



ESISAR

NE441 - Utilisation des couches transport

Examen de TP

Table des matières

1	Consignes générales.....	1
2	Exercice 2 : Réalisation d'un client TCP simple (7 points).....	2
2.1	Présentation générale du problème.....	2
2.2	Préparation de votre machine.....	3
2.3	Ecriture du programme PasswordTCPClient.....	3
3	Exercice 2 : Réalisation d'un client TCP complexe (13 points).....	3
3.1	Présentation générale du problème.....	4
3.2	Préparation de votre machine.....	5
3.3	Ecriture du programme PasswordTCPClient.....	5
4	Instructions pour le rendu de vos travaux.....	6

1 Consignes générales

La durée de cet examen de TP est de **2H00**.

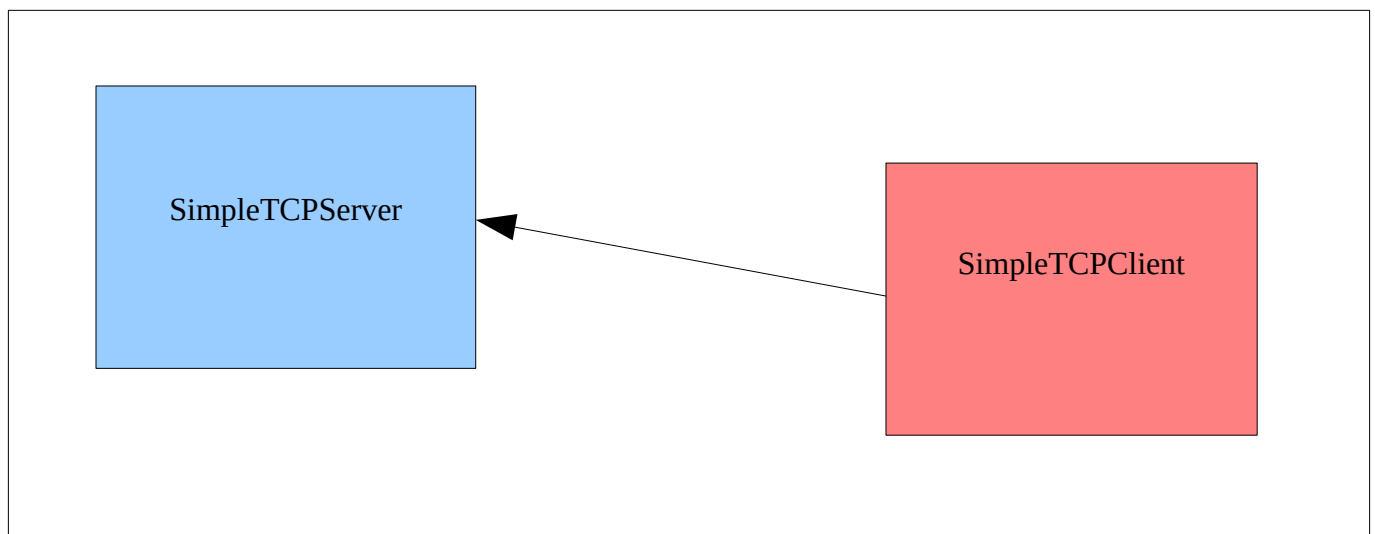
Attention : vous devez absolument respecter les consignes suivantes :

- 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 en fin de document.

2 Exercice 2 : Réalisation d'un client TCP simple (7 points)

2.1 Présentation générale du problème.

Vous allez devoir réaliser un client TCP pour trouver des mots de passe cachés dans le flux de données envoyé par serveur TCP.



