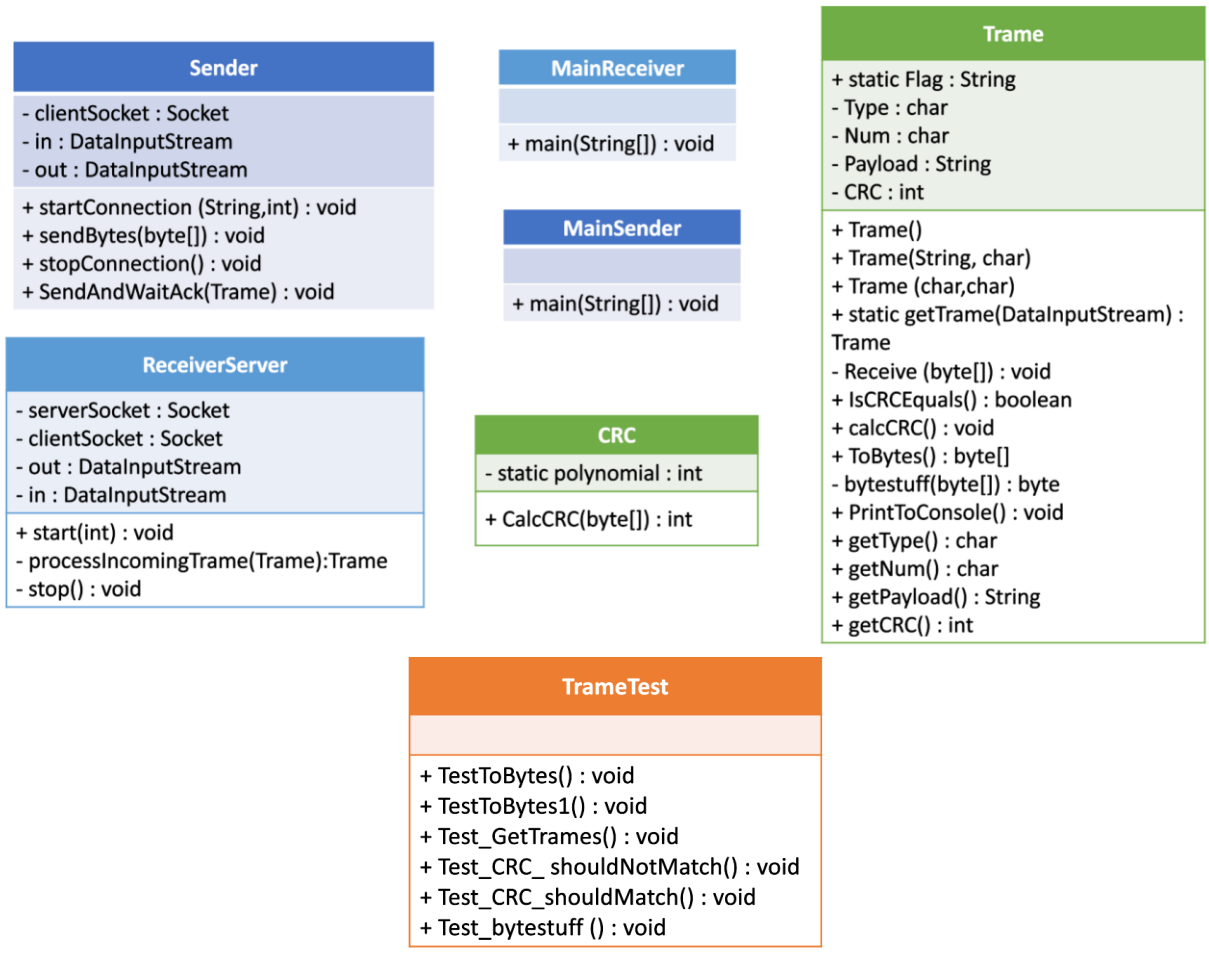
Tasha Bhojwani (20213638)
Nicolas Leblanc (20165600)

IFT 3325- Téléinformatique- Devoir 2

Rapport
29 novembre 2021

Diagramme de classe :



Description des classes et des méthodes utilisées :

Voici la description des classes de notre projet, rangées selon 4 répertoires : common, receiver, sender et mytest.

