

TD n° 6

Encore des classes abstraites et des interfaces !

1 Forme géométrique et factorisation de code

Dans cet exercice, on manipule des formes géométriques que l'on définit par l'interface suivante.

```
1 interface FormeGeometrique {  
2     double perimetre();  
3     double surface();  
4 }
```

Exercice 1

1. Supposons que nous souhaitions écrire maintenant les classes `Cercle`, `Triangle`, `Rectangle` et `Carre`.

Dessinez l'arbre d'héritage des types mentionnés jusque ici dans l'exercice en précisant quelles classe contiennent quelles implémentations de méthodes.

2. Un polygone est la forme définie par une ligne brisée
 - qui ne se coupe pas elle-même ;
 - qui commence et termine au même point.Quel sera le genre de type de `Polygone` (classe ? abstraite ou non ? interface ? Argumentez en fonction de ce que vous prévoyez d'en faire.) ? Quelle sera sa relation avec les classes précédentes ?

3. Écrire le type `Polygone` (sans vérifier que la ligne brisée ne se coupe pas elle-même : ce n'est pas un exercice de mathématiques!).
 - Implémenter la méthode `perimetre()` ¹.
 - Déclarer la méthode `surface()` comme abstraite

Au vu de ce qui est demandé dans cette question, peut-on définitivement conclure le débat de la question précédente quant au type `Polygone` ?

4. Réécrire les classes de la première question en tenant compte de `Polygone`, lorsque cela a du sens.
5. Un polygone est convexe si, lorsque l'on trace un segment entre deux points quelconques du polygone, alors tous les points du segment appartiennent au polygone.

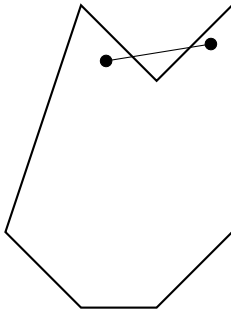
1. Si (x_1, y_1) et (x_2, y_2) sont deux points du plan, alors la distance les séparant est donnée par

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}.$$

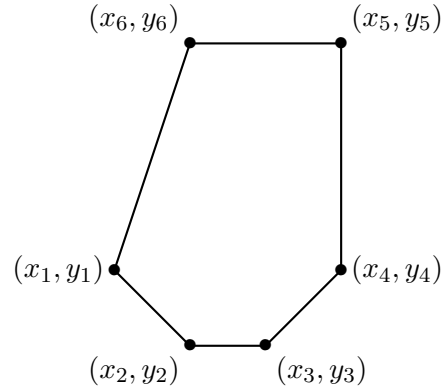
Sachant qu'on a une formule simple² pour calculer la surface d'un polygone convexe, on va programmer un type `PolygoneConvexe` qui contient une implémentation concrète de la méthode `surface()`. Du coup, quel genre de type ce doit être ?

Quelle est la relation de ce type avec les types déjà définis dans l'exercice ?

6. Programmez le type `PolygoneConvexe`³. Modifiez éventuellement les autres types afin de prendre en compte leurs nouvelles relations d'héritage/implémentation.



(a) Un polygone non-convexe



(b) Un polygone convexe

2 Quizz (Préparation du TP)

Le but de cet exercice est de faire un programme permettant de poser des quizz notés.

L'exécution du programme dans un terminal donnera quelque chose comme :

```

1 | Bonjour et bienvenu.
2 |
3 | >1) Quel est le nom du prof de POO ? (3 reponses) (2 points)
4 | Fauconnier
5 | >Bonne reponse
6 | -----
7 | >2) Combien y a-t-il de lunes autour de Jupiter ? (a 2 pres) (3 points)
8 | 60
9 | >Mauvaise reponse
10 | -----
11 | >3) Quelle est la densite du plomb ? (a 5% pres) (2 points)
12 | 10.4
13 | >Bonne reponse
14 | -----
15 | >FINI : Vous avez marque 4 points sur 7.
```

2. Soient $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$, les sommets d'un polygone convexe P , pris dans le sens trigonométrique (au contraire du sens des aiguilles d'une montre), alors la surface de P vaut :

$$\frac{1}{2} [(x_1 y_2 + x_2 y_3 + x_3 y_4 + \dots + x_n y_1) - (y_1 x_2 + y_2 x_3 + y_3 x_4 + \dots + y_n x_1)] .$$

3. On ne demande pas de vérifier si un polygone est convexe et on supposera que ses sommets sont donnés dans le sens trigonométrique.

On voit dans cet exemple que les questions posées sont de types différents. La première demande une chaîne de caractère, la deuxième demande une valeur entière avec une marge d'erreur entière et la troisième une valeur réelle avec une marge d'erreur en pourcentage. On peut même imaginer d'autres types de questions.

Ainsi, il est naturel ici de vouloir abstraire les caractéristiques communes des questions dans un type `Question`, dont le type `Quizz` contiendra une liste.

Exercice 2 On ne s'intéresse pour l'instant pas au calcul des points.

1. Au vu de l'exemple ci-dessus le déroulement du jeu pour chaque question est le suivant :
 - (a) on affiche la question ;
 - (b) puis on lit la réponse de l'utilisateur ;
 - (c) enfin on vérifie la réponse et on affiche le résultat.

Ainsi, de quelles méthodes aura-t-on besoin dans `Question` pour pouvoir écrire une méthode **void** `poser()` dans une classe `Quizz` sans avoir à se soucier de l'implémentation des questions ?

Écrivez la méthode `poser()` pour un déroulement dans un terminal comme dans l'exemple ci dessus.

2. Dans nos quizz, on aura potentiellement des questions de différents types, parmi :
 - des questions attendant un nombre entier comme réponse
 - des questions attendant un nombre réel
 - des questions attendant un ou des mots
 - des questions admettant plusieurs réponses correctes (de type entier, réel, ou chaîne)
 - parmi les questions attendant un nombre, on peut décider d'accepter un intervalle de tolérance (ex : « Quel est le nombre d'habitants en France à 1 million près ? »)
 - des questions à choix multiple (choix présentés dans la question, précédés d'un numéro, à fournir dans la réponse)

Proposez une hiérarchie de classes (ou interfaces) recouvrant tous ces cas, permettant de factoriser le plus de code. Donnez les attributs, les méthodes abstraites et les méthodes implémentées de chaque classe (sans écrire les implémentations, ce sera fait en TP).

3. Que peut-on faire pour tester `Quizz` ?