

Correction de l'examen

Question de cours (07 pts)

Donnez des réponses brèves et précises aux questions suivantes :

- 1) Quel est le principal inconvénient de la méthode GET ? **(01 pts)**

le principal inconvénient de la méthode GET est l'absence de protection des données. Les paramètres URL envoyés sont non seulement visibles par tous dans la barre d'adresse du navigateur, mais sont également stockés sans chiffrement dans l'historique du navigateur, dans le cache et dans le fichier log du serveur.

- 2) À votre avis, est-ce-que la mise en cache HTTP permet d'accélérer le processus de chargement des pages Web ? Justifiez votre réponse ? **(01 pts)**

Oui, elle permet d'accélérer le processus de chargement des pages Web car, beaucoup de ressources devant être chargées seront récupérées directement à partir du cache du navigateur sans le besoin de les faire demander, à chaque fois, du serveur.

- 3) Soit la portion de la requête HTTP suivante qui est récupérée à l'aide de l'outil « Network » du navigateur :

```
GET /ar/ceil.html HTTP/1.1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9
Connection: keep-alive
Cookie: _ga=GA1.2.308141566.1663223042; _gid=GA1.2.1555325232.1671876470
Host: www.univ-tiaret.dz
Referer: https://www.univ-tiaret.dz/ar/
Sec-Fetch-Dest: document
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64)
sec-ch-ua-platform: "Windows"
Content-Encoding: gzip
```

À votre avis, d'après cette requête :

- a) Qu'est-ce-que le navigateur demande du serveur ? **(0.5 pts)**

Le navigateur demande au serveur de lui envoyer le fichier « ceil.html ».

- b) Quelle est l'URL du document demandé ? **(0.5 pts)**

L'URL du document demandé est comme suit : <https://www.univ-tiaret.dz/ar/ceil.html>

- 4) Pour envoyer un objet d'une classe à distance en utilisant la méthode RMI, cette classe doit implémenter une interface précise. Quelle est cette interface ? et donner sa définition.

L'interface est serialisation, la sérialisation est un mécanisme de conversion de l'état d'un objet en un flux d'octets. La désérialisation est le processus inverse dans lequel le flux d'octets est utilisé pour recréer l'objet Java réel en mémoire. Ce mécanisme est utilisé pour conserver l'objet.

- 5) À quoi sert le langage IDL ? **(01 pts)**

Le langage IDL est un langage qui sert à définir l'interface, principalement des objets, indépendamment d'un langage de programmation particulier.

6) A quoi indiquer une réponse http avec le code 304 ? **(01 pts)**

Une réponse HTTP avec le code 304 indique que le document en cache est toujours valide (c.à.d. n'a pas été modifié du côté serveur), et donc, le corps de cette réponse sera vide et il ne contiendra aucune nouvelle version de la ressource demandée

7) Quelle est la taille maximale d'un message que nous pouvons l'envoyer par socket java (mode connecté) ? **(01 pts)**

La taille du message est n'est pas définie, nous pouvons envoyer des messages de très grande taille.

Exercice 01 (06 pts)

1) Donner uniquement les étapes qu'il faut suivre afin d'établir cette application ? **(1.5 pts)**

1/ Les étapes sont

Etape 1: Création de l'interface du service en IDL (fichier calcul.idl) :

Etape 2 : Exécuter la commande suivante qui permet d'engendrer les fichiers java client et serveur à partir de l'interface IDL : C:\Dell\jdk\bin> idlj -fall "D:\mywork\corba\calcul.idl"

Etape 3 : Création de la Classe java de l'objet « servant » (la classe qui fait le calcul)

Etape 4 : Création de la Classe du serveur :

Etape 5 : Création de la Classe du client

Etape 6 : Exécuter la commandes permettant d'activer l'ORB sur la machine serveur, de lancer le serveur, et d'exécuter le client

```
start orbd -ORBInitialPort 1500
```

