

 <p>Faculté des Sciences de Sfax Département d'informatique et des Communications</p>	<i>Année Universitaire</i> 2025 – 2026
Matière : Développement d'applications réparties Responsable du cours : Corinne Amel ZAYANI Enseignantes des TPs : Fatma ELLOUZE, Zeineb DAOUD	<i>Auditoire</i> LSI 3

TP 3 : Serveur Multi-threads - sockets en mode connecté TCP

Activité 3-2 : Amélioration de l'application calculatrice

Objectifs

Cette activité vise à :

- Comprendre et appliquer le principe d'un serveur multithread basé sur les sockets TCP en Java.

Environnement

- JDK >=1.8
- Eclipse IDE for Java developers Ou IntelliJ IDAE > 1.8 OU VS code
- Git

Type de l'activité

- Individuel

Ressources

- Consultez le cours “Développement d’applications réparties” : **séances 2, 3 et 4** (CC / objet / serveur multi-thread), 03/10 - 10/10 – 17/10/2025.
- Applications Client/Serveur développées dans les **TP1 et TP2**.

Livrables

- Le lien du repository dans l'espace de dépôt de la plateforme ENT
- Le fichier READ ME

Enoncé

Dans cette activité, vous allez reprendre le **service de calculatrice** développé dans **l'activité 2-2** du **TP2**. L'objectif est d'améliorer ce service afin de permettre au serveur d'accepter plusieurs connexions clientes simultanément et de traiter en parallèle les opérations envoyées par chacun d'eux.

Travail demandé

Vous allez transformer votre application en un serveur multi-thread, capable de créer un thread dédié pour chaque client connecté, tout en maintenant une communication indépendante entre le serveur et chaque client.

Cette activité se déroule en quatre étapes :

Étape 1 : Préparation de l'espace de travail

Créez un document Word nommé « **VotreNomPrenom_Activite_3_2.docx** » pour l'insertion des captures d'écran de chaque étape réalisée.

Étape 2 : Création du serveur multi-thread pour le service de calculatrice

Dans cette étape, vous êtes invités à étendre le code développé dans **l'activité 3-1** afin d'y intégrer les traitements nécessaires à la mise en œuvre du service de calculatrice sur un serveur multithread. Pour ce faire, vous devez :

1. Intégrez le service de calculatrice dans le serveur multithread en utilisant une communication client/serveur par objets.
2. Mettez à jour la classe Client afin qu'elle communique avec le serveur selon le même principe de communication par objets.
3. Testez le service en établissant des connexions entre le serveur et plusieurs clients sur votre machine locale.
4. Vérifiez le fonctionnement en réseau local, en testant la communication entre plusieurs machines connectées, dont l'une jouera le rôle de serveur et les autres celui de clients.

Étape 3 : Gestion d'un accès concurrent

On introduit un compteur global partagé indiquant le nombre total d'opérations traitées par le serveur. Ce compteur est une ressource critique, partagée entre les threads ClientProcess. L'accès à cette variable doit être synchronisé afin d'éviter les conflits d'accès concurrents.

À chaque opération calculée :

- le compteur global est incrémenté,
 - sa nouvelle valeur est affichée dans la console du serveur.
1. Testez le service avec plusieurs clients en parallèle.
 2. Vérifiez que le compteur global s'incrémente correctement et reste cohérent.
 3. Testez le service sur plusieurs machines du même réseau.

Étape 4 : Dépôt des livrables

Dans cette dernière étape, vous êtes invités à :

- Déposez votre projet sur GitHub ainsi que le lien du repository dans l'espace de dépôt de la plateforme ENT, et le mentionnez dans le document Word.
- Déposez votre document Word dans l'espace de dépôt sur l'ENT :
 - Ajoutez également le lien vers votre dépôt GitHub.
 - Le document doit contenir des imprimés écrans accompagnés d'une synthèse.