TP d'Informatique Industrielle (STS SNIR 2ème année)

Documents autorisés

Une entreprise spécialisée dans la découpe de pièces (dans différentes matières) dispose d'un logiciel capable de disposer les pièces à découper de façon optimale pour limiter les chutes de matières. Cette entreprise désire maintenant optimiser son logiciel de découpe afin de calculer le prix à facturer au client de la manière la plus juste possible. Le coût de la découpe doit être fonction du périmètre à découper (pour tenir compte de l'usure de l'outil de découpe) et de la surface de la pièce découpée (pour tenir compte du coût de la matière première).

Dans sa version actuelle, le logiciel de gestion de la découpe permet de découper n'importe quelle forme polygonale ou circulaire.

Diverses classes en C++ vont être créées afin de modéliser cette application.

- Une classe **Commande** chargée de répertorier l'ensemble des figures à découper pour un client et de calculer le prix de la découpe (fonction du périmètre et de la surface des pièces découpées).
- Une classe **Figure** permettant de définir les caractéristiques générales de n'importe quelle figure géométrique *(classe abstraite)*
- Une classe **Polygone** permettant de modéliser un polygone.
- Une classe **Cercle**
- Une classe **Point2D** permettant de modéliser un point dans le plan

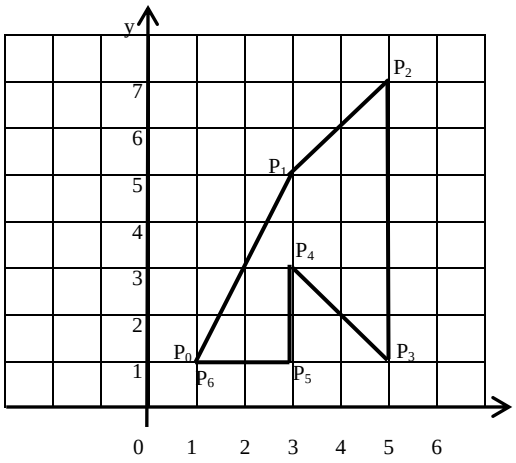
Formule permettant de calculer la surface d'un polygone quelconque (dont les lignes ne se croisent pas)

Soit un polygone constitué de ***n*** sommets numérotés ***0*** à ***(n-1)*** de coordonnées ***(x_i, y_i).*** Ajoutons aussi un sommet ***n*** identique au premier (pour symboliser la fermeture du polygone).

L'aire du polygone est :

$$A = \frac{1}{2}\left|\sum_{i=0}^{n-1}(x_i \,.\, y_{i+1} - x_{i+1} \,.\, y_i)\right|$$

Exemple :



Idem P_0

$P_0\begin{vmatrix}1\\1\end{vmatrix}$, $P_1\begin{vmatrix}3\\5\end{vmatrix}$, $P_2\begin{vmatrix}5\\7\end{vmatrix}$, $P_3\begin{vmatrix}5\\1\end{vmatrix}$, $P_4\begin{vmatrix}3\\3\end{vmatrix}$, $P_5\begin{vmatrix}3\\1\end{vmatrix}$, $P_6\begin{vmatrix}1\\1\end{vmatrix}$

$$A = \frac{1}{2}|(1*5-3*1)+(3*7-5*5)+(5*1-5*7)+(5*3-3*1)+(3*1-3*3)+(3*1-1*1)|$$

$$A = \frac{1}{2}|2-4-30+12-6+2| = 12$$

Questions préliminaires

2 pts **Question 1 :** Quelle(s) différence(s) y a-t-il entre un attribut privé, un attribut protégé et un attribut public ? Soyez le plus clair et complet possible en faisant par exemple un tableau comparatif.

1 pt **Question 2 :** A quoi sert la directive de précompilation `#pragma once` en haut des fichiers de déclaration des classes ? Par quoi pourrait-on remplacer cette directive ?

1 pt **Question 3 :** Que signifie la ligne `this->x = x;` située dans le constructeur de la classe Point2D ?

