

## TD2 – Permissions en Java

### Exercice 1:

1. Créer un `SecurityManager` qui permet à tout code qui l'interroge d'accéder au fichier `fich1.txt` du répertoire courant en lecture seulement.
  2. Modifier cette permission de sorte que tout fichier avec l'extension `txt` dans le répertoire `/home/dima` et ses sous-répertoires puisse être accessible en lecture.
  3. Écrire un programme qui installe le `SecurityManager` qu'on vient de créer et qui lit un nom de fichier à l'entrée standard et essaye d'ouvrir le fichier en lecture. Si le `SecurityManager` ne lui permet pas d'accéder au fichier, afficher un message d'erreur et demander un autre nom de fichier.
- 

### Exercice 2:

1. Résoudre les deux premières questions de l'exo précédent en écrivant des entrées `grant` dans le fichier `java.policy`.
  2. Modifier cette permission de sorte que tout fichier avec l'extension `txt` dans le répertoire `/home/dima` et ses sous-répertoires puisse être accessible en lecture.
  3. Modifier la solution de sorte que les permissions de lecture soient accordées seulement au code provenant du répertoire `/home/dima/java`.
- 

**Exercice 3:** On dispose d'une classe `A` qui possède une méthode `met(String, String)` qui s'occupe de l'insertion d'une chaîne de caractères en ordre alphabétique dans un fichier qu'on lui donne en 2e paramètre.

- Écrire une autre classe ayant une méthode `ouverture` qui utilisera `PrivilegedAction` pour accéder à cette méthode.
  - Écrire aussi des règles `java.policy` pour permettre à tout code provenant de l'adresse `www.confiance.fr` d'utiliser la méthode `met` sur les fichiers qui ont des extensions `tmp`.
  - A-t-on besoin d'ajouter des règles pour la classe `A` aussi ?
- 

**Exercice 4:** Considérons le code suivant :

```
import java.net.*;
import java.io.*;
public class URLTester {
    public static void main (String args[]) throws Exception {
        ClassLoader loader = new FileClassLoader(args[0]);
        Class c = loader.loadClass (args[1]);
        Object tester = c.newInstance();
        Method[] m = c.getDeclaredMethods();
        for (int i = 0; i < m.length; i++)
            System.out.println(m[i].toString()); }
}
```

1. Décrivez le comportement du programme.
  2. Considérons les permissions suivantes : `java.lang.RuntimePermission("createClassLoader")`, `java.lang.RuntimePermission("accessClassInPackage.home.dima.java")`, `java.lang.RuntimePermission("accessDeclaredMembers")`. Quelle permission sera vérifiée et quand ?
-