

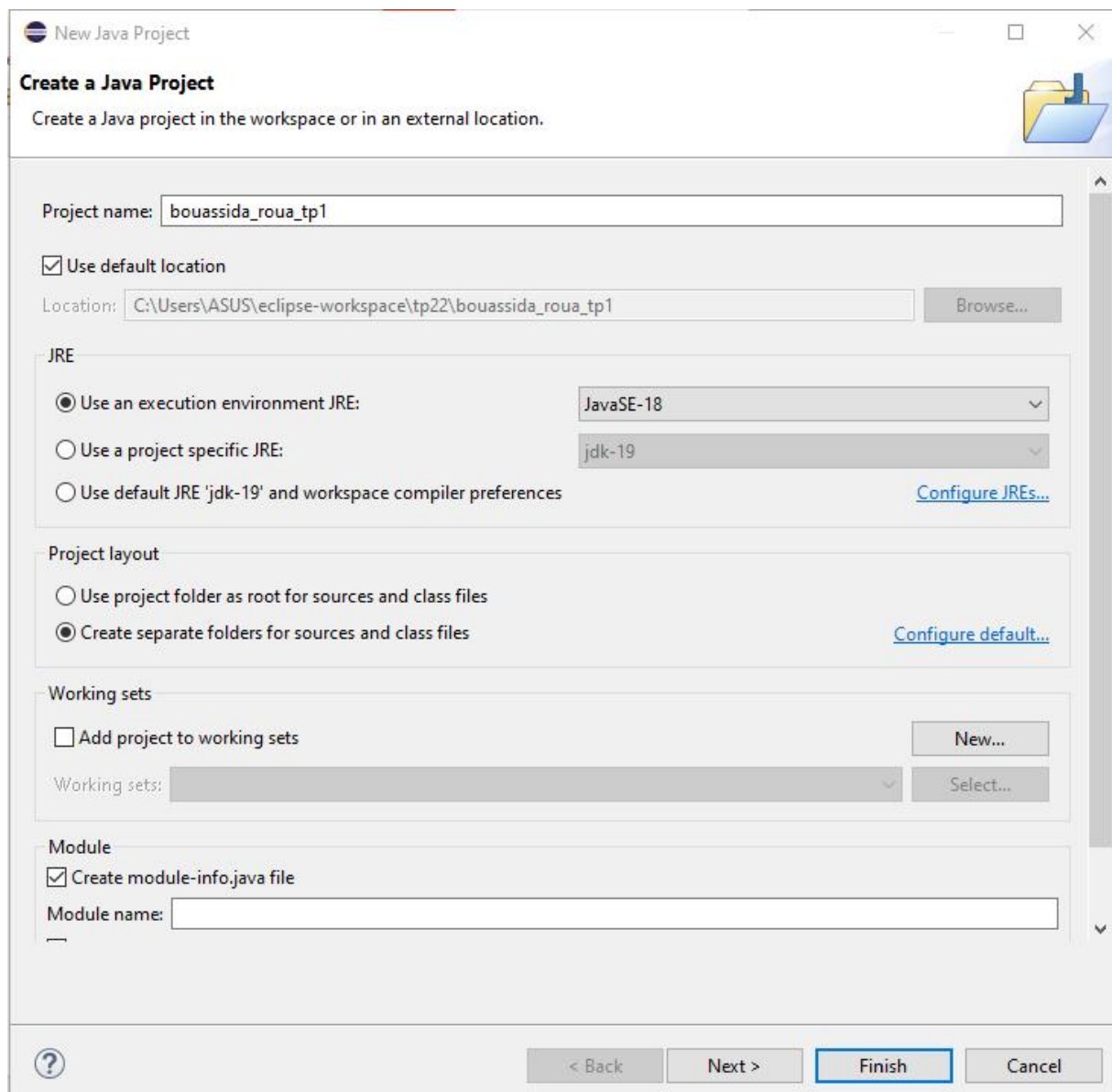
TP1 : Les sockets en mode connecté TCP

Activité 1-1 : Première connexion Client/Serveur

Description de la 2ème phase : Création du serveur

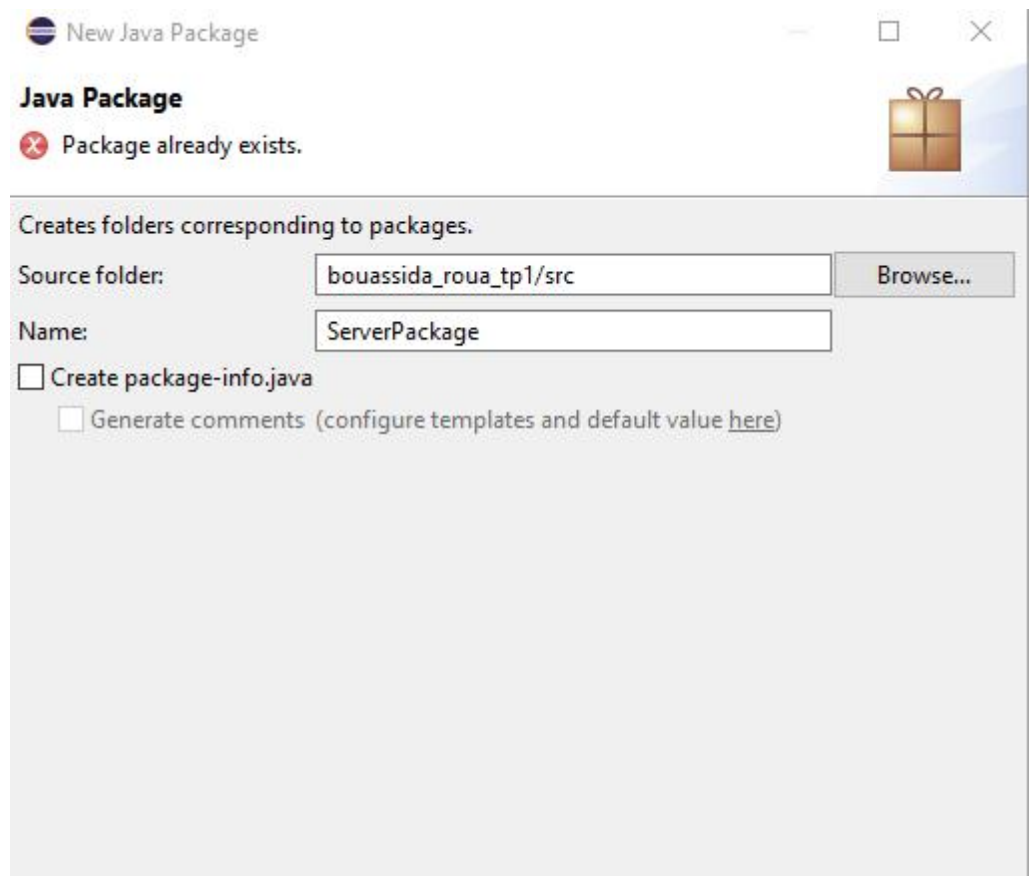
Dans cette phase, vous êtes demandé(e) de créer l'application du serveur.

