

Exercice d'application en .NET Remoting

Nous disposons d'un service qui permet d'offrir des opérations permettant la gestion de son compte courant. Voici le code des méthodes offertes par ce service :

```
void debiter(double montant) {
    }
void crediter(double montant) {
    }
double lireSolde() {
    }
```

1. On souhaite rendre chacune de ces méthodes accessibles à distance de manière à ce qu'elles définissent l'interface entre le client et le serveur. Ecrire cette interface.
2. Déduire la classe qui implémente l'interface.
3. Compléter le fichier `Serveur.cs` pour permettre l'enregistrement du service *Compte* auprès du CLR sous le nom de « Transaction ». Si l'objet *Compte* gère le même compte bancaire, à votre avis, faudra-t-il gérer l'objet en mode *SingleCall* ou en mode *Singleton* ?

```
using System;
using System.Runtime.Remoting;
using System.Runtime.Remoting.Channels;
using System.Runtime.Remoting.Channels.Tcp;
using namespaceCompte;

namespace nameSpaceServeur {
    // Implementation de la classe des objets serveurs.
    class Serveur {
        static void Main() {

            // A compléter

            // 3) Maintient du processus courant.
            Console.WriteLine("Pressez une touche pour stopper le serveur.");
            Console.Read();
        }
    }
}
```

4. Lancer le serveur.

5. Compléter le programme du client qui doit être lancé à partir d'une nouvelle instance de Visual C# 2008.

```

using System;
using System.Runtime.Remoting;
using System.Runtime.Remoting.Channels;
using System.Runtime.Remoting.Channels.Tcp;
using namespaceInterface;

namespace namespaceClient
{
    class Client
    {
        static void Main(string[] Argument)
        {

            // A compléter

            int op = int.Parse(Argument[0]);
            double m = double.Parse(Argument[1]);

            // Appel d'une methode sur l'objet distant.
            try
            {
                switch (op)
                {
                    case 1: obj.debiter(m);
                        break;
                    case 2: obj.crediter(m);
                        break;
                }

                Console.WriteLine("solde courant : {0}" + obj.lireSolde());
            }
            catch (RemotingException e) {
                Console.WriteLine("erreur : {0}" + e.GetBaseException());
            }
            Console.Read();
        }
    }
}

```

Remarques :

- Pour tester ces programmes, télécharger gratuitement *Visual C# 2008 Express Edition* et installez-le, à partir du lien suivant :

<http://msdn.microsoft.com/fr-fr/express/aa975050.aspx>

Pour exécuter correctement le client et le serveur, rajouter dans l'onglet *References* la librairie *System.RunTime.Remoting*.

- Les arguments du programme du client doivent être saisis dans un champ accessible à partir du menu *Projet/Propriétés/Debuguer*. Une valeur réelle doit être saisie avec une virgule (par exemple 13,5) et non avec un point.