

Rapport

ARAR TFTP - Client TFTP

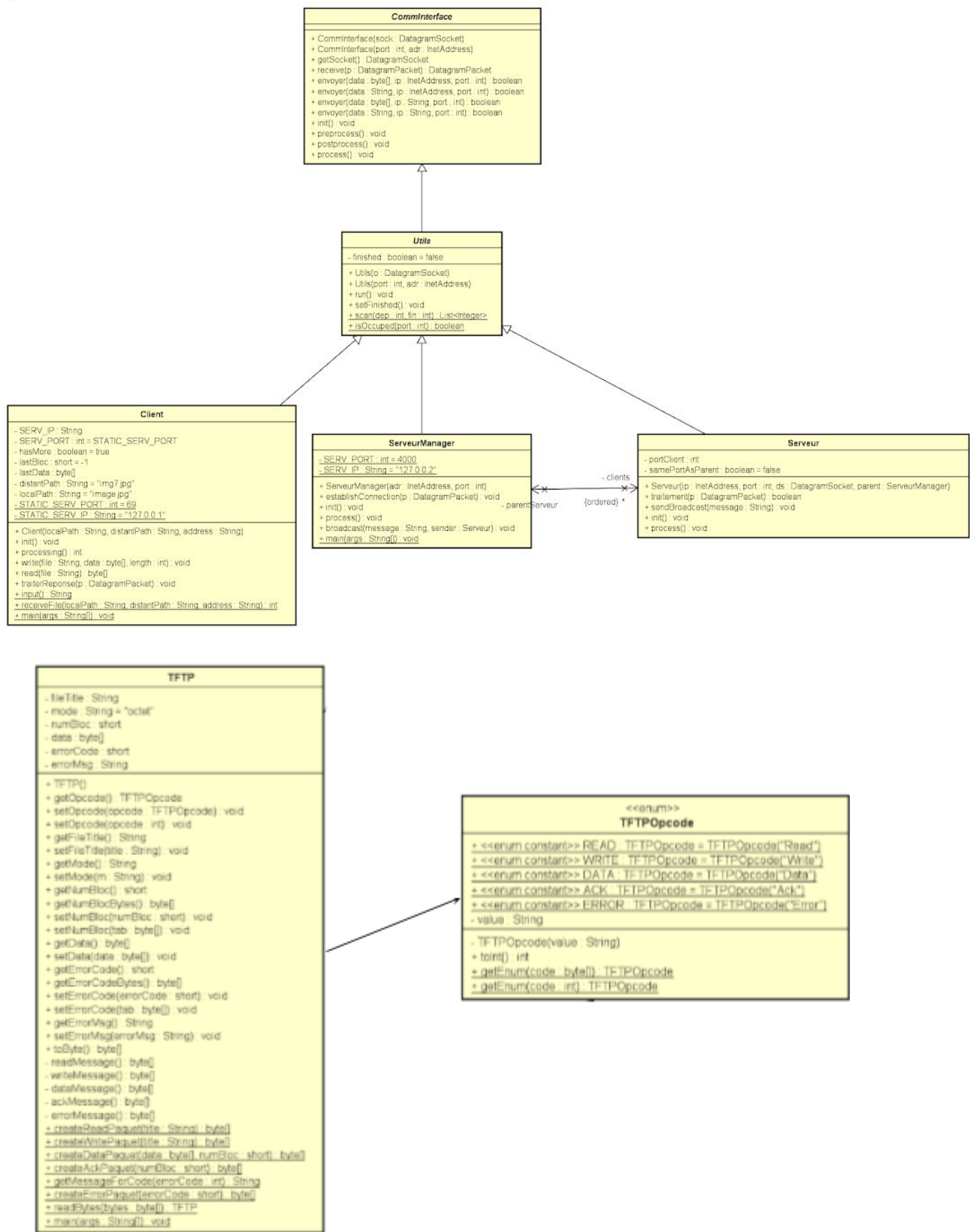
Fonctionnalités implémentées

- Ouverture et fermeture d'une liaison UDP entre le client et le serveur
- Gestion des requêtes GET côté client (tous type de fichier) :

```
receiveFile(String localPath, String distantPath, String address)
```

- Connexions concurrentes sur le serveur
- Connexions fonctionnelles en réseau

Diagramme de classes



CommInterface et Utils

La classe CommInterface contient les méthodes nécessaires à l'envoi et la réception de paquets par le client et le serveur. Ces méthodes utilisent le DatagramSocket instancié par la classe. La classe Utils est héritée de CommInterface et lui ajoute des fonctionnalités, dont notamment la fonction run qui définit des étapes de traitement pour le client et le serveur qui hériteront de cette classe. Le client et le serveur n'auront plus qu'à redéfinir ces étapes, sans avoir à redéfinir run.

Client

La classe Client héritée de Utils, gère tout ce qui est inhérent au client STF : l'interface utilisateur, et la récupération et l'enregistrement du fichier distant en fonction du fichier, du chemin et de l'adresse spécifiés par l'utilisateur via l'interface.

TFTP et TFTPOpcode

La classe TFTP sert à lire et à créer des paquets respectant le protocole TFTP. Les différents types de paquets possibles sont Read (opcode 1), Write (opcode 2), Data (opcode 3), Ack (opcode 4) et Error (opcode 5) : ces opcodes sont définis par la classe d'énumération TFTPOpcode.

Error

Classe héritée d'Exception servant à gérer les exceptions relatives aux erreurs TFTP. Les paquets de type Error envoyés avec TFTP lèvent une Exception de ce type.

Une instance de cette classe contiendra un code d'erreur (correspondant à CR_EM pour un paquet obtenu avec ReceiveFile ou CR_RV pour SendFile) ainsi que le message d'erreur associé. Ces codes d'erreurs (repris pour la plupart des erreurs gérées nativement par TFTP) peuvent être les suivants :

- -2 : Problème d'accès au fichier côté client
- -1 : Erreur inconnue côté client
- 1 : Fichier non trouvé
- 2 : Accès non autorisé
- 3 : Disque plein ou erreur d'allocation d'espace disque
- 4 : Violation du protocole TFTP
- 5 : Transfer ID inconnu
- 6 : Le fichier existe déjà
- 7 : Utilisateur inconnu



Les sources des différentes classes sont disponibles dans le dossier "src" contenu dans l'archive.

Scénario

Il s'agit du scénario demandé décrivant une requête GET avec la fonction que nous devons implémenter :

```
receiveFile(String localPath, String distantPath, String address)
```