L'application SimpleTCPServer est un serveur TCP, qui écoute sur le port 7000. Ce serveur fonctionne ainsi :

- le serveur attend les demandes de connexion
- quand un client se connecte, le serveur répond (par exemple) la trame suivante :

```
top**secret*
```

- cette trame contient un mot de passe, qui ici est `topsecret`
- pour décoder cette trame le principe est le suivant
 - le début de la trame contient le début du mot de passe (formé uniquement de caractères en minuscules), ici `top`
 - la trame contient ensuite deux étoiles : `**`
 - la trame contient ensuite la fin du mot de passe (formé uniquement de caractères en minuscules), puis le caractère `*` (ici `secret*`)
- le client doit alors répondre en envoyant le mot de passe terminé par un point virgule, par exemple

```
tressecret;
```

- le serveur vous envoie de nouveau une trame avec un format identique à la précédente
- vous devez décoder la trame suivante , et répondre avec le mot de passe suivant

A noter :

- le serveur accepte une seule connexion simultanée
- le serveur peut envoyer plusieurs trames à la suite : le serveur n'attend pas toujours d'avoir reçu le mot de passe pour envoyer la trame suivante.
- le serveur affiche votre score sur la console
- si le client met plus de 20 secondes pour envoyer le mot de passe ou si le mot de passe est erroné, le serveur ferme la connexion
- la longueur des mots de passe est au maximum 150 caractères

2.2 Préparation de votre machine.

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

Pour cela, télécharger le fichier **simple-password-tcp.jar** joint à ce document.

Lancer ce serveur avec la commande

```
java -jar simple-password-tcp.jar
```

2.3 Ecriture du programme PasswordTCPClient

Ecrivez maintenant une classe **SimpleTCPClient**, qui trouve les **10 premiers** mots de passe successifs.

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

```
Début de la connexion au serveur 1
```

```
Mot de passe 1 : tressecret
```

```
Mot de passe 2 : ultrasecret
```

```
Mot de passe 3 : topsecure
```

```
...
```

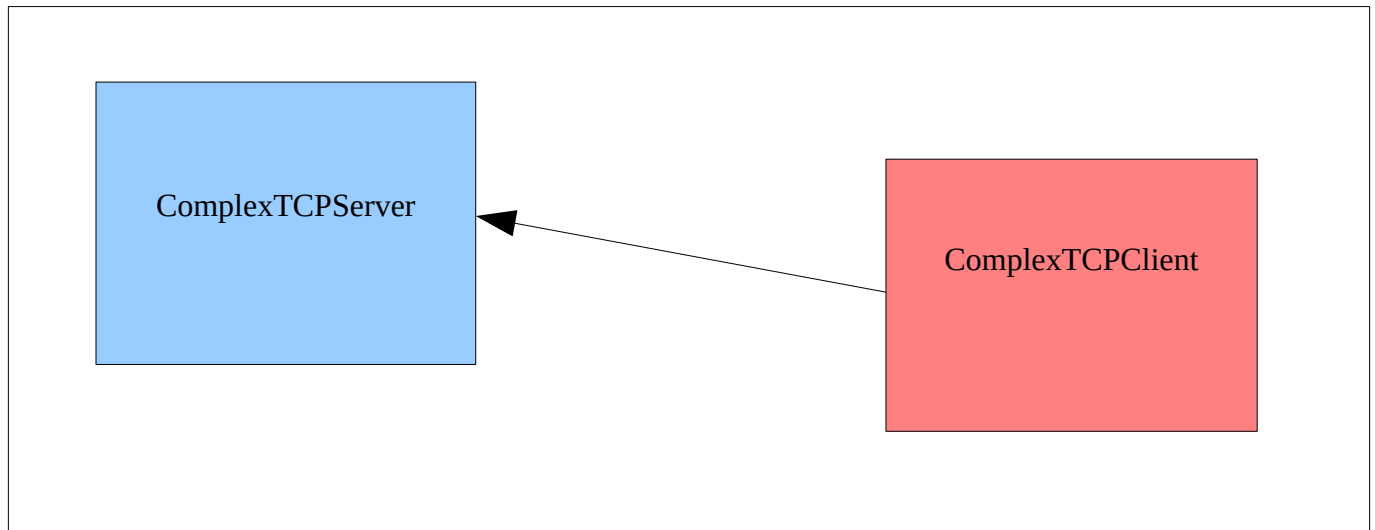
```
Mot de passe 10: aaa
```

```
Fin du programme
```

3 Exercice 2 : Réalisation d'un client TCP complexe (13 points)

3.1 Présentation générale du problème.

Cet exercice est très similaire au précédent, mais cette fois ci les mots de passe sont mieux cachés.



