

---

## Test

**Filière :** GLSI

**A.U :** 2022-2023

**Niveau :** Troisième année

**Durée :** 01H30

**Matière :** Développement d'applications réparties

**Documents :** non autorisés

**Enseignant:** M. SAKKARI

---

### Exercice 1 (14 pts)

Nous disposons d'un service "Calculatrice" permettant la saisie de deux nombres et une opération ( +, − , / , x ) et affichant le résultat. Les méthodes offertes par ce service sont les suivantes:

```
public double somme(double a,double b)
public double soustraction(double a,double b)
public double multiplication(double a,double b)
public double division(double a,double b)
```

On souhaite rendre chacune de ces méthodes accessibles à distance de manière à ce qu'elles définissent l'interface CalculatriceInterface entre le client et le serveur.

**Question 1 :** Écrire l'interface **CalculatriceInterface.java** (4pts)

**Question 2 :** Compléter le fichier suivant: CalculatriceImplementation.java qui matérialise le service qui offre ces méthodes. (4pts)

#### CalculatriceImplementation.java

---

```
package service;

import java.rmi.RemoteException;
import java.rmi.server.UnicastRemoteObject;

@SuppressWarnings("serial")

// à compléter

// ...
```

---

**Question 3 :** Compléter le fichier : CalculatriceServeur.java pour permettre l'enregistrement du service auprès de RMI Registry.(6 pts)

### CalculatriceServeur.java

---

```
package service;

import java.rmi.Naming;
import java.rmi.registry.LocateRegistry;

public class CalculatriceServeur {
    public static void main(String[] args) {
        try {
            //à compléter (Création de registre)
            . . . . .
            //à compléter (Instanciation)
            . . . . .
            //à compléter (Enregistrement)
            . . . . .
            System.out.println("Le serveur est prêt !");
        } catch (Exception e) {
            System.out.println("Échec de l'exécution du serveur :" +e);
        }
    }
}
```

---

**Vous devez remettre cette feuille sans signe distinctif  
(seule sera corrigée pour cette partie)**

## **Exercice 2 QCM (6 pts)**

**Q1.** Une architecture client-serveur représente l'environnement dans lequel des applications de machines clientes communiquent avec des applications de machines de type serveurs. Le serveur

- ☐ initie le contact (parle en premier), ouvre la session
- ☐ est la partie de l'application qui offre un service
- ☐ est à l'écoute des requêtes clientes
- ☐ peut répondre à plusieurs clients simultanément

**Q2.** Dans une architecture client-serveur

- ☐ Le client et le serveur ne sont pas identiques
- ☐ Le client et le serveur sont identiques
- ☐ les parties client et serveur de l'application peuvent s'exécuter sur des systèmes différents
- ☐ Une même machine peut implanter les côtés client et serveur de l'application

**Q3.** Dans une architecture client-serveur, Si toutes les ressources nécessaires sont présentes sur un seul serveur, on parle d'architecture

- ☐ à deux niveaux
- ☐ à 2 tiers
- ☐ à un niveau

**Q4.** CORBA

- ☐ Est un langage de programmation orientée objet
- ☐ Est l'un des langages de programmation les plus utilisés pour la mise en place des applications distribuées.
- ☐ Est basée sur un bus

**Q5.** Pour la mise en place d'objets distribués avec CORBA

- ☐ Les communications sont basées sur le mécanisme d'invocation de procédure distantes.
- ☐ IDL permet d'exprimer la coopération entre les fournisseurs et les utilisateurs de services en séparant l'interface de l'implémentation.

***Bon travail***