

TD n° 6

Un peu de classe abstraite

1 Interface vs Classe abstraite

Exercice 1

1. Peut-on instancier une interface ? une classe abstraite ?
2. Peut-on mettre un constructeur ? un constructeur avec un corps ?
3. Peut-on écrire le code suivant : `A a = new B();` ;
 - si `A` est une classe abstraite, dérivée par la classe `B` ?
 - si `A` est une interface, implémentée par une classe `B` ?
4. Une interface/classe abstraite peut-elle contenir des méthodes abstraites ? non-abstraites ? statiques et abstraites ?
5. Une interface/classe abstraite peut-elle contenir des attributs ? avec quels modificateurs ? doivent-ils être instanciés ?
6. Une interface peut-elle hériter d'une autre interface ? d'une classe abstraite ?
7. Une classe abstraite peut-elle hériter d'une autre classe abstraite ? d'une interface ?

2 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 2

1. Écrire par exemple les classes `Cercle`, `Triangle`, `Rectangle`, `Carre`.
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 type de `Polygone` ? Quelle sera sa relation avec les classes précédentes ? (On ne demande pas de vérifier si la ligne se coupe.)
On rappelle que si (x_1, y_1) et (x_2, y_2) sont deux points du plan, alors la distance les séparant est donnée par

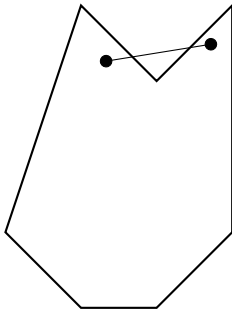
$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} .$$

3. 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.

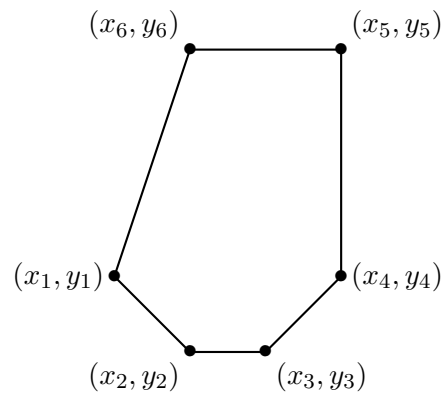
Si les sommets d'un polygone convexe, pris dans le sens trigonométrique (au contraire du sens des aiguilles du montre), sont $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ sa surface vaut

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

Quel sera le type de `PolygoneConvexe` ? Quelle est sa relation avec les types précédents ? (On ne demande pas de vérifier si un polygone est convexe et on supposera que ses sommets sont donnés dans le sens trigonométrique.)



(a) Un polygone non-convexe



(b) Un polygone convexe

3 Quizz

Le but de cette partie est de faire la structure des classes pour un programme permettant de poser des quizz notés. Chaque questionnaire sera la donnée d'une liste de questions, chacune valant un certain nombre de points.

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

```

1 | Bonjour et bienvenu.
2 |
3 | >1) Quel est le nom d'un des profs de POO ? (3 rep. possibles) (2 points)
4 | Yunes
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 plombs? (a 5% pres) (2 points)
12 | 10.4
13 | >Bonne reponse
14 | >-----
15 | >FINI : Vous avez marque 4 points sur 7.
```

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. Il est donc naturel ici de penser une structure de classes avec de l'abstraction. On vous propose de définir un type `Question` et un type `Quizz`.

Exercice 3 On ne s'intéresse pour l'instant pas au calcul des points.

1. Au vu de l'exemple ci-dessus et du déroulement du jeu (pour chaque question : on pose la question, on récupère une réponse, on vérifie la validité), 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. Que peut-on faire pour tester `Quizz` ?
3. Une question peut avoir une réponse textuelle entière (avec une éventuelle marge d'erreur entière), ou réelle (avec une éventuelle marge d'erreur en pourcentage). Comment faire cela.
4. Comment implémenter une question textuelle avec plusieurs bonnes réponses.
5. Comment implémenter une question à choix multiple.

Exercice 4 On va maintenant s'occuper du comptage des points.

1. Proposer une manière d'ajouter la gestion des points.
2. Une même question posée à des élèves de L1 ou de L2 ne vaudra pas le même nombre de points. Comment peut-on faire cela sans créer plusieurs fois la même question. (Dans le cas des QCM, il est de plus usuel d'attribuer des points négatifs en cas de mauvaise réponse).

Exercice 5 Comment faire pour permettre des questions à laquelle l'utilisateur doit donner plusieurs réponses comme l'illustrent les exemples suivants.

- *Donner le nom de 5 planètes du système solaire.*
- *Donner 10 nombres premiers.*