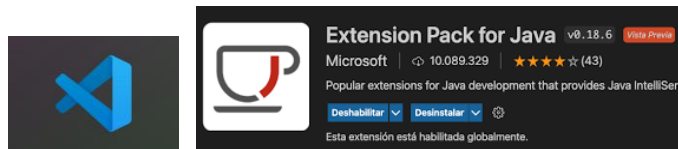


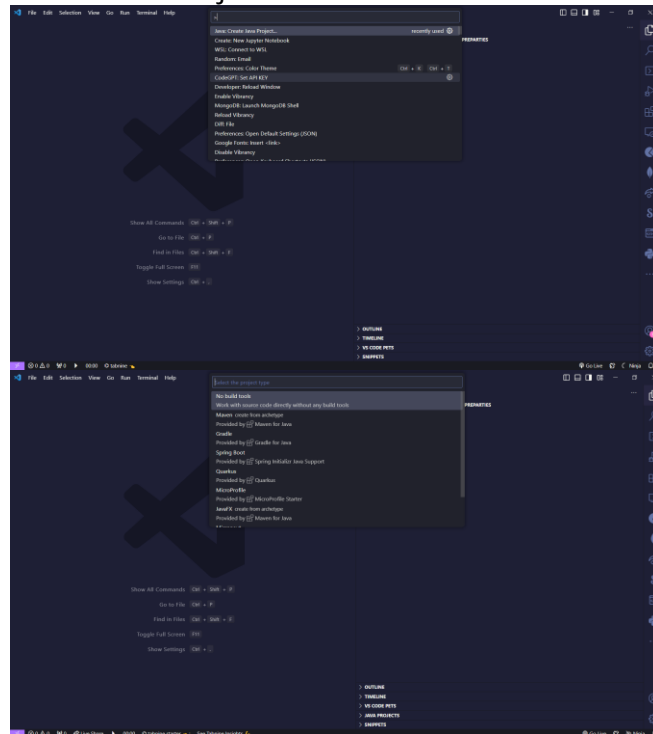
Tp1 : Développement d'applications réparties  
Hammami Mouhamed Yassine  
GLSI3-1C-gp2

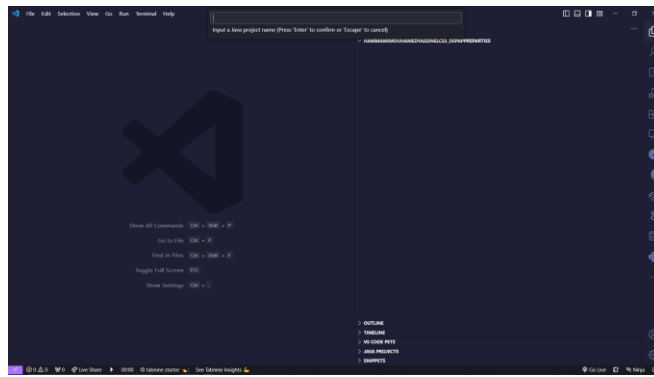
### Environnement de travail :

- L'éditeur du text VSCode + Java Extension pack :

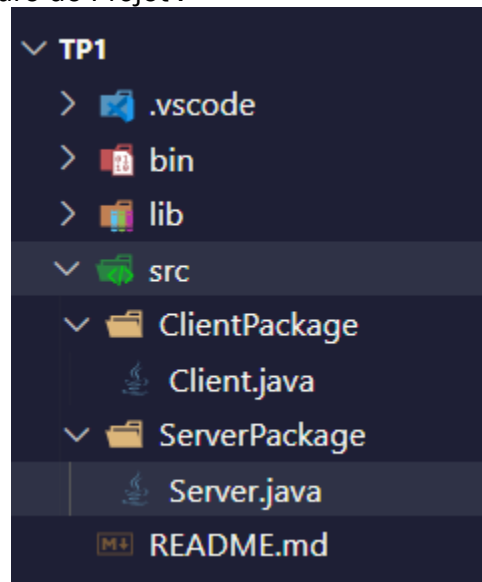


- Création de Projet





- Structure de Projet :



