

Java DUT 1 Feuille TD10
Université Paris-Est Marne-la-Vallée

On rappelle que l'on doit mettre une seule classe ou interface par fichier.

Exercice 1.—

Reprendre les classes `Company` et `Employee` du TD8.

- a) Écrire une méthode qui permet de sauvegarder une `Company` dans un fichier de sérialisation dont le nom est passé en argument. Dans quelles classes ou interfaces doit-on mettre cette méthode ?

```
public void saveCompany (String name) throws Exception {  
}
```

- b) Écrire une méthode qui permet de lire une `Company` dans un fichier de sérialisation dont le nom est passé en argument.

```
public Company restoreCompany (String name) throws Exception {  
}
```

- b) Écrire une méthode `main` dans une classe `Test1` où on crée une `Company` avec un ou deux `Employee`. On sauvegarde la `Company` dans un fichier de sérialisation dont le nom sera passé en argument de la ligne de commande. Écrire une méthode `main` dans une classe `Test2` où on lit un objet `Company` dans un fichier de sérialisation dont le nom est passé en argument de la ligne de commande, et où on affiche ensuite cette `Company`.

Exercice 2.—

On considère une classe `StringList` qui contient une liste de mots de type `ArrayList<String>`. Une telle liste peut par exemple contenir les mots suivants :

```
titi toto tutu toto tutu tete tut tot tutu
```

- a) Écrire dans la classe `StringList` une méthode qui renvoie `true` s'il existe un mot présent dans la liste avec plusieurs occurrences. Pensez à utiliser un `Set`.

```
public boolean checkDouble(){  
}
```

- b) Écrire dans la classe `StringList` une méthode qui renvoie un ensemble de type `HashSet<String>` qui comprend chaque élément de la liste en un seul exemplaire. Les questions a) et b) sont très similaires.

```
public Set<String> getSingle(){  
}
```

c) Tester vos deux méthodes (avec une méthode `main` dans une autre classe).

Exercice 3.—

On reprend les classes du cours pour manipuler des formes géométriques 2D

- l'interface `Shape`;
- la classe abstraite `AbstractShape`;
- les classes `Rectangle` et `Ellipse`.

On désire pouvoir effectuer des translations de ces formes géométriques. Ajouter une méthode `translate` au (ou aux) bons endroits.