IUT de Lens - DUT Informatique – S4 IPI - Année 2021/2022

TP2 - Sockets mode connecté (TCP)

Jean-François Condotta

Nous souhaitons réaliser une application permettant à deux utilisateurs de dialoguer à travers le réseau. Pour se faire nous utiliserons les sockets en mode connecté (TCP). Le serveur correspondra à un des utilisateurs, le client à l'autre utilisateur. Les utilisateurs s'enverront alternativement des messages (chaînes de caractères). Le client débutera l'échange et le terminera en envoyant la chaîne 'FIN'. Deux classes devront être écrites : la classe Serveur et la classe Client.

Le programme du client sera paramétré par l'adresse IP et le port du serveur. Le programme du serveur sera paramétré par le port sur lequel le serveur attendra la demande de connexion du client.

Exercice 1 (La classe Serveur)

La classe Serveur aura pour seul attribut l'attribut serverSocket de classe ServerSocket.

Question 1

Dans la classe Serveur, écrire la méthode **enregistrementService**(int port) permettant au serveur d'enregistrer son service sur un port donné en paramètre. Quatre clients en attente au maximum seront permis. Cette méthode mettra à jour l'attribut serverSocket avec l'instance de la classe ServerSocket qui permettra au serveur d'être à l'écoute. Un message « Serveur en attente » sera affiché sur la console. Si l'enregistrement n'est pas possible, le programme s'arrête en retournant un code de sortie 1.

Question 2

Dans la classe Serveur, écrire la méthode **Socket nouvelleConnexion()** permettant d'attendre une nouvelle connexion et de retourner l'instance de la classe Socket qui sera utilisée pour la communication avec le client. Un message « Serveur en attente d'une connexion » sera affiché lors de l'attente et un message « Nouvelle connexion » sera affiché sur la console après une nouvelle connexion. Si l'attente ou la création d'une nouvelle connexion ne sont pas possibles, le programme s'arrête en retournant un code de sortie 1.

Question 3

Dans la classe Serveur, écrire la méthode **BufferedReader fluxEntrant(Socket socket)** permettant d'obtenir un flux entrant instance de la classe BufferedReader du Socket passé en paramètre. Pour tout problème, le programme s'arrêtera en retournant un code de sortie 1.

Question 4

Dans la classe Serveur, écrire la méthode **PrintWriter fluxSortant(Socket socket)** permettant d'obtenir un flux sortant instance de la classe PrintWriter du Socket passé

en paramètre. Pour tout problème, le programme s'arrêtera en retournant un code de sortie 1.

Question 5

Dans la classe Serveur, écrire la méthode **void dialogue(Socket socket)**. Cette méthode gère le dialogue avec le client. Le serveur attend un message du client et l'affiche, puis envoie un message et ainsi de suite. Le dialogue s'arrête lorsque le message reçu par le client est la chaîne de caractères « FIN ». Pour tout problème, le programme s'arrêtera en retournant un code de sortie 1.

Question 6

Dans la classe Serveur, écrire la méthode **void principal()**. Cette méthode permet d'enregistrer l'enregistrement du serveur sur un port (2000). Elle contient également une boucle sans fin permettant au serveur de se connecter avec un nouveau client et de se connecter avec lui.

Question 7

Dans la classe Serveur, écrire la méthode **public static void main(String[] args)**. Cette méthode permet simplement de lancer un nouveau serveur.

Exercice 2 (La classe Client)

La classe Client aura pour seul attribut, l'attribut socket de classe Socket.

Question 1

Dans la classe Client, écrire la méthode **void connexion(InetAddress adresseServeur,int port)** permettant au client de se connecter à un serveur. Cette méthode mettra à jour l'attribut socket avec l'instance de la classe socket qui sera utilisée pour la communication avec le serveur. Si l'enregistrement n'est pas possible, le programme s'arrête en retournant un code de sortie 1.

Question 2

Définir les méthodes **BufferedReader fluxEntrant(), PrintWriter fluxSortant(), void dialogue(), void principal()** et **public static void main(String[] args)** pour la classe Client.

Exemple d'affichage (côté serveur)

```
*** Serveur en attente ***
*** Serveur en attente d'une nouvelle connexion ***
*** Nouvelle connexion ***
Client >Bonjour Serveur !
Serveur >Bonjour Client !
Client >Je dois travailler !
Serveur >Ok.
Client >FIN
*** Serveur en attente d'une nouvelle connexion ***
```

Exemple d'affichage (côté client)

```
*** Connexion établie ***

Client >Bonjour Serveur !
Serveur >Bonjour Client !
Client >Je dois travailler !
Serveur >Ok.
Client >FIN
```