Quelques remarques et explications sur notre projet :

- Notre système se base sur un go-back-n avec une fenêtre de 1.
- Nous avons décidé de travailler en octets (et non en bits) car c'est plus efficace en pratique. En effet, si on décide d'utiliser des bits, on devrait les coder soit en int ce qui prendrait 4 octets pour chaque bit, soit en char, ce qui prendrait 1 octet pour chaque bit. Notre méthode permet de coder dans 1 octets 8 bits. De plus, l'énoncé précise que les trames sont "orientées caractères".
- Nous avons décidé de ne pas faire de bit stuffing mais character stuffing parce que nous travaillons en octets.
- L'émetteur envoie des trames de type I (données), C (demande de connexion) et F (fin de communication).
- Le récepteur envoie des trames de type A, et R (REJ).
- Nous n'avons pas codé de méthode pour les P-bit car cela nous a semblé inutile. En effet, l'émetteur renvoie directement la trame en cas de problème (lorsque le temporisateur expire avant de recevoir une confirmation, ou lorsqu'il reçoit une trame R).

Common

Class Trame

- **Attributs** : flag, type, num, payload (données), CRC
- **Méthodes** :
  - **Constructeurs** :
    - Avec data et num : on crée une trame d'information avec les données et le numéro de trame en paramètres
    - Avec rien : on crée juste l'objet trame
    - Avec le type et num : on crée une trame du type donné en paramètre et avec le numéro donné
  - **GetTrame (DataInputStream in)** : lit et renvoie la trame reçue
  - **Receive(byte[] b)** : méthode qui initialise une trame à partir d'une séquence de bytes reçue
  - **IsCRCEquals()** : compare le CRC calculé au CRC reçu. Renvoie le booléen de l'égalité.
  - **CalcCRC()** : calcule le CRC des données de la trame
  - **ToBytes()** : renvoie la trame transformée en bytes
  - **Bytestuff(byte[] b)** : rajoute un byte 'E' avant les caractères spéciaux (les flags, et lui-même)
  - **PrintToConsole()** : affiche la trame

- **getType(), getNum(), getPayload(), getCRC()** : getters, méthodes qui renvoient les attributs de la trame

Class CRC

- **Attributs** : polynomial (constante int, le polynôme générateur)
- **Méthodes**
  - **CalcCRC(byte[] byte)** : calcule et retourne le CRC pour les bytes passés en paramètre

Receiver

Class ReceiverServer

- **Attributs** : serverSocket, clientSocket, out (stream), in (stream)
- **Méthodes**
  - **Start(int port)** : méthode qui permet au Receveur de se connecter avec un émetteur sur le port port, de recevoir des trames, de les analyser, d'y répondre, et ensuite de fermer la connexion une fois toutes les données reçues.
  - **processIncomingTrame(Trame Incoming)** : rend la trame correspondante à la réponse du receveur selon la trame reçue. Rend une trame de confirmation si Incoming est une trame d'information ou une demande de connexion. Rend une trame null si la trame reçue est une trame de fin de communication.
  - **stop()** : méthode qui arrête toutes les procédures en cours

Class MainReceiver : main, programme principal du receveur, permet de lire le port de la commande d'exécution du receveur, crée une instance ReceiveServer, et lance la communication avec la méthode start de ReceiveServer.

Sender

Class SenderClient

- **Attributs** : clientSocket, in (stream), out(stream)
- **Méthodes** :
  - **startConnection(String ip, int port)** : constructeur
  - **sendBytes(byte[] b)** : écrit b dans le stream
  - **stopConnection()** : envoie une trame de fin de communication au récepteur, et arrête toutes les procédures en cours
  - **sendAndWaitAck(Trame trame)** : permet d'envoyer la trame en paramètre et d'attendre la confirmation de réception

Class MainSender : main, programme principal de l'émetteur. Lit les données de la commande d'exécution de l'émetteur (numéro de machine, numéro de port, nom de fichier, et méthode go-back-n). Établit une connexion de l'émetteur avec le récepteur, permet à l'émetteur d'envoyer le contenu du fichier donné au récepteur, et ferme la connexion.

MyTest

Class TrameTest :

- **Test_Tobytes()** : teste la conversion en bytes par ToBytes()
- **Test_ToBytes1()** : teste la conversion en bytes par ToBytes() et un constructeur de trame
- **Test_GetTrame()** : teste la méthode getTrame qui lit le stream et renvoie la trame
- **Test_CRC_shouldNotMatch()** : teste le calcul du CRC et la comparaison CRC
- **Test_CRC_shouldMatch()** : teste le calcul du CRC et la comparaison CRC
- **Test_bytestuff()** : teste le byte stuffing