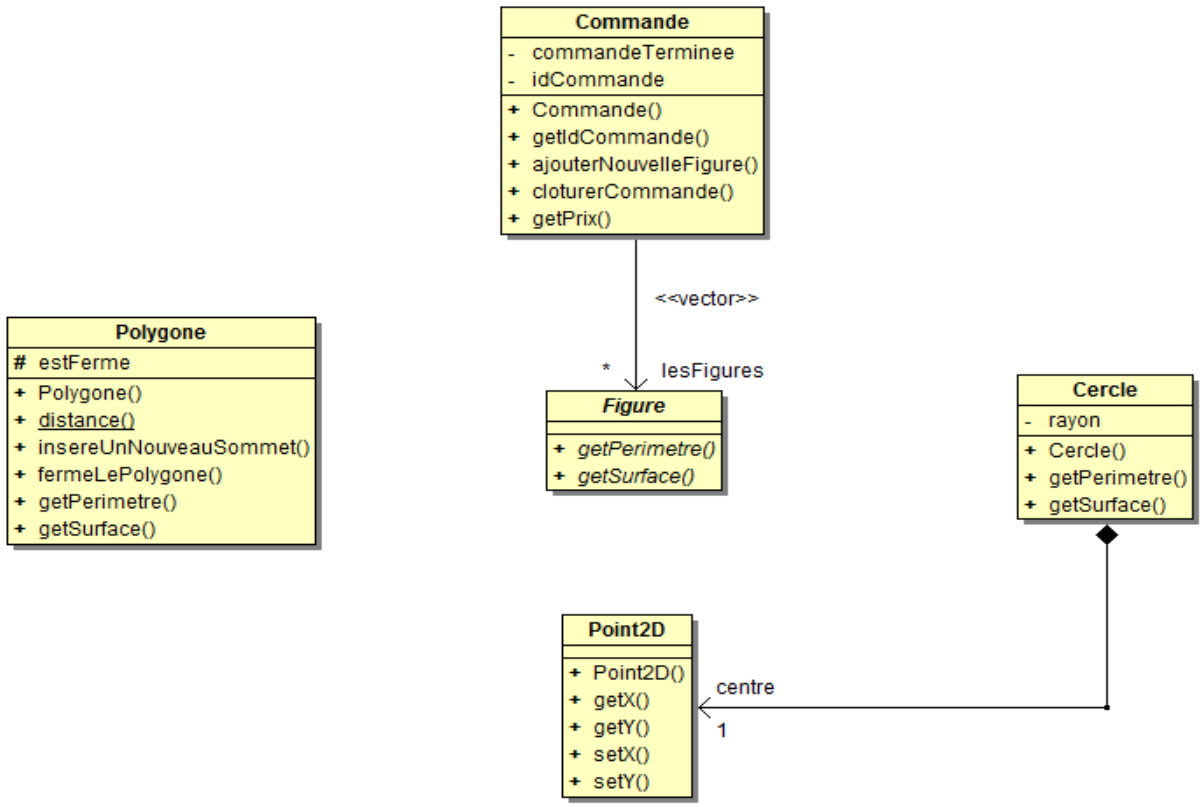
UML et C++

Voici ci-dessous un extrait de la classe Polygone

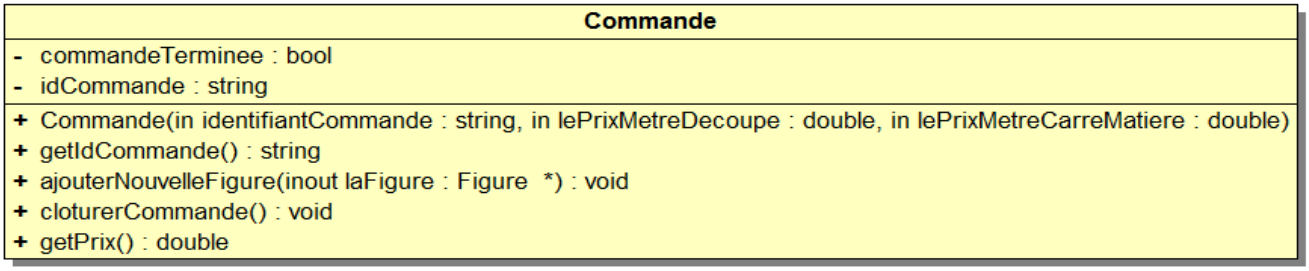
```
class Polygone
{
  protected:
    vector<Point2D *> lesSommets;
…
};
```