- Serveur.java

```
Server.java x Client.java
src > ServerPackage > Server.java > ...
1 package ServerPackage;
2
3 import java.net.ServerSocket;
4
5 public class Server {
6     Run | Debug
7     public static void main(String[] args) throws Exception {
8         ServerSocket socketServeur = new ServerSocket
9         (port:8000);
10        System.out.println(x:"Serveur est en attent d'un
11        client ");
12        socketServeur.accept();
13        System.out.println(x:"un client est connecté");
14        socketServeur.close();
15    }
16 }
```

- Client.java

```

src > ClientPackage > Client.java > Client > main(String[])
1 package ClientPackage;
2
3 import java.net.Socket;
4
5 public class Client {
6     public static void main(String[] args) throws Exception {
7         System.out.println(x:"client non connecté");
8         Socket socketClient = new Socket(host:"localhost",
9             port:8000);
10        System.out.println(x:"Client Connecté");
11
12        socketClient.close();
13        // Les Sockets sont des ressources alors il faut les
14        // fermer a la fin du methode
15    }
16 }

```

### Remarque :

On ajoute throws Exception pour les fonctions main car les constructeurs Socket(), ServerSocket() lancent des IOException on peut la remplacer par un bloc try catch.

### Interprétation selon scénario :

Lorsqu'on ouvre le programme Serveur seulement il affiche 'client non connecté' car la méthode accept() bloque le code qui la suivent jusqu'à l'ouverture du Programme client.

→ Si on lance le programme Client une erreur se produit car il n'y a pas des sockets actifs sur le port de l'adresse entrée,

```

1 package ClientPackage;
2
3 import java.io.IOException;
4 import java.net.Socket;
5
6 public class Client {
7     public static void main(String[] args) throws IOException
8     {
9         System.out.println(x:"client non connecté");
10        Socket socketClient = new Socket(host:"localhost",
11            port:8000);
12        System.out.println(x:"Client Connecté");
13
14        socketClient.close();
15        // Les Sockets sont des ressources alors il faut les
16        // fermer a la fin du methode
17    }
18 }

```

```

Powershell 7.3.8
Hey friend

In an effort to grow oh-my-posh, the decision was made to no longer support the PowerShell module. Over the past year, the added benefit of the module disappeared, while the burden of maintaining it increased.

However, this doesn't mean oh-my-posh disappears from your terminal, it just means that you'll have to use a different tool to install it.

All you need to do, is follow the migration guide here:
https://ohmyposh.dev/docs/migrating
Loading personal and system profiles took 144ms.
PS [?] & "C:\Program Files\Java\jdk-21\bin\java.exe" "-Xc:showCodeDetail=
IOExceptionMessages" "-cp" "C:\Users\Yac\Desktop\utman\Iphomedrass\src\CS3\DepAppParties\TP1\bin\" "ClientPackage.Client"
client non connecté
Exception in thread "main" java.net.ConnectException: Connection refused: co
nnect
    at java.base/sun.nio.ch.Net.connect0(Native Method)
    at java.base/sun.nio.ch.Net.connect(Net.java:509)
    at java.base/sun.nio.ch.Net.connect(Net.java:570)
    at java.base/sun.nio.ch.AsSocketImpl.connect(AsSocketImpl.java:583)
    at java.base/java.net.SocketImpl.connect(SocketImpl.java:327)
    at java.base/java.net.SocksSocketImpl.connect(SocksSocketImpl.java:327)
    at java.base/java.net.Socket.connect(Socket.java:751)
    at java.base/java.net.Socket.connect(Socket.java:606)
    at java.base/java.net.Socket.<init>(Socket.java:555)
    at java.base/java.net.Socket.<init>(Socket.java:524)
    at ClientPackage.Client.main(Client.java:9)

