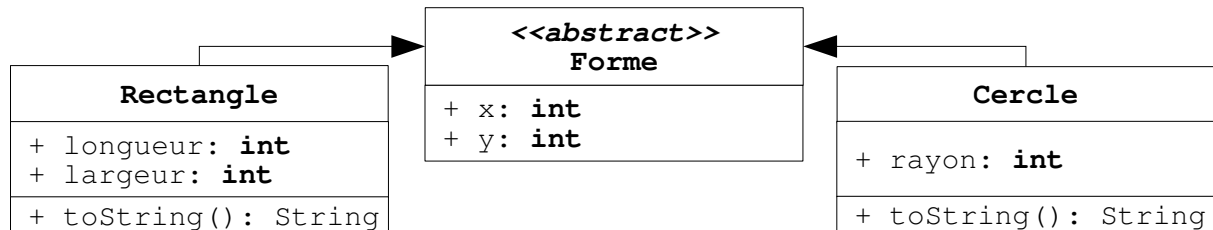


## Exercice 1

Écrire, en utilisant la documentation Java de la classe `java.io.File`, un programme qui prend en paramètre un chemin d'accès vers un fichier et qui teste tout d'abord si ce fichier existe, puis, si ce fichier est un répertoire, affiche la liste des fichiers de ce répertoire ou sinon affiche la taille de ce fichier ainsi que son chemin absolu.

## Exercice 2

On cherche à stocker et à lire des objets dans des fichiers. Ces objets sont des formes de deux types différents :



On propose 2 manières différentes de stocker ces objets : dans un fichier texte et dans un fichier binaire. On crée une classe pour chacun des types de stockage possibles, chacune implémentant l'interface suivante :

```

import java.io.IOException;
import java.util.Collection;

public interface IOForme {
    public Collection<Forme> lire(String fileName)
        throws IOException;

    public void sauver(Collection<Forme> formes, String fileName)
        throws IOException;
}

```

La méthode `main()` crée une collection de formes et l'écrit dans un fichier dans un premier temps, puis en crée une nouvelle en lisant le contenu de ce même fichier dans un deuxième temps. Ceci permet de tester à la fois les méthodes de lecture et d'écriture.

```

public static void main(String[] args) throws IOException {
    String fileName = "nomFichier";
    IOForme io = new XXXIOForme(); // PlainIOForme, ObjectIOForme...
    Collection<Forme> formes1 = new ArrayList<Forme>();
    formes1.add(...);
    ...
    io.sauver(formes1, fileName);
    Collection<Forme> formes2 = io.lire(fileName);
    for (Forme f : formes2) { // on vérifie que formes1 = formes 2
        System.out.println(f);
    }
}

```

## Stockage dans un fichier texte

Créer une classe `PlainIOForme` implémentant l'interface `IOForme` qui stocke une collection de formes dans des fichiers texte écrits sous la forme suivante :

```

rectangle 3 2 5 6
cercle 3 0 5

```

Dans la pratique, on utilise des formats de données plus complexes car des API existent déjà pour lire et écrire facilement des données. Parmi ces formats, on retrouve le XML, le JSON...

## Stockage dans un fichier binaire

Créer une classe `ObjectIOForme` implémentant l'interface `IOForme` qui stocke une collection de formes dans des fichiers binaires.