

Exercices sur les interfaces

Exercice 5.1.1 *Animaux*

Certain animaux peuvent crier, d'autres sont muets. On représentera le fait de crier au moyen d'une méthode affichant à l'écran le cri de l'animal.

- écrire une interface contenant la méthode permettant de crier.
- écrire les classes des chats, des chiens et des lapins (qui sont muets)
- écrire un programme avec un tableau pour les animaux qui savent crier, le remplir avec des chiens et des chats, puis faire crier tous ces animaux. Décrire ce qui s'affiche à l'écran à l'exécution de ce programme.

Exercice 5.1.2 *conversion*

On propose de pouvoir comparer des objets de différentes classes au moyen d'une conversion vers les nombres entiers. Pour cela on va utiliser une interface avec la méthode de conversion.

```
interface Convertible{  
    int toInt();  
}
```

Question 1

Modifiez les classes `Compte` et `Date` vues en cours pour qu'elles implémentent cette interface.

Question 2

Ecrivez une classe proposant des méthodes statiques pour comparer deux objets convertibles : une pour le test plus grand strict, une pour le test plus petit strict, une pour le test d'égalité, en comparant les entiers obtenus par conversion.

Question 3

Ajouter à la classe précédente une méthode statique permettant de trier en ordre croissant un tableau d'objets convertibles.

Exercice 5.1.3 déplacements

On reprend l'exemple du cours à propos de formes géométriques disposant d'une opération de translation.

```
interface AvecTranslation{
    void translation(double deplHor, double deplVer);
}
class Point implements AvecTranslation{
    double x, y;
    Point (double xi, double yi){
        x = xi;
        y = yi;
    }
    static double distance(Point p1, Point p2){
        return Math.sqrt((p1.x-p2.x)*(p1.x-p2.x) + (p1.y-p2.y) *(p1.y-p2.y));
    }
    public void translation(double deplHor, double deplVer){
        x = x + deplHor;
        y = y + deplHor;
    }
}
interface AvecSurface{
    double surface(); // Pas de corps de méthode
}
// Cercle implémente les deux interfaces
class Cercle implements AvecSurface, AvecTranslation{
    Point centre;
    double rayon;
    Cercle(Point ctr, double r){
        centre = ctr;
        rayon = r;
    }
    public double surface(){
        return Math.PI *rayon *rayon;
    }
    public void translation(double deplHor, double deplVer){
        centre.translation(deplHor,deplVer);
    }
}
class Rectangle implements AvecSurface, AvecTranslation{
    Point basGauche;
    double dimHor, dimVer;
    Rectangle(Point bg, double dh, double dv){
        basGauche = bg;
        dimHor = dh;
        dimVer = dv;
    }
    public double surface(){
        return dimHor *dimVer;
    }
    public void translation(double deplHor, double deplVer){
        basGauche.translation(deplHor,deplVer);
    }
}
```

```

    }
}
class Triangle implements AvecSurface, AvecTranslation{
    Point p1, p2, p3;
    Triangle(Point p1i, Point p2i, Point p3i){
        p1 = p1i;
        p2 = p2i;
        p3 = p3i;
    }
    public double surface(){
        double a = Point.distance(p1,p2);
        double b = Point.distance(p1,p3);
        double c = Point.distance(p2,p3);
        double demiper = (a+b+c)/2;
        return Math.sqrt(demiper*(demiper-a)*(demiper-b)*(demiper-c));
    }
    public void translation(double deplHor, double deplVer){
        p1.translation(deplHor,deplVer);
        p2.translation(deplHor,deplVer);
        p3.translation(deplHor,deplVer);
    }
}

```

On veut à présent pouvoir mettre en mouvement des pièces, c'est à dire effectuer une translation donnée à chaque top d'horloge, ce top étant matérialisé par une méthode. A l'appel de cette méthode, la translation est effectuée.

On va écrire une classe contenant dans une variable un objet à déplacer au top, cet objet étant n'importe quel objet proposant l'interface AvecTranslation. Les données du déplacement seront stockées dans autant de variables que nécessaire.

Question 1

Ecrire cette classe.

Question 2

Ecrire une méthode `main` avec un tableau contenant quatre objets de la classe écrite à la question précédente, chacun d'entre eux contenant une instance de figure ou de point différente.

Question 3

Ajoute à l'interface AvecTranslation une méthode permettant d'afficher la position d'un objet. Utiliser cette méthode pour afficher la situation à trois top horloge successifs.