```
public interface Videotheque extends Remote{
	public void enregistrer(Film f) throws RemoteException;
	public Film rechercher(String titre) throws RemoteException;
	}

public class VideothequeImpl extends UnicastRemoteObject implements
Videotheque{
	private static final long serialVersionUID = 1L;
	private HashMap<String, Film> films;
	public VideothequeImpl() throws RemoteException {
		films = new HashMap<>();
	}

	public void enregistrer(Film f) throws RemoteException{
		 films.put(f.getTitre(), f);
	}
	public Film rechercher(String titre) throws RemoteException{
		return films.get(titre);
	}
	public void main(String args[]) {
		try{
			Registry reg=
LocateRegistry.createRegistry(8080);
			Videotheque vd = new VideothequeImpl();
			Naming.rebind("//localhost/reg", vd);
		}catch(Exception ex) {
			ex.printStackTrace();
		}
	}
}

public interface Film extends Serializable{
	public String getTitre();
	public void setTitre(String titre);
	public String getHost();
	public void setHost(String host);
	public int getPort();
	public void setPort(int port);

}
public class FilmImpl implements Film{
	private static final long serialVersionUID = 1L; //numero
long unique
	String titre;
	String host;
	int port;
	public FilmImpl(String titre, String host, int port) {
		this.titre = titre;
		this.host= host;
		this.port= port;
	}

	public String getTitre(){
		return this.titre;
	}
	public void setTitre(String titre) {
		this.titre = titre;
	}


	public String getHost() {
		return this.host;
	}
	public void setHost(String host) {
		this.host = host;
	}


	public int getPort() {
		return this.port;
	}
	public void setPort(int port) {
		this.port=port;
	}
}

public class Client {
    public static void main(String[] args) {
        try {
            // Obtention de l'objet distant
            Videotheque v = (Videotheque)
Naming.lookup("Videotheque");

            // Enregistrement
            Film film = new FilmImpl("Le Seigneur des Anneaux",
"localhost", 8080);
            v.enregistrer(film);

            // Recherche
            Film filmRecherche = v.rechercher("Le Seigneur des
Anneaux");
            System.out.println("Film trouvé: " +
filmRecherche.getTitre() + ", " + filmRecherche.getHost() + ", " +
filmRecherche.getPort());
```

```
        } catch (Exception e) {
            System.err.println("Client exception: " +
e.toString());
            e.printStackTrace();
        }
    }
}
public class ServerStreaming{

    public static void main(String[] args) {
        String titre = args[0];
        String fichier = args[1];
        int port = Integer.parseInt(args[2]);

        try {
            // Enregistrement dans la vidéothèque

            Videotheque v = (Videotheque)
Naming.lookup("Videotheque");
            //InetAddress.getLocalHost(), qui retourne une instance
de InetAddress représentant l'adresse locale. Pour obtenir
l'adresse IP sous forme de String, vous devez ensuite appeler
getHostAddress() sur cette instance.
            v.enregistrer(new FilmImpl(titre,
InetAddress.getLocalHost().getHostAddress(), port));

            // Démarrage du serveur de streaming
            ServerSocket sserveur = new ServerSocket(port);
            System.out.println("StreamingServer est à l'écoute sur
le port " + port);

            while (true) {
                Slave slave = new Slave(sserveur.accept(),
fichier);
                slave.start();
            }

        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
public class Slave extends Thread{
       private Socket sclient;
       private String fichier;
       public Slave(Socket s1, String f1 ) {
              this.sclient = s1;
              this.fichier = f1;
       }
       public void run() {
              try {
              Socket socket = sclient; // Utilise la socket passée
au constructeur
        FileInputStream fis = new FileInputStream(fichier); //
Ouvre le fichier à streamer
         OutputStream os = socket.getOutputStream();// Obtient le
flux de sortie associé à la socket

       byte[] buffer = new byte[4096]; // Crée un buffer pour la
lecture du fichier
       int nblu = fis.read(buffer);
       while (nblu> 0) { // Lit le fichier jusqu'à la fin
                                os.write(buffer, 0, nblu);
                }
       fis.close();
       os.close();
       sclient.close();
       }catch (Exception e) {
        e.printStackTrace();
    }

public class ClientStreamin {

    public static void main(String[] args) {

        String titre = args[0];

        try {
            // Consultation de la vidéothèque

            Videotheque v = (Videotheque) Naming.lookup("Videotheque");
            Film film = v.rechercher(titre);

             return;


            // Connexion au StreamingServer
            Socket socket = new Socket(film.getHost(), film.getPort());
            InputStream is = socket.getInputStream();

            byte[] buffer = new byte[4096]; // Crée un buffer pour la
lecture du fichier
            int nblu = is.read(buffer);
            while (nblu>0) {
                VLC.show(buffer);//VLC est une méthode qui diffuse en
continue.
            }cath...
```

```
}
```