

Laboratoire 2
Séries de Fourier et réseau



CEG 3585 - Communication de données et réseautage

Université d'Ottawa

Professeur : Mohamed Ali Ibrahim

Noms et numéros des étudiants :

Gbegbe Decaho Jacques 300094197
Sissoko Ramatoullaye Bahio 300144949

Date de soumission:26-02-2023

Table des matières:

1. Description du problème
2. Explication du programme de solution
3. Document Représentatif (UML)
4. Captures d'écran
5. Discussion
6. Conclusion

1- Description du problème

L'objectif du laboratoire est, dans un premier temps, de trouver et de programmer la série de Fourier de chaque fonction du tableau 1 et d'afficher la courbe correspondante sur la machine locale. Dans un deuxième temps, vous allez mettre en œuvre un système client-serveur pour envoyer les données de fonction représentant la courbe correspondante sur le réseau afin d'afficher ces courbes dans une machine distante.

2- Explication du programme de solution

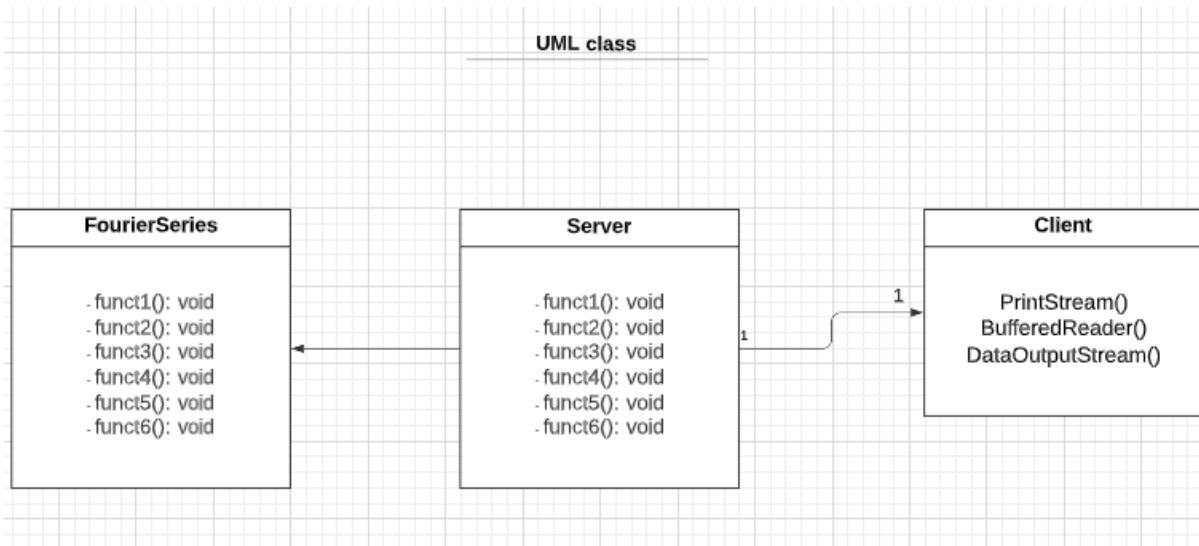
Dans ce lab, nous avons été amené à nous familiariser un peu plus avec les socket et les threads. Ce programme traite de certaines fonctions de la série de fourier qui ont été fournies dans un tableau dans le laboratoire. Pour pouvoir résoudre ce problème, nous avons configuré une classe Server.java et Client.java dans laquelle nous avons utilisé des sockets.

Afin de configurer notre thread, dans les deux fonctions nous avons utilisé certaines méthode de la bibliothèque java tel que (PrintStream, BufferedReader et DataOutputStream) qui chacun:

- PrintStream: afin d'envoyer des données
- BufferedReader: afin de faire la lecture des données reçues
- DataOutputStream: afin d'envoyer des données entre les 2 classes

Afin de faciliter notre travail, nous avons implémenté directement les fonctions de la série de fourier dans la classe Server ceci pour ne pas avoir à faire appel à une autre classe comme intermédiaire. Ce qui en somme à rendu les interactions avec la classe client plus fluide et facile à concevoir pour nous. Puis à la fin de nos 2 classes nous avons fermé les sockets qui servaient à maintenir les thread et la connexion entre eux.

3- Document Représentatif (UML)



4- Capture d'écran

```

Server.java
no usages
13 >    class Server {
14      //Square wave
15      1 usage
16      static void funct1(){...}
17      //Rectangular pulse train
18      1 usage
19      static void funct2(){...}
20      //Sawtooth wave
21      1 usage
22      static void funct3(){...}
23      //Triangular wave
24      1 usage
25      static void funct4(){...}
26      //Half-wave rectified sine
27      1 usage
28      static void funct5(){...}
29      //Full-wave rectified sine
30      1 usage
31      static void funct6(){...}
32      no usages
33      public static void main(String[] args)
34        throws Exception
35      {
36      }

Client.java
no usages
13 >    class Client {
14      no usages
15      public static void main(String args[])
16        throws Exception
17      {
18
19        System.out.println("Client connexion....");
20        // Create client socket
21        Socket socket = new Socket( host: "localhost", port: 4444 );
22        System.out.print("Enter you name: ");
23
24        // to send data to the server
25        DataOutputStream dos
26          = new DataOutputStream(
27            socket.getOutputStream());
28
29        // to read data coming from the server
30        BufferedReader br
31          = new BufferedReader(
32            new InputStreamReader(
33              socket.getInputStream()));
34      }
35    }
36  }

Server
C:\Users\decah\.jdks\openjdk-19.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2023.2.1\lib\idea_rt.jar" -Dfile.encoding=UTF-8
↑ Server is waiting.....
↓ Connection established
→ Ce serveur comprend les méthodes de calcul de
   la série de Fourier du tableau1 du laboratoire 2.....