L'application ComplexTCPServer est un serveur TCP, qui écoute sur le port 7500. Ce serveur fonctionne ainsi :

- le serveur attend les demandes de connexion
- quand un client se connecte, le serveur répond (par exemple) la trame suivante :

```
tres*12*abczahjdnydosecret*
```

- cette trame contient un mot de passe, qui ici est `tressecret`
- pour décoder cette trame le principe est le suivant
 - le début de la trame contient le début du mot de passe (formé uniquement de caractères en minuscules), ici `tres`
 - la trame contient ensuite un nombre X compris entre deux étoiles : ici `*12*`
 - la trame contient ensuite X octets aléatoires : ici `abczahjdnydo`. Ces octets peuvent prendre toutes les valeurs possibles.
 - la trame contient ensuite la fin du mot de passe (formé uniquement de caractères en minuscules), puis le caractère `*` (ici `secret*`)
- le client doit alors répondre en envoyant le mot de passe terminé par un point virgule, par exemple

```
tressecret;
```

- le serveur vous envoie de nouveau une trame avec un format identique à la précédente
- vous devez décoder la trame suivante , et répondre avec le mot de passe suivant

A noter :

- le serveur accepte une seule connexion simultanée
- le serveur peut envoyer plusieurs trames à la suite : le serveur n'attend pas toujours d'avoir reçu le mot de passe pour envoyer la trame suivante.
- le serveur affiche votre score sur la console
- le nombre X est un nombre compris entre 1 et 2 000 000 000.
- si le client met plus de 20 secondes pour envoyer le mot de passe ou si le mot de passe est erroné, le serveur ferme la connexion
- la longueur des mots de passe est au maximum 150 caractères

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 **complex-password-tcp.jar** joint à ce document.

Lancer ce serveur avec la commande

```
java -jar complex-password-tcp.jar
```

3.3 Ecriture du programme PasswordTCPClient

Ecrivez maintenant une classe ComplexTCPClient, qui trouve les **16 premiers** mots de passe successifs.

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

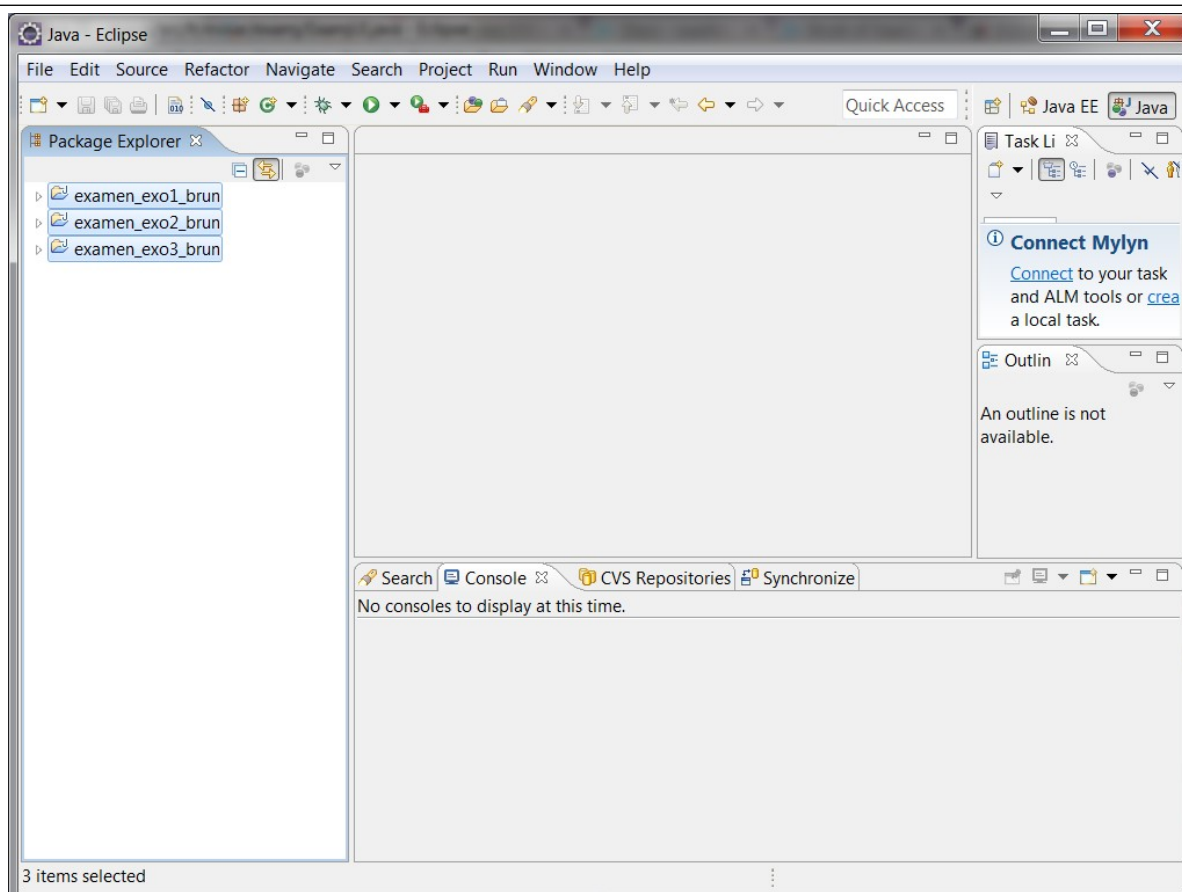
```
Début de la connexion au serveur 1
Mot de passe 1 : tressecret
Mot de passe 2 : ultrasecret
Mot de passe 3 : topsecure
...
Mot de passe 16 : topsecure
Fin du programme
```

