

TP n°1 : Programmation Orientée Objet, UML

Compilation et exécution en Java

Écrivez chaque classe dans un fichier séparé, et nommez ces fichiers de la même manière que votre classe. Exemple : vous écrirez la classe `Position` dans le fichier `Position.java`. On rappelle que le nom d'une classe commence par une majuscule (ceci est une convention, donc ce n'est pas obligatoire mais vivement recommandé), et il en sera de même pour son fichier.

Pour compiler votre fichier `Position.java`, tapez dans un terminal :

```
$> javac Position.java
```

Si la classe `Position` contient une méthode `main`, vous pouvez l'exécuter par :

```
$> java Position
```

Vous cherchez une information sur une classe de la bibliothèque standard de Java ? La réponse est ici : <http://docs.oracle.com/javase/7/docs/api/index.html>

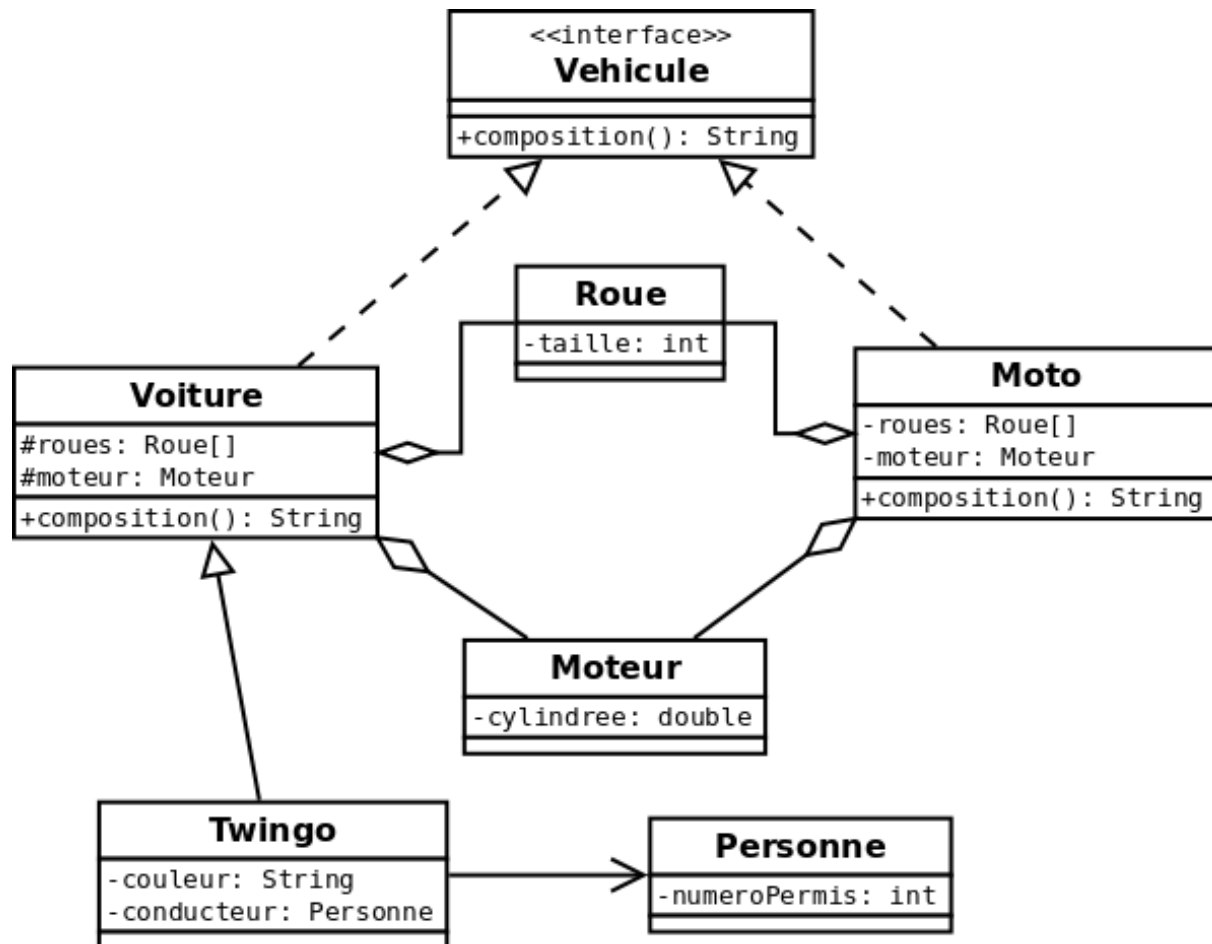
Exercice 1 (Antennes-relais)

Une antenne-relais est définie par un identifiant entier, une position et une plage de fréquences en MHz. La plage des hautes fréquences se situe en général entre 300MHz et 3GHz.

1. Définir une classe `Position` avec un constructeur et des assesseurs.
2. Définir une classe `Antenne` avec un constructeur et des assesseurs.
3. Implémenter un opérateur d'écriture (méthode `toString()`) pour écrire une antenne sous la forme suivante :
`76:[20-1000]@(3.5, 5.0) //id:plage@position`
4. Écrire une méthode qui calcule la distance entre deux antennes.
5. Écrire une méthode qui retourne la largeur de la plage de fréquences.
6. Écrire un code client qui déclare un ensemble d'antennes trié dans l'ordre croissant des largeurs des plages de fréquences. Utiliser la collection `SortedSet` de la bibliothèque standard en la paramétrant par le type des éléments `Antenne`.
7. Supposons qu'une classe `Intervalle` représentant des intervalles d'entiers soit équipée d'un calcul de largeur. Réutilisez cette classe pour représenter des antennes.

Exercice 2 (De l'UML à Java)

Codez les classe du diagramme UML ci-dessous (oui, c'est l'exercice 2 du TD 1).



Exercice 3 (De Java à l'UML)

Utiliser le logiciel *dia* (ou un autre si vous en connaissez un autre) pour faire le diagramme de classe UML décrivant les classes Java ci-dessous.

```
interface Animal
{
    public void cri();
    public boolean estCarnivore();
}

class Chien implements Animal
{
    public String nom;
    protected int age;
    private String tatouage_;

    public void cri() { System.out.println("Aboie"); }

    public boolean estCarnivore()
    {
        return true;
    }
}

class Lapin implements Animal
{
    private double taille_;

    public void cri() { System.out.println("Clapit"); }

    public boolean estCarnivore()
    {
        return false;
    }

    void combienDeCarottes(int carottes)
    {
        System.out.println("Mange_ " + carottes + "carotte(s)");
    }
}

class Levrier extends Chien
{
    protected Nourriture nourritureFavorite;

    public void cri() { System.out.println("Jappe"); }
}

class Nourriture
{
    public String nom;
    int valeurCalorique;
}
```