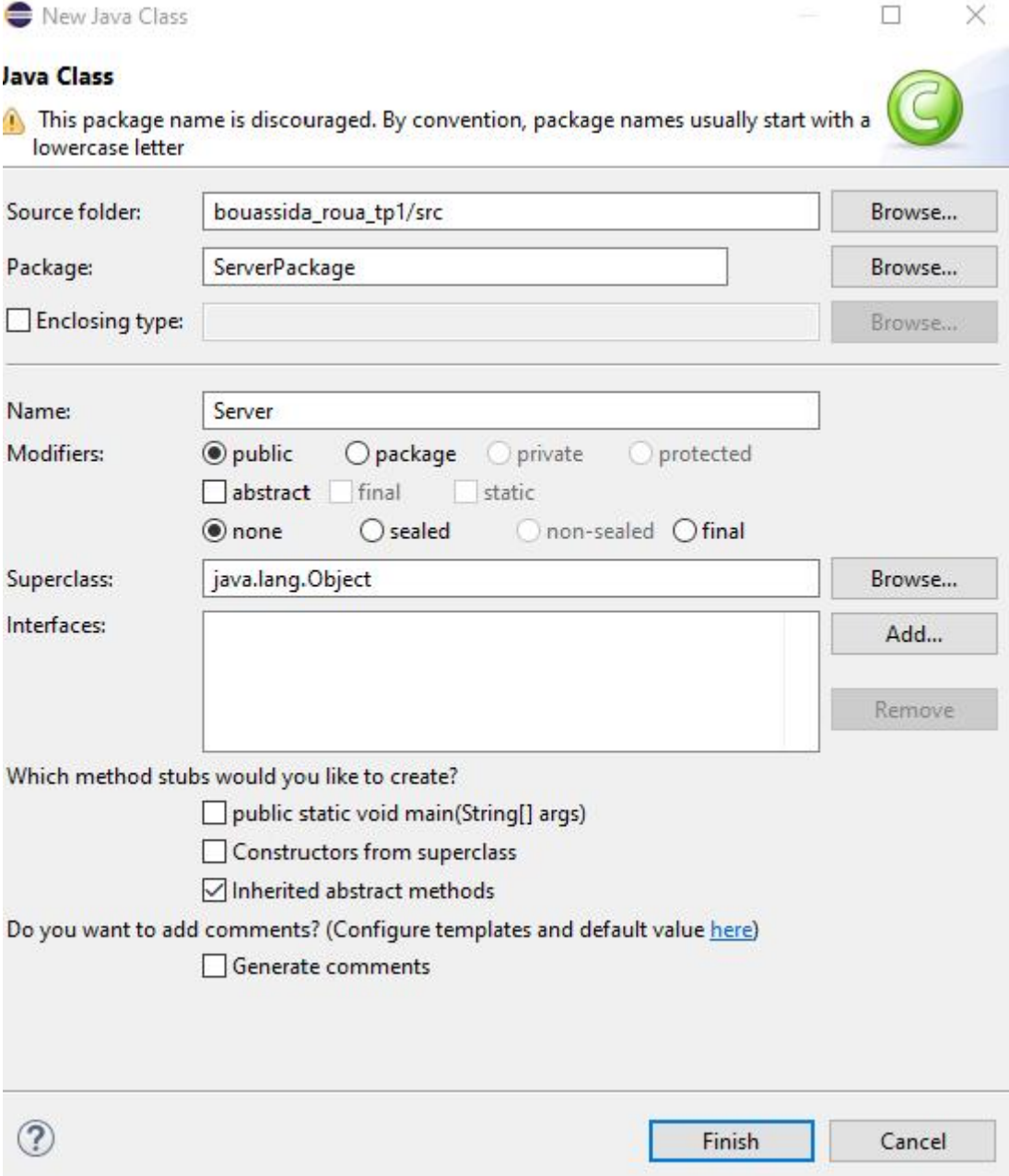
1. Démarrez l'IDE Eclipse (ou IntelliJ) en vous référant à votre répertoire du TP1.
2. Créez un Projet Java en acceptant les réglages proposés par défaut sous le nom : Nom_Prénom_TP1.



