***Exercice3***

**Objectif de l’exercice :** La création d’une application distribuée qui e est composée d'un client et d'un serveur.Ainsi, le client envoie des données concernant des personnes au serveur. Ces données sont composées de 2 informations : un entier (int) pour l'âge de la personne et une chaîne de caractères (String) pour son nom. A chaque envoi d'informations sur une personne , le serveur retourne au client un entier correspondant à l'identificateur du client.

**Coté Serveur:**

1. **Définition du port d'écoute :** Configurez le serveur pour écouter sur un port spécifique, tel que 10000.
2. **Création du ServerSocket :** Créez un objet ServerSocket et liez-le au port spécifié.
3. **Attente et acceptation de la connexion :** Le serveur est en attente qu'un client se connecte en utilisant la méthode accept() du ServerSocket. Lorsqu'une connexion est établie, le serveur accepte la connexion, créant un nouveau socket dédié à la communication avec le client.
4. **Création des flux de communication :** Crée des flux d'entrée et de sortie pour envoyer et recevoir des objets sur le socket.
5. **Lecture des informations sur la personne du client :** Le serveur lit un entier représentant l'âge et une chaîne de caractères représentant le nom du client à partir des flux d'entrée.
6. **Affichage des informations reçues :** Affichez côté serveur les informations reçues du client, telles que l'âge et le nom.
7. **Attribution d'un identificateur unique au client :** Le serveur attribue un identifiant unique (clientId) au client pour le distinguer.
8. **Retour de l'identificateur au client :** Envoyez l'identificateur unique au client en utilisant le flux de sortie.
9. **Incrémentation de l'identifiant unique :** Incrémentez l'identifiant unique pour le prochain client, assurant ainsi l'unicité des identifiants.
10. **Fermeture de la connexion avec le client :** Une fois la transaction terminée, le serveur ferme la connexion avec le client.
11. **Gestion des erreurs :** Attrape toute exception pouvant survenir pendant l'exécution du code et affiche un message d'erreur correspondant.

**Côté client**

1. **Définition de l'adresse IP et du port du serveur :** Configurez le client avec l'adresse IP (par exemple, "localhost") et le port du serveur (par exemple, 10000).
2. **Création du socket client :** Créez un objet Socket et connectez-vous au serveur en utilisant l'adresse IP et le port spécifiés.
3. **Création des flux de communication :** Instanciez des flux de sortie (ObjectOutputStream) et d'entrée (ObjectInputStream) pour faciliter l'échange d'objets entre le client et le serveur.
4. **Saisie des informations sur la personne :** Le client utilise un objet Scanner pour saisir l'âge et le nom de la personne à partir de la console.
5. **Envoi des informations au serveur :** Le client envoie l'âge et le nom de la personne au serveur à l'aide du flux de sortie.
6. **Réception de l'identificateur unique du serveur :** Le client lit l'identificateur unique renvoyé par le serveur à l'aide du flux d'entrée.
7. **Affichage de l'identificateur unique :** Le client affiche l'identificateur unique reçu du serveur.
8. **Fermeture de la connexion avec le serveur :** Une fois la transaction terminée, le client ferme la connexion avec le serveur.
9. **Gestion des erreurs :** capture toutes les exceptions qui pourraient se produire lors de l'exécution du code dans le bloc try et les affiche dans la console.