TRES IMPORTANT : la consommation mémoire de votre programme doit être inférieure à 50 Mo. Vous ne devez pas stocker tout le flux inutilement. L'exercice ne sera pas noté si vous ne respectez pas cette consigne.

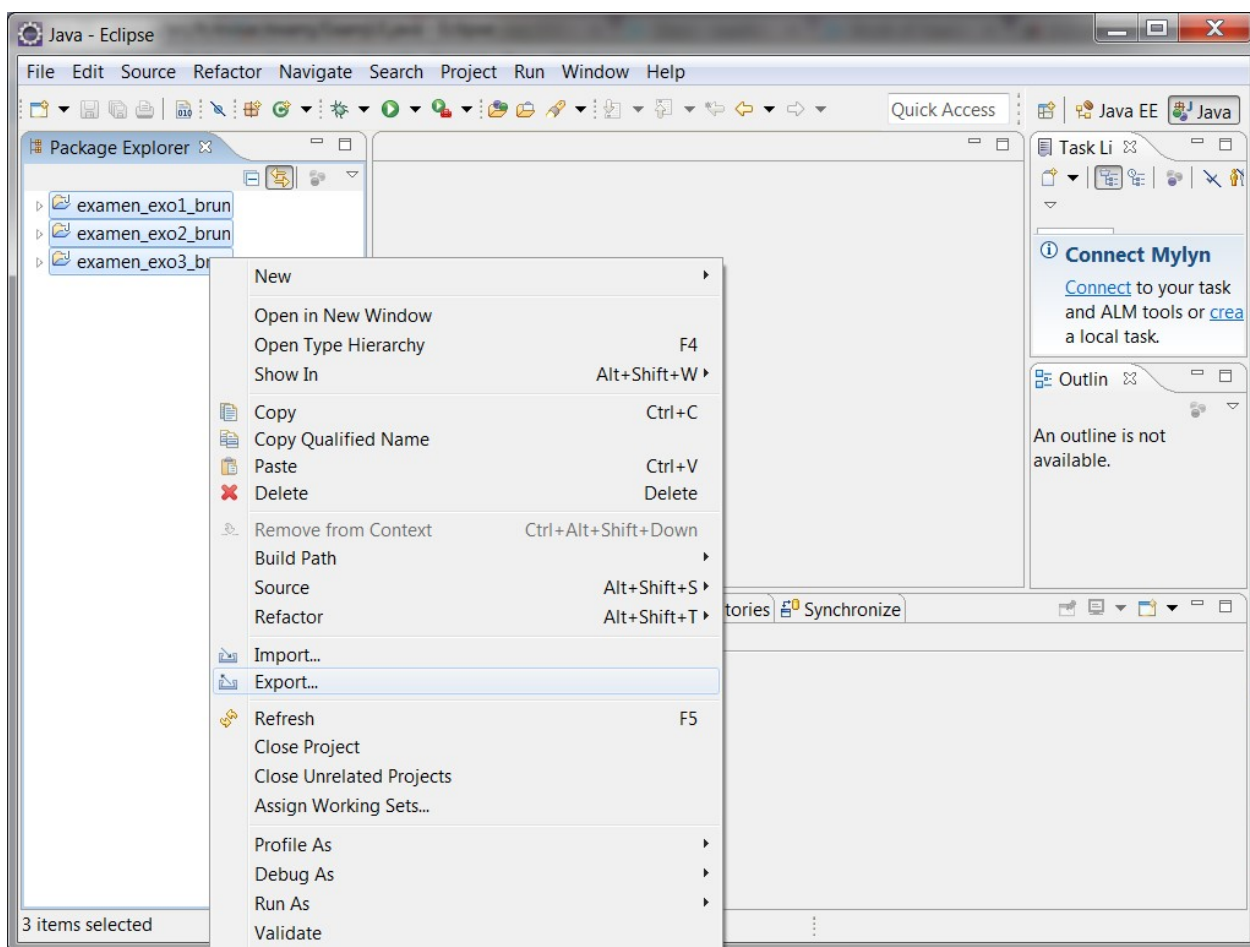
4 Instructions pour le rendu de vos travaux

