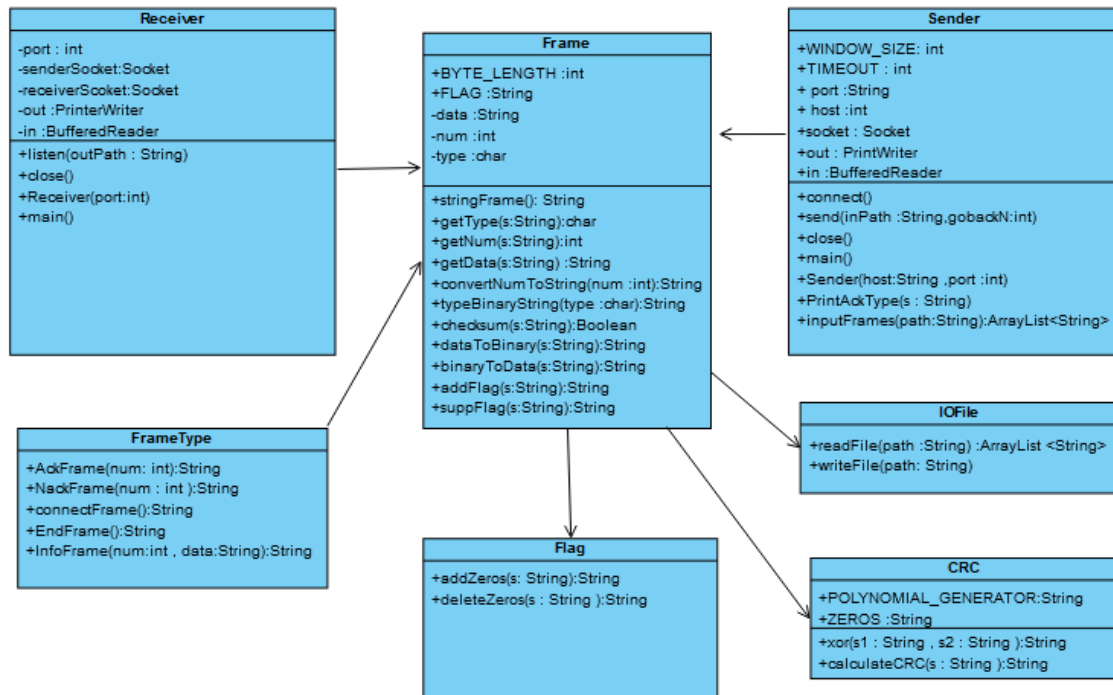


IFT 3325 Téléinformatique :

Partie Rapport :

Diagramme de classe :



Remarque : j'ai pas représenté la classe Test dans le diagramme parce qu'elle contient que des exemples simples .

Ce rapport présente une implémentation d'une version simplifiée du protocole HDLC utilisant plusieurs classes Java pour faciliter l'échange de données entre un émetteur (Sender) et un récepteur (Receiver).

La classe Sender :

La classe Sender est responsable de l'envoi des trames au récepteur (receiver). Elle utilise une fenêtre de communication et des mécanismes d'acquittement pour assurer la transmission fiable des données.

Fonctions :

- **connect():** Établit une connexion avec le récepteur en envoyant un trame de connexion et attend la réponse.
- **send() :** Envoie les trames au récepteur à partir d'un fichier d'entrée spécifié.
- **ACKType():** Détermine le type d'acquittement (ACK ou NACK) en fonction du trame reçu.
- **close():** Ferme la connexion avec le récepteur (receiver).

- **inputFrames()** : lit les lignes d'un fichier, les convertit en "frames" en utilisant la classe FrameType, et stocke ces "frames" dans une liste

La classe Receiver :

La classe Receiver est responsable de recevoir les trames provenant de l'émetteur (Sender). Elle traite les trames reçus, détecte s'il y a des erreurs, et stocke les données reçues.

Fonctions :

- **listen()**: recevoir les trames , traite les différents types de trame et envoie des acquittements (ACK ou NACK)
- **close()**: Ferme la connexion avec l'émetteur (sender).

La classe Frame :

La classe Frame construit une trame avec des informations telles que le type, le numéro, et les données. Elle est utilisée par le Sender et le Receiver pour la création et le traitement des trames.

Fonctions :

- **stringFrame()** :Convertir la trame en une représentation binaire pour la transmission.
- **getType()** : Extraire le type d'une trame à partir de sa représentation binaire.
- **getNum()** : Extraire le numéro de trame d'une représentation binaire.
- **getData()** : Obtenir les données d'une représentation binaire de la trame.
- **getCRC()** : Obtenir le CRC d'une représentation binaire de la trame.
- **addFlag()** : Ajouter des indicateurs de début et de fin à la trame.
- **suppFlag()** : Retirer les indicateurs de début et de fin de la trame.
- **convertNumToString()** : Convertir le numéro de trame en binaire.
- **typeBinaryString()** : Convertir le type de trame en binaire.
- **Checksum()** : détecter si il y a des erreurs .

- **dataToBinary()** : Convertir les données en binaire.
- **binaryToData()** : Convertir une chaîne binaire en données.

La Classe FrameType

La classe FrameType fournit des méthodes pour créer différents types de trame : les trames d'acquittement (ACK ou NACK), les trames d'informations, les trames de connexion et les trames de fin de connexion .

Fonctions :

- **AckFrame()** :Elle Crée une trame d'acquittement (ACK) avec le numéro spécifié.
- **NackFrame()** : Elle Crée une trame de non-acquittement (NACK) avec le numéro spécifié.
- **InfoFrame()** : Elle Crée r une trame d'information avec le numéro et les données spécifiés.
- **EndFrame()** : Elle Crée une trame de fin indiquant la fin de la transmission.
- **connectFrame()** : Elle Crée une trame de demande de connexion.

La classe CRC

La classe CRC implémente un algorithme de calcul de (CRC) en prenant compte du polynôme générateur. Elle est utilisée pour détecter les erreurs de transmission.

Fonctions :

- **calculateCRC()**: Calculer le CRC d'une chaîne binaire donnée.
- **xor()** : calcule le ou exclusive entre deux chaine de caractère binaires.

La classe Flag

La classe Flag implémente des fonctions qui ajoute et supprime des zéros (drapeaux) dans une séquence de données binaires (stuffing et destuffing).

Fonctions :

- **addZeros ()**:Ajouter des zéros après chaque séquence de 5 bits '1' consécutifs dans les données binaires d'entrée.
- **deleteZeros ()** :Supprimer des zéros après chaque séquence de 5 bits '1' consécutifs dans les données binaires d'entrée (opération inverse du bit stuffing).