```

→ Si on Connecte sur un port invalide une erreur se produite

The screenshot shows an IDE with a file named `Client.java` in the `ClientPackage` package. The code defines a `Client` class with a `main` method that attempts to connect to `localhost` on port 8000. The output window shows the following error:

```
Exception in thread "main" java.net.ConnectException: Connection refused: co
nnect
    at java.base/sun.nio.ch.Net.connect0(Native Method)
    at java.base/sun.nio.ch.Net.connect(Net.java:589)
    at java.base/sun.nio.ch.Net.connect(Net.java:578)
    at java.base/sun.nio.ch.NioSocketImpl.connect(NioSocketImpl.java:583)
    at java.base/java.net.SocksSocketImpl.connect(SocksSocketImpl.java:3
27)
    at java.base/java.net.Socket.connect(Socket.java:751)
    at java.base/java.net.Socket.connect(Socket.java:686)
    at java.base/java.net.Socket.<init>(Socket.java:555)
    at java.base/java.net.Socket.<init>(Socket.java:524)
    at ClientPackage.Client.main(Client.java:9)
    at ClientPackage.Client.main(Client.java:9)
```

### Question 4-f)

On peut remplacer «localhost » par la methode statisque de la classe InetAddress  
`InetAddress.getLocalHost()`

The screenshot shows the same IDE with the `Client.java` file. The code has been modified to use `InetAddress.getLocalHost()` instead of `localhost`. The output window shows the following error:

```
Exception in thread "main" java.net.ConnectException: Connection refused: co
nnect
    at java.base/sun.nio.ch.Net.connect0(Native Method)
    at java.base/sun.nio.ch.Net.connect(Net.java:589)
    at java.base/sun.nio.ch.Net.connect(Net.java:578)
    at java.base/sun.nio.ch.NioSocketImpl.connect(NioSocketImpl.java:583)
    at java.base/java.net.SocksSocketImpl.connect(SocksSocketImpl.java:3
27)
    at java.base/java.net.Socket.connect(Socket.java:751)
    at java.base/java.net.Socket.connect(Socket.java:686)
    at java.base/java.net.Socket.<init>(Socket.java:555)
    at java.base/java.net.Socket.<init>(Socket.java:524)
    at ClientPackage.Client.main(Client.java:10)
    at ClientPackage.Client.main(Client.java:10)
```

## • Exercice 2 :

The screenshot shows an IDE with a Java file named `Client.java` and a terminal window. The code defines a `Server` class with a `main` method that listens on port 8000. It uses `ServerSocket` to accept connections and `InputStream`/`OutputStream` to communicate with the client. The terminal output shows the server starting and waiting for a client connection.

```

1 package ServerPackage;
2
3 import java.io.IOException;
4 import java.io.InputStream;
5 import java.io.OutputStream;
6 import java.net.ServerSocket;
7 import java.net.Socket;
8
9 public class Server {
10     public static void main(String[] args) throws IOException
11     {
12         ServerSocket socketServeur = new ServerSocket
13         (port:8000);
14         System.out.println("Serveur est en attente d'un
15         client ");
16         Socket socket = socketServeur.accept();
17         System.out.println("un client est connecté");
18         // EX2
19         // Pour lire les flux de données envoyé par le client
20         InputStream srvtInput = socket.getInputStream();
21         // Pour envoyer des flux de données au client
22         OutputStream srvtOutput = socket.getOutputStream();
23
24         int val = srvtInput.read();
25         System.out.println("Reçu");
26         int rslt = val * 5;
27         srvtOutput.write(rslt);
28         System.out.println("Terminé");
29
30         socketServeur.close();
31     }
32 }

```

Terminal output:

```

added benefit of the module disappeared, while the burden of
maintaining it increased.

However, this doesn't mean oh-my-posh disappears from your
terminal, it just means that you'll have to use a different
tool to install it.

All you need to do, is follow the migration guide here:

https://ohmyposh.dev/docs/migrating
Loading personal and system profiles took 1468ms.
TP1 & 'C:\Program Files\Java\jdk-21\bin\java.exe' -XX:+ShowCode
DetailsInExceptionMessages -cp 'C:\Users\Yacine\Desktop\Hamam\Houham
edYassin\CS3_DepAppReparties\TP1\bin' 'ServerPackage.Server'
Serveur est en attente d'un client
un client est connecté
Reçu
Terminé
15s 522ms 49% 50% 21:33:53

