



## ESISAR

### NE448 - Programmation répartie

### Examen de TP

#### Table des matières

1 Consignes générales.....	1
2 Exercice 1 : Table de multiplication UDP (7 points).....	2
2.1 Présentation générale du problème.....	2
2.2 Préparation de votre machine.....	2
2.3 Ecriture du programme.....	2
3 Exercice 2 : Port scanning TCP (13 points).....	4
3.1 Présentation générale du problème.....	4
3.2 Préparation de votre machine.....	4
3.3 Essai manuel.....	5
3.4 Ecriture du programme.....	5

#### 1 Consignes générales

La durée de cet examen de TP est de **1H30**.

Vous devez impérativement travailler sur la station fournie (pas de PC personnel).

Attention : vous devez absolument respecter les consignes suivantes :

- un projet Eclipse par exercice
- le nom du projet Eclipse doit être **examen\_exo1\_[votrenom]** pour l'exercice 1, **examen\_exo2\_[votrenom]** pour l'exercice 2, ...
- vous devez rendre vos travaux en suivant les instructions données par le serveur d'examen dont l'URL est indiquée au tableau.

L'accès à Internet n'est pas autorisé, tous les documents sont autorisés (documents papiers + une clé USB).

## 2 Exercice 1 : Table de multiplication UDP (7 points)

### 2.1 Présentation générale du problème.

Un serveur UDP est lancé sur une machine, ce serveur écoute en UDP sur le port 11000.

Ce serveur fonctionne ainsi :

- il attend un paquet UDP contenant le texte « JOUER »
- quand il reçoit ce paquet avec JOUER, il répond avec un premier paquet UDP contenant un nombre compris entre 0 et 9 (par exemple 2), puis immédiatement après il envoie également un deuxième paquet UDP contenant un nombre compris entre 0 et 9 (par exemple 8)
- vous devez alors lui renvoyer le résultat de la multiplication des 2 nombres dans un paquet UDP , c'est à dire 16 dans notre exemple, avec un point virgule à la fin
- dans notre exemple, vous devez donc retourner 16 ;
- le serveur vous répond « GAGNE » ou « PERDU » si votre réponse est correcte ou non.

L'objectif de cet exercice est de réaliser un programme capable de jouer avec ce serveur.

### 2.2 Préparation de votre machine.

Pour pouvoir tester votre programme, vous allez installer sur votre machine ce serveur UDP.

Pour cela, télécharger le fichier **multiplication.jar** disponible sur le serveur d'examen.

Lancer ce serveur UDP avec la commande

```
java -jar multiplication.jar
```

Vous avez maintenant un serveur UDP écoutant sur un port 11000.

### 2.3 Ecriture du programme

Ecrivez maintenant une classe `ClientMultiplication` capable de jouer seule face à ce serveur. Votre classe doit faire 10 parties consécutives.

Si on exécute le main de cette classe, on obtient l'affichage suivant :

```
=====
Début de la partie 1
Envoi d'un paquet UDP avec JOUER
Le serveur a répondu 9 et 5
Envoi d'un paquet UDP avec 45;
Le serveur a répondu :GAGNE
Fin de la partie 1
=====
Début de la partie 2
Envoi d'un paquet UDP avec JOUER
...
=====
Début de la partie 10
Envoi d'un paquet UDP avec JOUER
Le serveur a répondu 7 et 2
Envoi d'un paquet UDP avec 14;
Le serveur a répondu :GAGNE
Fin de la partie 10
```

### 3 Exercice 2 : Port scanning TCP (13 points)

#### 3.1 Présentation générale du problème.

Un serveur TCP est lancé sur une machine, ce serveur écoute en TCP sur **tous** les ports compris entre 30 000 et 32 000 (inclus).

Ce serveur fonctionne ainsi :

- le serveur écoute sur **tous** les ports compris entre 30 000 et 32 000 (inclus)
- quand un client se connecte sur un de ces ports, le serveur attend un message contenant une seule lettre : ?
- quand le serveur reçoit cette lettre ? , **le serveur répond 5 secondes plus tard**
- la réponse du serveur peut être
  - VOUS AVEZ GAGNE!
  - VOUS AVEZ PERDU!
  - VOUS AVEZ FAIT UNE ERREUR.

Un seul numéro de port répond « VOUS AVEZ GAGNE! ».

Tous les autres numéros de port répondent « VOUS AVEZ PERDU! ».

Le serveur répond « VOUS AVEZ FAIT UNE ERREUR. » si la lettre envoyée par le client est différente de ?.

L'objectif de cet exercice est de réaliser un programme capable de déterminer le port X correspondant à « VOUS AVEZ GAGNE! » et de l'afficher.

#### 3.2 Préparation de votre machine.

Pour pouvoir tester votre programme, vous allez installer sur votre machine ce serveur TCP.

Pour cela, télécharger le fichier **port-scanning-tcp.jar** disponible sur le serveur d'examen.

Lancer ce serveur UDP avec la commande

```
/opt/java/bin/java -jar port-scanning-tcp.jar
```

### ***3.3 Essai manuel.***

Faites un essai avec netcat et le serveur fourni.

Vous vous connecterez en TCP sur le port 30000, vous enverrez la lettre ? et vous visualiserez la réponse du serveur.

Notez les commandes que vous utilisez dans un fichier netcat.txt, que vous mettrez dans votre projet **examen\_exo2\_[votrenom]**.

### ***3.4 Ecriture du programme***

Ecrivez maintenant une classe ClientTCPScanner. Si on exécute le main de cette classe, on obtient l'affichage suivant :

```
Début de la recherche ...
```

```
Le port d'écoute qui répond VOUS AVEZ GAGNE! est 31500
```

```
Fin du programme
```

**Cet affichage devra se faire dans un délai inférieur à 1 minute.**

Dans un commentaire tout en haut de cette classe, vous indiquerez la valeur de X que vous avez trouvé.

L'adresse IP de la machine scannée (ici "127.0.0.1") sera écrite en dur dans votre programme.