1 pt **Question 4 :** Ecrivez une phrase qui permet de décrire le lien entre la classe Polygone et la classe Point2D

2 pts **Question 5 :** En fonction de ce qui a été décrit dans le sujet et de la réponse à la question précédente, complétez le diagramme de modélisation UML de cette application en y indiquant les relations entre classes, les cardinalités et les rôles.



Précisions sur les classes :

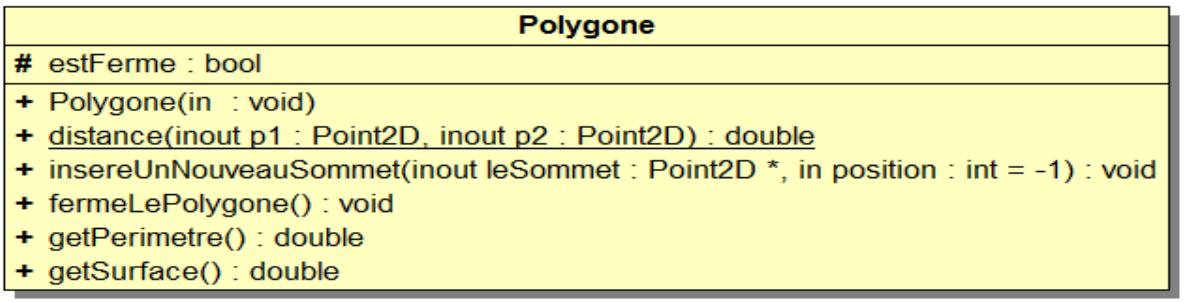


Attributs

- **CommandeTerminee** : drapeau initialisé à faux qui passe à vrai quand la commande est clôturée.
- **idCommande** : chaine de caractères identifiant la commande

Méthodes

- **Le constructeur** : admet en paramètres l'identifiant de la commande, le prix de la découpe au mètre linéaire et le prix du m^2 de matière.
- **getIdCommande** : renvoie l'identifiant de la commande
- **ajouterNouvelleFigure** : admet en paramètre un pointeur sur la figure à ajouter à la commande
- **cloturerCommande** : comme son nom l'indique, cette méthode clôture la commande
- **getPrix** : renvoie le prix total de la commande (périmètres des figures * prix de la découpe) + (surfaces des figures * prix du m^2 de matière). Cette méthode renvoie 0 si la commande n'est pas close.

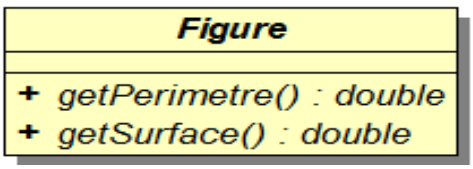


Attribut

- **estFerme** : drapeau initialisé à faux qui passe à vrai quand le polygone est fermé (à la fin)

Méthodes

- **distance** : méthode statique de classe permettant de calculer la distance entre 2 points (passés par référence).
- **insereUnNouveauSommet** : Insère le sommet passé en argument (par pointeur) à la position indiquée. Si la position est égale à -1 alors le sommet est ajouté à la fin du polygone.
- **fermeLePolygone** : ferme le polygone en rajoutant le 1er point à la fin



Classe abstraite comportant deux méthodes virtuelles pures : getPerimetre() et getSurface().