Pour rendre vos travaux, vous devez d'abord faire un seul zip avec l'ensemble de vos travaux.

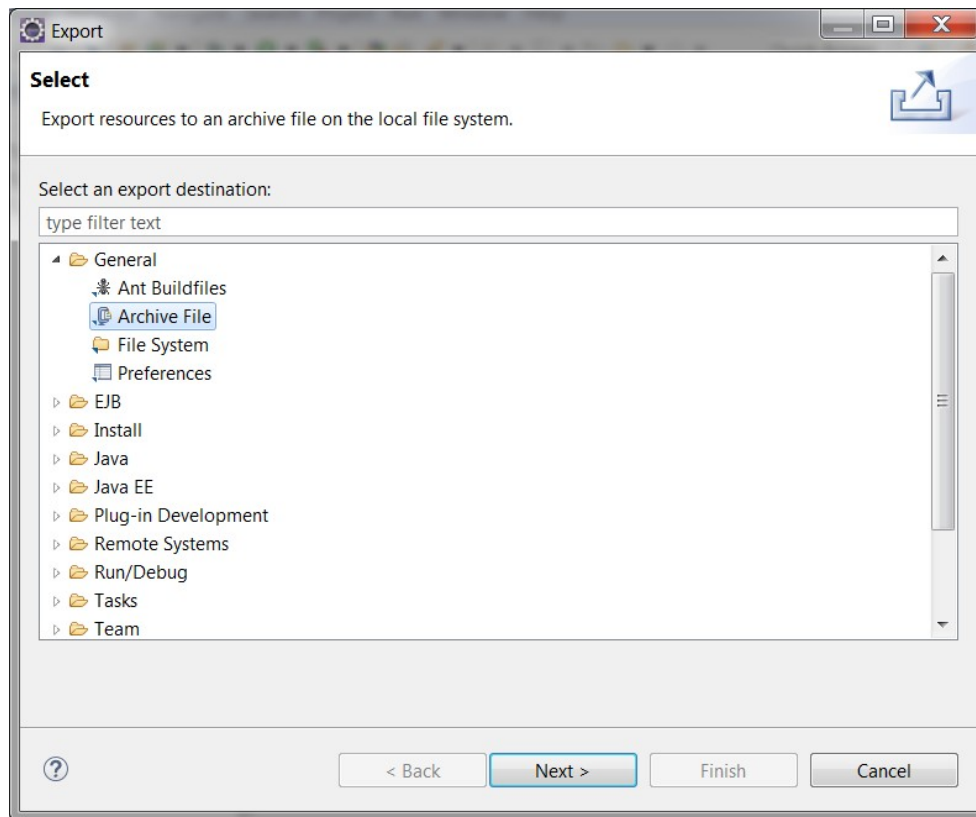
Pour cela, allez dans Eclipse, sélectionnez tous vos projets :



