

Java DUT 1 Feuille TP5  
Université Paris-Est Marne-la-Vallée

**Exercice 1.**—

- a) Écrire une classe `City` pour représenter des villes. Une `City` aura un champ `name` pour indiquer son nom, un champ `populationNumber` pour le nombre d'habitants et un champ `country` pour indiquer le pays. Les champs `name` et `country` seront non mutables.
- b) Écrire une classe `CityTest` avec une méthode `main`. Créez deux villes et affichez les.
- c) Écrire une classe `PopulationNumberException` dérivant de `Exception`.
- d) Effectuez une gestion d'exceptions avec la levée d'une exception lorsque l'on tente de créer une ville qui a population négative. L'exception sera capturée dans un bloc `try/catch` dans `CityTest`.
- e) On désire maintenant afficher à l'utilisateur le nombre d'habitants erroné négatif lorsque l'exception est capturée. Comme faire ?
- f) Créer une autre classe d'exceptions qui sera levée lorsqu'une ville dont le noms a moins de 3 caractères est créée. On affichera au moment de la capture un message de l'exception : "The city name has less than 3 characters".
- g) Etes vous maintenant bien convaincus qu'il vaut mieux utiliser le moins possible d'exceptions qui ne sont pas des `RuntimeException` ?

**Exercice 2.**—

On souhaite manipuler des formes géométriques placées dans le plan. Les formes que l'on considèrera seront par exemple des rectangles ou des ellipses. Chacune des formes aura une largeur et une hauteur de type `double` et la position dans le plan sera donnée par les coordonnées `x` et `y` de type `double` du centre de la forme géométrique.

Pour manipuler ces formes, on propose les méthodes suivantes dans une interface `Shape` : `move` pour déplacer la forme dans le plan, `perimeter` pour calculer le périmètre de la figure, et `area` pour calculer la surface de la figure.

```
interface Shape {  
    void move(double dx, double dy);  
    double perimeter();  
    double area();  
}
```

- a) Écrire une classe abstraite commençant l'implémentation de l'interface `Shape`. Les champs `height` et `width` seront non mutables.

```
public abstract class AbstractShape implements Shape {  
    ...  
}
```

- b) Écrire les classes `Rectangle` et `Ellipse`. Pour le périmètre de l'ellipse, on prendra une valeur approchée :  $\pi/4 * \text{height} * \text{width}$ .
- c) Ajouter une levée d'exception de la classe `RuntimeException` lorsque l'on tenter de créer une figure qui a une largeur ou une hauteur négative.
- d) Ajouter une méthode `toString` qui permettra d'afficher les 4 champs d'une forme géométrique.
- e) On désire modifier l'affichage précédent pour obtenir devant les 4 champs de la forme géométrique "Rectangle" si c'est un rectangle et "Ellipse" si c'est une ellipse. Que doit-on modifier ou ajouter ?
- f) Écrire une classe `ShapeTest` avec une méthode `main`. Créez un rectangle et une ellipse et affichez les.
- g) Que se passe-t-il si on écrit `Shape r = new AbstractShape(1,1,2,4);` ? Pourquoi ?