Le travail de codage sur machine commence à ce stade. Récupérez d'abord le projet à compléter sur le poste faisant office de serveur. Au endroits indiqués « Validation enseignant », vous devrez m'appeler pour que je contrôle votre travail et valide l'étape concernée.

Astuce de codage utile : Si A est une classe abstraite et si B est une sous classe concrète de A, on peut écrire :

```
A * objet ;  // on peut créer des pointeurs sur des classes abstraites

objet = new B ;  // on ne peut créer que des objets de classes concrètes

((B *)objet)->MethodeDeB() ;  // cast car la méthode de B n'est
                              // applicable que sur des objets B
```

3 pts

Complétez la classe Figure

Complétez la classe Cercle et écrivez un petit bout de programme principal ultra simple pour tester ses principales méthodes

Validation enseignant

5 pts

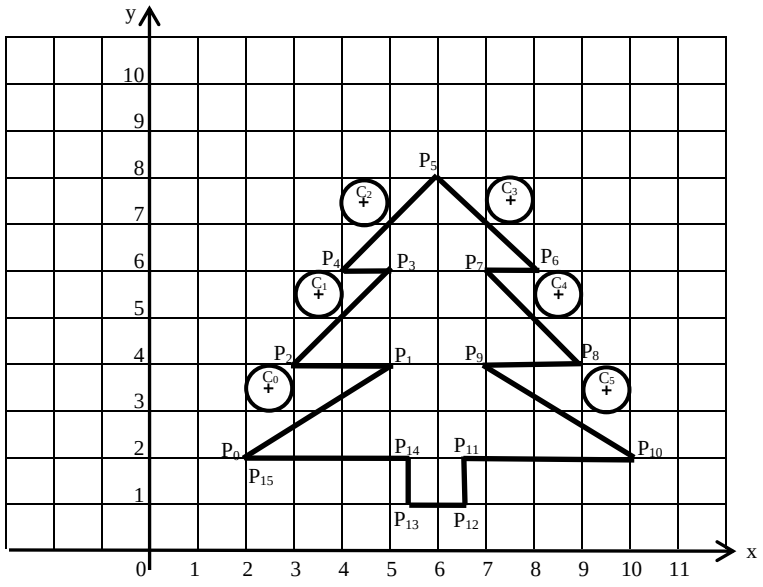
Complétez la classe Polygone et testez les méthodes calculant le périmètre et la surface en écrivant un petit programme principal modélisant la figure de la page 1. Vous devez trouver un périmètre de 20.129 et une surface de 12

Validation enseignant

La société vient de recevoir une commande urgente du Père Noël. Il s'agit d'une commande de 6 boules de Noël (symbolisées par des cercles) et d'un très grand sapin (7m de haut)

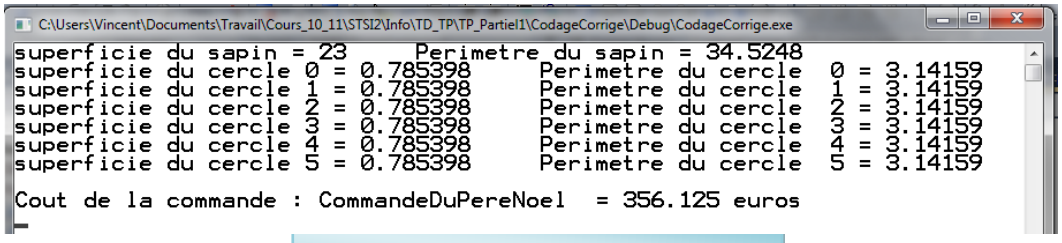
Le prix du mètre linéaire de découpe est de 0.26 € et le prix de la matière est de 12.35 €/m²

Le logiciel existant a placé les objets à découper comme sur le schéma ci-contre.



5 pts

Créez la classe Commande puis complétez le programme principal en suivant les commentaires indiqués afin de gérer la commande du père Noël.
Le programme principal final doit afficher ceci :



Validation enseignant