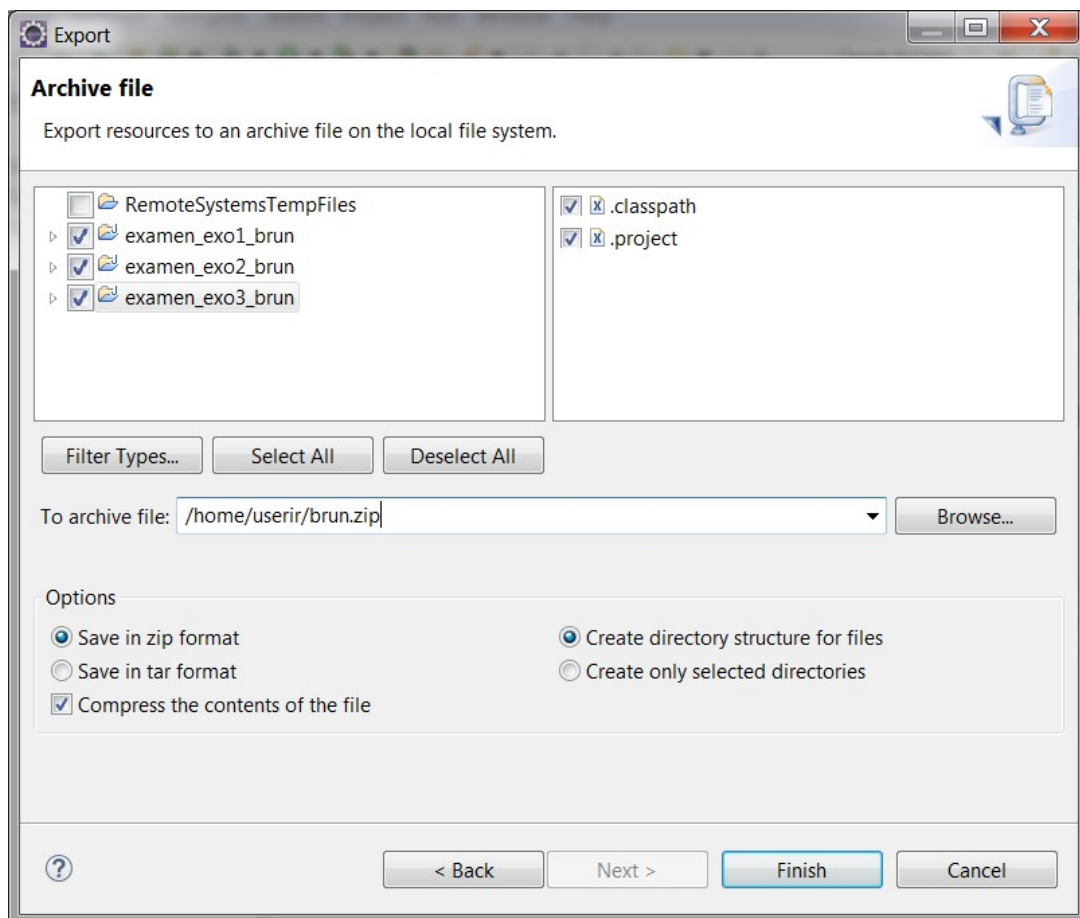
Ensuite vous faites clic droit, puis Export



Ensuite vous choisissez General / Archive File



Ensuite vous sauvegardez vos projets dans un fichier [votre nom].zip (voir ci dessous).



Vous cliquez sur Finish. Vous avez donc maintenant sauvegardé tous vos travaux dans un fichier [votre nom].zip qui se trouve dans /home/userir/.

Vous devez ensuite envoyer ce fichier par Chamilo.