

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

## TP 5 : Implémenter des communications client-serveur via le Common Object Request Broker Architecture (CORBA)

### Activité : Service Banque en CORBA

---

#### Objectifs

- Implémenter une application distribuée complète (Client-Serveur) en Java utilisant le middleware CORBA (Common Object Request Broker Architecture) pour la gestion de comptes bancaires.

#### Environnement

- JDK 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 5 et 6** (Corbapart1 / Corbapart2) 24/10 – 31/10/2025.

#### Livrables

- Le lien du repository dans l'espace de dépôt de la plateforme ENT
- Le document Word

#### Critères d'évaluation :

- Lien partagé avec un TP conforme aux consignes et dans le délai : **5 pts**,
- Lien partagé avec TP incomplet ou partiellement conforme ou hors délais : **2-4 pts**,
- Absence de dépôt ou hors sujet ou en retard d'une période : **0-1 pt**

## Enoncé et travail demandé

L'objectif de cette activité du TP est la création d'une application répartie en utilisant le middleware CORBA où un Serveur gère l'ensemble des comptes bancaires.

Chaque Compte est défini par un **code** et un **solde**. Le serveur doit permettre aux clients d'effectuer les opérations suivantes :

- Créer un compte avec une somme initiale.
- Verser une somme d'argent sur un compte (sachant son code).
- Retirer une somme d'argent d'un compte (sachant son code).
- Consulter un compte (sachant son code).
- Consulter la liste de tous les comptes.
- Effectuer une conversion de montant : Euro (€) vers Dinar Tunisien (DT).

Cette activité se déroule en cinq étapes :

### Étape 1 : Préparation de l'espace de travail

1. Dans votre espace de travail, créez un répertoire sous le nom « **votreNomPrenom\_LSI3\_DevAppReparties\_TP5** ».
2. **Lancement de votre IDE** : Démarrez votre IDE en désignant ce nouveau répertoire comme votre Workspace.
3. **Vérification de l'Environnement** : Assurez-vous que le JDK (1.8 recommandé) est installé et que la variable PATH est correctement configurée.
4. Créez un document Word nommé « **VotreNomPrenom\_ActiviteTP5.docx** » pour l'insertion des captures d'écran de chaque étape réalisée.

### Étape 2 : Développement du Serveur (Définition IDL et Implémentation)

L'objectif est de définir l'interface du service bancaire, de générer le code CORBA, et d'implémenter l'objet distant ainsi que le serveur d'annuaire.

1. **Création du Projet Serveur** : Créez un nouveau Projet Java nommé « **TP5\_CorbaBanque\_Server** ».
2. **Écriture de l'Interface IDL** : Créez le fichier « **Banque.idl** » dans le répertoire src. Il doit déclarer les éléments suivants :
  - Un module **corbaBanque**.
  - Une structure (struct) **Compte** (avec code: long et solde: float).

- Un alias (typedef) **tabComptes** (pour un tableau de comptes).
- L'interface **IBanqueRemote** déclarant toutes les méthodes distantes :
  - **creerCompte** qui possède un paramètre « cpte » de type **Compte** en entrée
  - **verser** : qui possède deux paramètres en entrée en « mt » de type **float** et « code » de type **long**
  - **retirer** : qui possède deux paramètres en entrée en « mt » de type **float** et « code » de type **long**
  - **getCompte** : qui retourne un objet de type **Compte** sachant son code.
  - **getComptes** : qui retourne un tableau de comptes.
  - **conversion** qui retourne un **double** possède un paramètre « mt » de type **float** en entrée

### 3. **Compilation IDL** : Compilez « **Banque.idl** » en utilisant l'une de ces deux méthodes :

- Méthode 1 : Ligne de commande
  - Accédez au répertoire *src* de votre projet et tapez *cmd* dans le chemin du répertoire
  - Exécutez la commande de compilation : **idlj -fall -v Banque.idl**
- Méthode 2 : Configuration d'Eclipse
  - Dans la barre de menu, cliquez sur :
    - **Run > External Tools > External Tools Configurations...**
  - Dans le panneau de gauche, faites un clic droit sur **Program**, puis choisissez **New Configuration**.
  - Dans le panneau de droite, saisissez **CompilationIDL** comme nom de configuration.
  - Sous **Location** : cliquez sur **Browse File System...**, puis recherchez le programme **idlj.exe** situé dans le répertoire *bin* du JDK (par exemple : C:\Program Files\Java\jdk1.8.0\bin\).
  - Sous **Working Directory** : cliquez sur **Browse Workspace...** et sélectionnez le répertoire *src* de votre projet.
  - Dans le champ **Arguments** : saisissez :
    - -fall -v Banque.idl
  - Cliquez sur **Apply**, puis sur **Run**.  
Ensuite, faites un **refresh** sur votre projet.

