $x$  et  $y$ .



4. Ecrire une fonction d'enregistrement du maillage dans un fichier : l'enregistrement est réalisé par triangle en écrivant sur 4 lignes les coordonnées des points du triangle, le premier point étant répété en 4eme position, chaque triangle étant séparé par une ligne blanche.

Le fichier résultant du maillage d'un carré par 2 triangles aura donc la structure suivante.

```

0. 0. 1
1. 1. 2
1. 0. 3
0. 0. 4
5
0. 0. 6
0. 1. 7
1. 1. 8
0. 0. 9

```

On pourra par exemple l'afficher en utilisant la commande `gnuplot` suivante :

```
plot [-0.2:1.2][-0.2:1.2] "carre.dat" w l 1
```

5. Ecrire une fonction permettant la fusion de deux maillages qui ne se recouvrent pas.
6. Ecrire une fonction permettant d'effectuer une transformation affine d'un maillage.
7. Ecrire une fonction permettant le maillage d'un rectangle quelconque en s'appuyant sur les fonctions précédentes.

Les conteneurs utilisés devront être passés en paramètres (template) des classes.

### **Phase 2 : Analyse**

1. Etudier les performances de votre code en fonction de la taille du maillage.
2. Etudier les performances de votre code en fonction de la nature des conteneurs STL utilisés.
3. Etudier les performances de votre code en fonction de la nature des algorithmes STL utilisés.