The screenshot shows the 'New Java Project' dialog box in Eclipse. The 'Project name' field is filled with 'bouassida_roua_tp1'. The 'Use default location' checkbox is checked, and the 'Location' field shows the default workspace path. Under the 'JRE' section, 'Use an execution environment JRE:' is selected, with 'JavaSE-18' chosen from the dropdown. The 'Project layout' section has 'Create separate folders for sources and class files' selected. In the 'Working sets' section, 'Add project to working sets' is unchecked. The 'Module' section has 'Create module-info.java file' checked, and the 'Module name' field is empty. At the bottom, there are buttons for '< Back', 'Next >', 'Finish' (highlighted with a blue border), and 'Cancel'.

3. Écrivez le code java correspondant à l'application Serveur
- a) Créez un package nommé : "SeverPackage" sous le répertoire "src", dans lequel, vous créez une classe qui porte le nom : "Server" et choisissez la génération automatique de la méthode main().



 The image shows a 'New Java Class' dialog box in an IDE. At the top, there's a title bar with a question mark icon and window controls. Below the title, a warning icon and text state: 'This package name is discouraged. By convention, package names usually start with a lowercase letter'. A green circular icon with a 'C' is on the right. The dialog is divided into sections: 'Source folder' with a text field 'bouassida_roua_tp1/src' and a 'Browse...' button; 'Package' with a text field 'ServerPackage' and a 'Browse...' button; 'Enclosing type' with an unchecked checkbox and a 'Browse...' button. The 'Name' section has a text field 'Server'. The 'Modifiers' section has radio buttons for 'public' (selected), 'package', 'private', and 'protected', and checkboxes for 'abstract', 'final', and 'static'. Below these are radio buttons for 'none' (selected), 'sealed', 'non-sealed', and 'final'. The 'Superclass' section has a text field 'java.lang.Object' and a 'Browse...' button. The 'Interfaces' section has an empty list box, an 'Add...' button, and a 'Remove' button. The 'Which method stubs would you like to create?' section has checkboxes for 'public static void main(String[] args)', 'Constructors from superclass', and 'Inherited abstract methods' (checked). The 'Do you want to add comments? (Configure templates and default value [here](#))' section has a checkbox for 'Generate comments'. At the bottom, there's a question mark icon, a 'Finish' button, and a 'Cancel' button.

b) Dans le corps de la méthode main de la classe Server, tapez le code suivant :

// La première étape :

//à compléter (TODO)