#### 4. Implémentation du Servant :

- a. Créez le package service dans *src*.
- b. Créez la classe « **BanqueImpl** » qui hérite de « **IBanqueRemotePOA** ».
- c. Implémentez concrètement toutes les méthodes de l'interface « **IBanqueRemoteOperations** » (le code métier).

#### 5. Configuration JNDI : Dans le répertoire *src*, créez le fichier *jndi.properties* contenant l'adresse de l'annuaire de noms (naming service) :

```
java.naming.factory.initial = com.sun.jndi.cosnaming.CNCtxFactory
java.naming.provider.url = iiop://localhost:900
```

- #### 6. Développement du Serveur :
- Créez le package **corbaServer** dans lequel vous êtes demandés de créer une classe nommée « **BanqueServer** » dans *src*. Cette classe doit :
- a. initialiser l'ORB,
  - b. initialiser l'adaptateur d'objets (POA) : Pour ce faire, il faut récupérer une référence sur le RootPOA et activer le POAManager,
  - c. configurer l'annuaire JNDI,
  - d. créer le servant (**BanqueImpl**),
  - e. enregistrer sa référence pour la rendre disponible dans l'ORB,
  - f. et mettre l'ORB en attente de requêtes des clients.

### Étape 3 : Développement du Client

L'objectif est de développer l'application Client qui communique avec le serveur via le Stub généré par CORBA.

1. **Création du Projet Client** : Créez un nouveau Projet Java nommé « **TP5\_CorbaBanque\_Client** ».
2. **Préparation du Client** : Copiez les fichiers « **Banque.idl** » et « **jndi.properties** » dans le répertoire « **TP5\_CorbaBanque\_Client/src** ».
3. **Compilation IDL Client** : Compilez de nouveau le fichier « **Banque.idl** » dans le projet Client (de la même manière qu'à l'étape 2).
4. **Développement du Client** : Créez le package « **corbaClient** » dans lequel vous êtes demandés de créer la classe « **BanqueClient** » dans *src*. Cette classe doit :
  - a. Récupérer les propriétés de l'annuaire **JNDI**.

- b. Récupérer la référence de l'objet distant.
- c. Convertir la référence vers le type `IbanqueRemote` (via `narrow` et `IBanqueRemoteHelper`) pour créer le **Stub**.
- d. **Appeler à distance** les méthodes du service bancaire via le Stub (tester chaque méthode).

#### **Étape 4 : Exécution de l'application**

L'exécution doit respecter l'ordre d'initialisation des composants du middleware.

1. **Lancer l'Annuaire (Naming Service)** : Démarrez l'annuaire de noms CORBA avec l'une de ces deux méthodes :

- a. Méthode 1 : Ligne de commande :

- Accédez au dossier du projet serveur, et tapez *cmd* dans le chemin du répertoire du projet serveur,
- Exécutez la commande **tnameserv** ou bien **tnamserv start**

- b. Méthode 2 : Configuration Externe Eclipse

- Cliquez sur le menu
  - **Run > External Tools > External Tools Configurations...**
- Faites un clic avec le bouton droit sur « **Program** » dans le panneau de gauche puis choisissez « **New Configuration** » ;
- Dans le panneau de droite, tapez **tnserv** comme nom ;
- Sous **Location** : Cliquez sur **Browse File System...**, puis recherchez le programme **tnameserv** qui se trouve dans le répertoire *bin* du JDK (par exemple : `C:\Program Files\Java\jdk1.8.0\bin\`);
- Sous **Working Directory** : Cliquez sur **Browse Workspace...** et recherchez le répertoire *src* ou *bin* de votre projet serveur.

2. **Lancer le Serveur** : Exécutez la classe **BanqueServer**.

3. **Lancer le Client** : Exécutez la classe **BanqueClient**.

**NB :**

- Importez les packages nécessaires et ajoutez les exceptions appropriées.
- A chaque étape, il faut insérer les imprimés écrans correspondants dans le document Word créé dans la 1ère étape.

### **Étape 5 : 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.