UF1.NF1.Seguretat

1.- Crea un mètode que llegeixi un fitxer .txt i mostri el contingut per pantalla. Executa l'aplicació amb i sense el gestor de seguretat des del cmd. Quina és la diferència?

```java
public static void main(String[] args) throws Exception {
    String rutaArchiu = "C:\\Users\\jrour\\Documents\\Institut Vidreres\\2nDAM\\M09\\UF1\\UF1_NF1_Seguretat\\src\\Ho

    File fitxer = new File(rutaArchiu);

    FileReader fr = new FileReader(fitxer);
    StringWriter sw = new StringWriter();

    while (fr.ready()) {
        sw.write(fr.read());
    }

    System.out.println("Contingut del fitxer: \n" + sw.toString());

    fr.close();
    sw.close();
}
```

Sense el gestor de seguretat:

```
C:\Users\jrour\Documents\Institut Vidreres\2nDAM\M09\UF1\UF1_NF1_Seguretat>java src/App.java
Contingut del fitxer:
Hola bon dia
```

Amb el gestor de seguretat:

```
C:\Users\jrour\Documents\Institut Vidreres\2nDAM\M09\UF1\UF1_NF1_Seguretat>java -Djava.security.manager src/App.java
WARNING: A command line option has enabled the Security Manager
WARNING: The Security Manager is deprecated and will be removed in a future release
Exception in thread "main" java.security.AccessControlException: access denied ("java.lang.RuntimePermission" "access
ClassInPackage.jdk.internal.misc")
        at java.base/java.security.AccessControlContext.checkPermission(AccessControlContext.java:485)
        at java.base/java.security.AccessController.checkPermission(AccessController.java:1068)
        at java.base/java.lang.SecurityManager.checkPermission(SecurityManager.java:416)
        at java.base/java.lang.SecurityManager.checkPackageAccess(SecurityManager.java:1332)
        at java.base/jdk.internal.loader.ClassLoaders$AppClassLoader.loadClass(ClassLoaders.java:184)
        at java.base/java.lang.ClassLoader.loadClass(ClassLoader.java:520)
        at jdk.compiler/com.sun.tools.javac.launcher.Main.main(Main.java:132)

C:\Users\jrour\Documents\Institut Vidreres\2nDAM\M09\UF1\UF1_NF1_Seguretat>
```

2.- Modifica els paràmetres d'execució al IDE perquè s'executi l'exercici 1 amb el gestor de seguretat per defecte.

3.- Crea un fitxer de polítiques amb PolicyTool que permeti llegir i executar tots els fitxers de la carpeta de l'exercici 1.

4.- Executa l'exercici 1 des del cmd amb el fitxer de polítiques creat. Què passa?

5.- Crea i guarda en un magatzem de claus una clau privada per poder executar un sòcol servidor. Fes-ho des del cmd.

```
C:\Windows\System32>keytool  -genkey -alias "srvAlias" -keyalg RSA -keystore ServerKeyStore.jks -keysize 2048
Introduzca la contraseña del almacén de claves:
Volver a escribir la contraseña nueva:
```

```
¿Cuáles son su nombre y su apellido?
  [Joan Roura Lema ]:
¿Cuál es el nombre de su unidad de organización?
  [Seguretat]:
¿Cuál es el nombre de su organización?
  [Institut Vidreres]:
¿Cuál es el nombre de su ciudad o localidad?
  [Vidreres]:
¿Cuál es el nombre de su estado o provincia?
  [Catalunya]:
¿Cuál es el código de país de dos letras de la unidad?
  [ES]:
¿Es correcto CN="Joan Roura Lema ", OU=Seguretat, O=Institut Vidreres, L=Vidreres, ST=Catalunya, C=ES?
  [no]:  si

Generando par de claves RSA de 2.048 bits para certificado autofirmado (SHA256withRSA) con una validez de 90 días
        para: CN="Joan Roura Lema ", OU=Seguretat, O=Institut Vidreres, L=Vidreres, ST=Catalunya, C=ES

C:\Windows\System32>
```

6.- Crea la classe SocketSSLServidor que creï un socket servidor a partir de la clau privada creada a l'exercici anterior.

```
class SSLServerSocketFactory {
    Run | Debug
    public static void main(String[] args) {
        System.setProperty(key:"javax.net.ssl.keyStore", value:"C:\\Users\\jrour\\Desktop\\ServerKeyStore.jks");
        System.setProperty(key:"java.net.ssl.keyStorePassword", value:"20031003jMj");
```

7.- Crea un mètode rebreDades que esperi la connexió d'un socket client i mostri per pantalla tot el que li envia.

```java
public static void rebreDades(SSLSocket socket, int client) throws IOException {
    String dades = "";

    do {
        Scanner llegir = new Scanner(socket.getInputStream());

        dades = llegir.nextLine();

        System.out.println(client + " " +  dades);
        System.out.flush();

    } while (dades.equals(anObject:""));
}
```

8.- Crea la classe SocketSSLClient amb un mètode enviarDades que es connecti a l'adreça i port on està esperant la connexió el socket servidor i li envïi missatges per a ser llegits. Fes-ho sense importar el certificat de l'emissor al magatzem de confiança del client. Què passa?

9.- Importa al magatzem de confiança del client la clau privada del servidor.

10.- Modifica el constructor del SocketSSLClient perquè llegeixi el magatzem de confiança i creï un socket per connectar-se a una adreça i un port on hi hagi un socket servidor en espera. Simula ara l'enviament de missatges entre el SocketSSLClient i el SocketSSLServidor.

11.- Realitza l'exercici anterior amb un company utilitzant SSLSockets.

12.- Cerca el significat de conjunt de xifrat (cipher suite) i crea un mètode que mostri per pantalla quins utilitzen les classes dels SSLSockets.

13.- Fes l'enviament entre un sòcol client i un sòcol servidor del teu nom i cognoms

utilitzant buffers i canals.

14.Crea un mètode que, a partir d'un usuari i contrasenya retorni un parell de claus asimètriques a partir d'una clau privada.