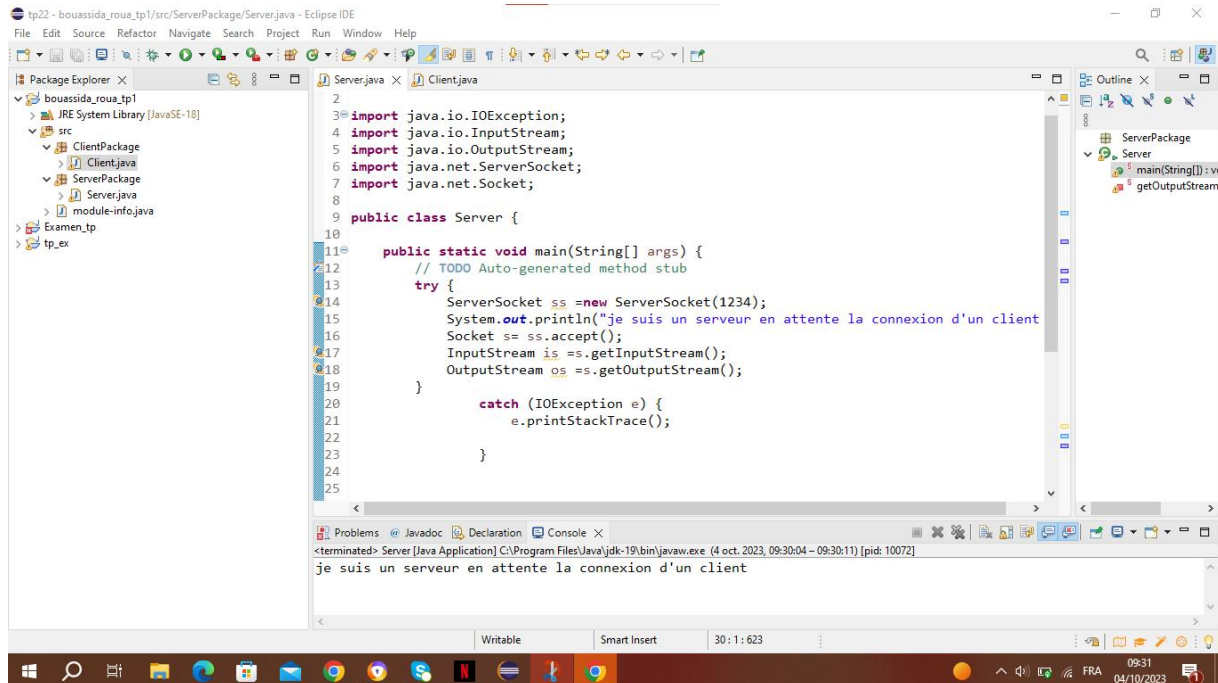
S.o.p("Je suis un serveur en attente la connexion d'un client ");

// La deuxième étape : S.o.p("un client est connecté");

// La dernière étape :Fermer socket

// à compléter (TODO)

- c) Essayez de compléter les parties du code où s'est marqué à compléter (TODO) dans le code tapé.
N.B. Importez les paquetages nécessaires et ajoutez les exceptions appropriées.
- d) Lancez le serveur.
- e) Interprétez le résultat affiché dans la console



Description de la 3ème phase : Création du client

Dans cette phase, vous êtes demandé(e) de créer l'application du client.

1. Écrivez le code java correspondant à l'application Client
 - a) Créez un package nommé : "ClientPackage" sous le répertoire "src", dans lequel, vous créez une classe qui porte le nom : "Client" et choisissez la génération automatique de la méthode main().

New Java Package

Java Package

⚠ Discouraged package name. By convention, package names usually start with a lowercase letter

Creates folders corresponding to packages.

Source folder:


Name:

☐ Create package-info.java

☐ Generate comments (configure templates and default value [here](#))

New Java Class

Java Class

 Type already exists.

Source folder:

Package:

☐ Enclosing type:

Name:

Modifiers: ☒ public ☐ package ☐ private ☐ protected
☐ abstract ☐ final ☐ static
☒ none ☐ sealed ☐ non-sealed ☐ final


Superclass:

Interfaces:

Which method stubs would you like to create?

☒ public static void main(String[] args)
☐ Constructors from superclass
☒ Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))
☐ Generate comments



- b) Dans le corps de la méthode main, copiez le code suivant :


```
// La première étape :
//à compléter (TODO)
s.o.p("Je suis un client pas encore connecté...");
//à compléter (TODO)
s.o.p("je suis un client connecté");
// La dernière étape : Fermer socket
//à compléter (TODO)
```
- c) Idem pour 3.c). de la 2ème phase.
- d) Lancez le client.
- e) Interprétez le résultat affiché dans la console.
- f) Changez le nom du localhost par une adresse IP