```

Lancement du serveur

```

Server.java
no usages
13 >    class Server {
14      //Square wave
15      1 usage
16      static void funct1(){...}
17      //Rectangular pulse train
18      1 usage
19      static void funct2(){...}
20      //Sawtooth wave
21      1 usage
22      static void funct3(){...}
23      //Triangular wave
24      1 usage
25      static void funct4(){...}
26      //Half-wave rectified sine
27      1 usage
28      static void funct5(){...}
29      //Full-wave rectified sine
30      1 usage
31      static void funct6(){...}
32      no usages
33      public static void main(String[] args)
34        throws Exception
35      {
36      }

Client.java
no usages
13 >    class Client {
14      no usages
15      public static void main(String args[])
16        throws Exception
17      {
18
19        System.out.println("Client connexion....");
20        // Create client socket
21        Socket socket = new Socket( host: "localhost", port: 4444 );
22        System.out.print("Enter you name: ");
23
24        // to send data to the server
25        DataOutputStream dos
26          = new DataOutputStream(
27            socket.getOutputStream());
28
29        // to read data coming from the server
30        BufferedReader br
31          = new BufferedReader(
32            new InputStreamReader(
33              socket.getInputStream()));
34      }
35    }
36  }

Server
C:\Users\decah\.jdks\openjdk-19.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2023.2.1\lib\idea_rt.jar" -Dfile.encoding=UTF-8
↑ Server is waiting.....
↓ Connection established
→ Ce serveur comprend les méthodes de calcul de
   la série de Fourier du tableau1 du laboratoire 2.....

```

Lancement du client

```

Server
Server is waiting.....
Connection established
Ce serveur comprend les méthodes de calcul de
la série de Fourier du tableau1 du laboratoire 2.....
Press 'ok or enter' pour continuer : ok

```

entrer de “Ok” dans le server pour continuer

```

Client
C:\Users\decah\.jdks\openjdk-19.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2023.2.1\lib\idea_rt.jar" -Dfile.encoding=UTF-8
↑ Client connexion.....
↓ Enter you name: Decaho
→ Entrez le numéro de la fonction voulu (1 à 6):
4

```

Client entre le numéro du modèle de la série de fourier qu'il souhaite calculer

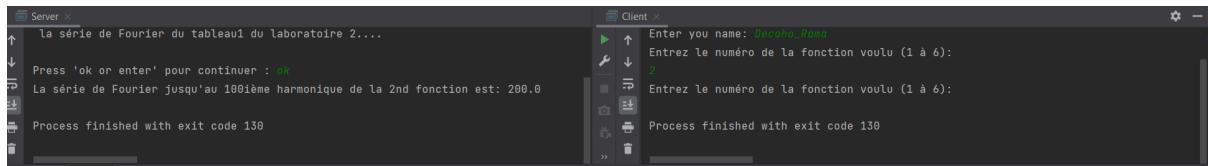
```

Server
Connection established
Ce serveur comprend les méthodes de calcul de
la série de Fourier du tableau1 du laboratoire 2.....
Press 'ok or enter' pour continuer : ok
La série de Fourier jusqu'au 100ième harmonique de la 4th fonction est: -498.3334336414654

Client
C:\Users\decah\.jdks\openjdk-19.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2023.2.1\lib\idea_rt.jar" -Dfile.encoding=UTF-8
↑ Client connexion.....
↓ Enter you name: Decaho
→ Entrez le numéro de la fonction voulu (1 à 6):
4
Entrez le numéro de la fonction voulu (1 à 6):

```

Affichage du résultat dans la boîte de dialogue du serveur



Fermeture du socket et des programmes

5- Discussion

Dans ce laboratoire on a utilisé toutes les données misent à notre disposition pour mener à bien le travail demandé. Néanmoins on a eu quelques difficultés notamment la connexion entre le serveur et le client tout en faisant un appel de fonction dans un thread en cours. Ceci a été la partie qui nous a donné le plus de mal à être réalisé mais toutefois ce fut une belle expérience de nous familiariser un peu plus avec les sockets et les thread. Pour la résolution de ce problème nous avons simplement implémenter la méthode directement dans le serveur de ce fait le client grâce aux sockets pouvait obtenir les résultats directement via le serveur sans passer par une classe intermédiaire. Finalement, ce fut une expérience différente vu que comparé à un serveur sur lequel 2 clients discutent, celui-ci ne prend en compte qu'un seul client mais n'interagit pas directement avec lui, il fait juste office de canal sur lequel le client peut recevoir la réponse à sa demande.

6- Conclusion

L'objectif du laboratoire 2 était, dans un premier temps, de trouver et de programmer la série de Fourier du tableau 1 et d'afficher la courbe correspondante sur la machine locale. Dans un deuxième temps, vous allez mettre en œuvre un système client-serveur pour envoyer les données de fonction représentant la courbe correspondante sur le réseau afin d'afficher ces courbes dans une machine distante. L'application a été conçue avec le langage de programmation Java.

En somme, ce laboratoire a été gratifiant pour nous car il nous a permis d'approfondir nos connaissances en matière de sockets, de clavardage entre clients, la connexion client-serveur et les séries de fourier. On a pu affronter et régler les différents problèmes rencontrés lors de l'implémentation des différentes classes et nous nous en servirons pour les prochaines expériences.