2) Proposer une interface du service en IDL (Interface Definition Language) du programme. **(01 pts)**

```
module calcul {  
  
    interface calcul_factoriel {  
  
        double calcul_fact (in int n);  
  
    }; };
```

3) Donner l'implémentation de la classe Servant. **(01.5 pts)**

```
class Calcul_Servant extends calcul_calcul_factorielPOA {  
    private ORB orb;  
  
    public void setORB(ORB orb_val) {  
        orb = orb_val;  
    }  
  
    public int calcul_factoriel (int n) {  
        int result = 1;  
        for (int i = 1; i <= n; i++) {  
            result = result * i;  
        }  
        return result;  
    } };
```

4) Donner la commande permettant d'engendrer les fichiers java client et serveur à partir de l'interface IDL.

(01 pts)

`C:\Dell\jdk\bin> idlj -fall "D:\mywork\corba\calcul.idl"`

5) Donner les commandes permettant d'activer l'ORB sur la machine serveur, de lancer le serveur, et d'exécuter le client. **(01 pts)**

`start orbd -ORBInitialPort 1500`

Exercice 02 (07 pts)

Voici un squelette de code qui illustre la signature de cette méthode offerte par ce service :

```
import java.rmi.Remote;
import java.rmi.RemoteException ;
publ interface IOcategorie extends Remote {
String categorie ( int n) throws RemoteException ;
}
```

- 1) Écrivez la classe d'implémentation correspondant à l'interface écrite dans la question précédente (Veuillez appeler cette classe « Impcategorie » ?). **(03 pts)**

```
public class Impcategorie extends UnicastRemoteObject implements
IOcategorie {
    Impcategorie() throws RemoteException{
        super();
    }
    @Override
    public String categorie(int age) {
        if( age<6 )
return "votre age est inferieure à 6 ans ";
        if( age >= 6 && age <= 7)
return" Vous etes poussin ";
        if( age >= 8 && age <= 9 )
return"vous etes pupille ";
        if( age >= 10 && age <= 11)
return" Vous etes minime ";
        if( age >= 12 )
return"Vous etes Cadet";
        else return "votre age est superieur à 12 ans ";
    }
}
```

- 2) Complétez le fichier suivant : « LancerServeur.java » pour permettre la mise à disposition de l'objet distant voulu tout en supposant que l'annuaire sera dans l'écoute du port 2000 ? . (02 pts)

```
                                //LancerServeur.java
package Serveur;
import java.net.InetAddress;
import java.rmi.registry.Registry;
import java.rmi.registry.LocateRegistry;
import java.rmi.RemoteException;
import java.rmi.server.UnicastRemoteObject;
    public class Server {
        public static void main(String args[]) {
            try {
                Registry annuaire = LocateRegistry.createRegistry(2000);
                System.out.println("createRegistry .... OK "+annuaire);
                Impcategorie categorie = new Impcategorie();
                annuaire.rebind("Hello14", categorie);
            } catch (Exception ex) {
                ex.printStackTrace();
            }
        }
    }
}
```

- 3) Complétez le fichier suivant : « Client.java » . (02 pts)

```
                                //Client.java
import Serveur.IOcategorie;
import java.net.InetAddress;
import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;
import java.util.Scanner;
public class Client {
    private Client() {}
    public static void main(String[] args) {
        try {
            System.out.println("Je suis le client ...");
            Registry annuaire = LocateRegistry.getRegistry("localhost",2000);
            System.out.print ( "Donnez l'age ? " ) ;
            Scanner sc = new Scanner ( System.in ) ;
            IOcategorie hello_interfacel = (IOcategorie)annuaire.lookup("Hello14");

            int n = sc.nextInt( ) ;
            System.out.println("Serveur :"+hello_interfacel.categorie(n));

        } catch (Exception ex) {
            ex.printStackTrace();
        }
    }
}
```