

Exercice 1

Définir une classe `Ville` modélisant une ville :

- Les deux propriétés d'une ville sont son nom et le nombre d'habitants . Ces deux propriétés sont protégées et sont initialisées à la création d'un objet : `new Ville("Nanterre",95782)`.
- L'interface d'un objet `Ville` fournit des accesseurs en lecture et en écriture vers chacun de ses attributs d'instance.
- L'interface fournit aussi une méthode `description()` qui retourne un `String` renseignant sur l'état de l'objet : `Nanterre : 95782 habitants`.

1. Définir la classe
2. Proposer une classe de démonstration de votre classe.

Exercice 2

Définir une classe `Aire` modélisant une aire urbaine (regroupement de villes) :

- Une aire urbaine ne peut contenir qu'un **nombre limité** de villes et cette limite est une propriété **publique** d'une aire urbaine (vous fixerez arbitrairement la valeur de cette limite). Si vous définissez d'autres propriétés, elles devront être protégées et vous ne devrez définir aucun accesseur en lecture vers ces propriétés. L'instruction `new Aire()` crée une nouvelle aire urbaine.
- L'interface de la classe fournit deux méthodes :

- Une méthode pour ajouter une ville à une aire urbaine:

```
Ville ville = new Ville("Nanterre",95782)
Aire aire = new Aire();
aire.ajouter(ville);
ville.setNom("Courbevoie")
ville.setPopulation(82074)
aire.ajouter(ville);
```

- Une méthode pour afficher les villes d'une aire urbaine en marquant d'une étoile la ville la plus peuplée (en cas d'ex-aequo, on marque toutes les villes ayant la population la plus élevée): `aire.afficher()`

2 villes

```
Nanterre : 95782 habitants *
Courbevoie : 82074 habitants
```

Note. Si l'aire urbaine est vide, `aire.afficher()` affiche : 0 villes. Une ville ne doit pas pouvoir être ajouté deux fois. S'il n'est plus possible d'ajouter une ville à une aire urbaine, un court message l'indiquant est affiché dans la console.

1. Définir la classe
2. Proposer une classe de démonstration de votre classe.