```

The screenshot shows an IDE with a Java file named `Client.java` and a terminal window. The code defines a `Client` class with a `main` method that connects to the server on port 8000. It uses `Socket` to establish the connection and `InputStream`/`OutputStream` to communicate with the server. The terminal output shows the client starting and connecting to the server.

```

13 System.out.println("Client non connecté");
14 Socket socketClient = new Socket(InetAddress.
15     getLocalHost(), port:8000);
16 System.out.println("Client Connecté");
17
18 // EX2
19 // Pour lire les flux de données envoyé par le serveur
20 InputStream cltInput = socketClient.getInputStream();
21 // Pour envoyer des flux de données au serveur
22 OutputStream cltOutput = socketClient.getOutputStream();
23
24 Scanner Scanner = new Scanner(System.in);
25 System.out.println("Donner le nombre désiré");
26 int val = 0;
27 try {
28     val = Scanner.nextInt();
29 } catch (InputMismatchException e) {
30     System.out.println("Merci d'entrer un entier");
31 }
32 cltOutput.write(val);
33 int rslt = cltInput.read();
34 System.out.println("La resultat est: " + rslt);
35
36 socketClient.close();
37 // Les Sockets sont des ressources alors il faut les
38 // fermer a la fin du methode
39 }
40 }

```

Terminal output:

```

added benefit of the module disappeared, while the burden of
maintaining it increased.

However, this doesn't mean oh-my-posh disappears from your
terminal, it just means that you'll have to use a different
tool to install it.

All you need to do, is follow the migration guide here:

https://ohmyposh.dev/docs/migrating
Loading personal and system profiles took 1468ms.
TP1 & 'C:\Program Files\Java\jdk-21\bin\java.exe' -XX:+ShowCode
DetailsInExceptionMessages -cp 'C:\Users\Yacine\Desktop\Hamam\Houham
edYassin\CS3_DepAppReparties\TP1\bin' 'ServerPackage.Server'
Serveur est en attente d'un client
un client est connecté
Reçu
Terminé
15s 522ms 49% 50% 21:33:53

```

## • Gestion des exceptions :

The screenshot shows an IDE with a Java file named `Client.java` and a terminal window. The code defines a `Server` class with a `main` method that listens on port 8000. It uses `ServerSocket` to accept connections and `InputStream`/`OutputStream` to communicate with the client. The terminal output shows the server starting and waiting for a client connection.

```

4 import java.io.InputStream;
5 import java.io.OutputStream;
6 import java.net.ServerSocket;
7 import java.net.Socket;
8
9 public class Server {
10     public static void main(String[] args) throws IOException
11     {
12         ServerSocket socketServeur = new ServerSocket
13         (port:8000);
14         System.out.println("Serveur est en attente d'un
15         client ");
16         Socket socket = socketServeur.accept();
17         System.out.println("un client est connecté");
18         // EX2
19         // Pour lire les flux de données envoyé par le client
20         InputStream srvtInput = socket.getInputStream();
21         // Pour envoyer des flux de données au client
22         OutputStream srvtOutput = socket.getOutputStream();
23
24         int val = srvtInput.read();
25         System.out.println("Reçu");
26         int rslt = val * 5;
27         srvtOutput.write(rslt);
28         System.out.println("Terminé");
29
30         socketServeur.close();
31     }
32 }

```

Terminal output:

```

All you need to do, is follow the migration guide here:

https://ohmyposh.dev/docs/migrating
Loading personal and system profiles took 1468ms.
TP1 & 'C:\Program Files\Java\jdk-21\bin\java.exe' -XX:+ShowCode
DetailsInExceptionMessages -cp 'C:\Users\Yacine\Desktop\Hamam\Houham
edYassin\CS3_DepAppReparties\TP1\bin' 'ServerPackage.Server'
Serveur est en attente d'un client
un client est connecté
Reçu
Terminé
18s 529ms 50% 50% 21:36:49

```

### Explication :

Les méthodes `getInputStream()` `getOutputStream()` n'accepte que des octet tous valeur supérieur lance une erreur .