

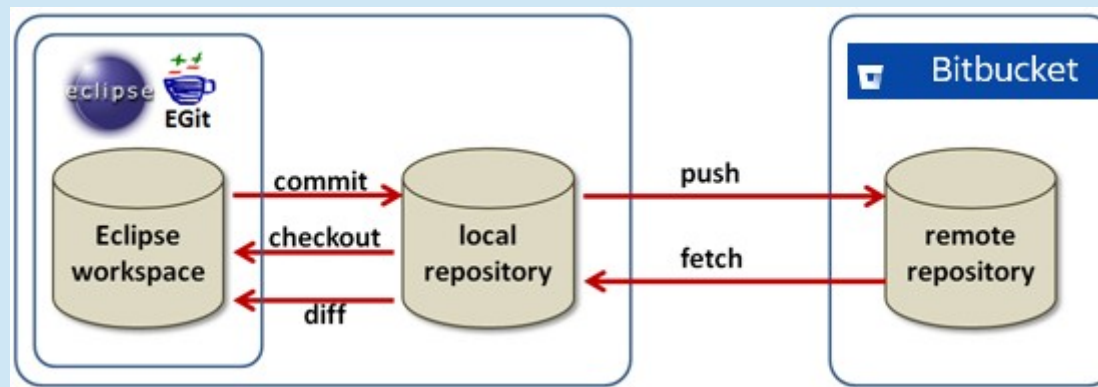
# LP WCE 17/18 – Cours GIT

A la fin de ce cours :

- vous saurez comment travailler de façon collaborative avec git, en utilisant Eclipse et bitbucket

# Configuration

- Eclipse Oxygen (juin 2017)
- Java 8
- <https://bitbucket.org>



# Création du dépôt

- **Pierre** crée le dépôt privé *CoursCollaboration* sous bitbucket et invite **Stéphane** avec le rôle *Admin*

The screenshot shows the Bitbucket interface for the repository 'CoursCollaboration'. The left sidebar contains navigation links: Overview, Source, Commits, Branches, Pull requests, Pipelines, Downloads, Boards, and Settings (highlighted with a green arrow). The main content area is titled 'Settings' and includes a breadcrumb 'Pierre Laroche / CoursCollaboration'. Under the 'GENERAL' section, 'User and group access' is selected (highlighted with a green arrow). The 'User and group access' section contains a description and a 'Users' table. The 'Users' table lists two users: 'Pierre Laroche' (owner) and 'Stéphane JOYEUX' (ADMIN, highlighted with a green arrow). The 'Groups' section is also visible at the bottom.

**Settings**

Pierre Laroche / CoursCollaboration

**GENERAL**

- Repository details
- User and group access**
- Access keys
- Username aliases

**WORKFLOW**

- Branch permissions
- Default reviewers
- Webhooks
- Links

**FEATURES**

**User and group access**

Here's where you grant users and groups access to this repository. For a list of all users with access to any of your private repositories, see which users count towards your bill on the [Users on plan](#) page. [Learn more](#)

**Users**

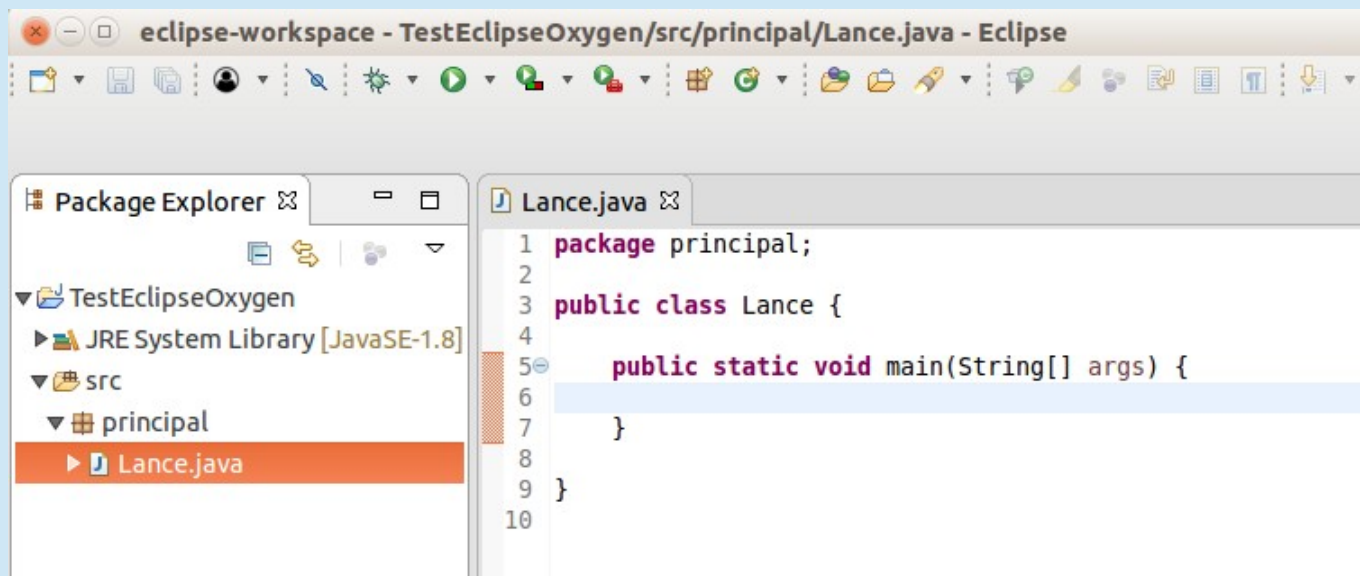
Username or email address	Role		Add
Pierre Laroche	owner		
Stéphane JOYEUX	<span>READ</span> <span>WRITE</span> <b>ADMIN</b>		

**Groups**

Select a group	Role		Add
	Read		

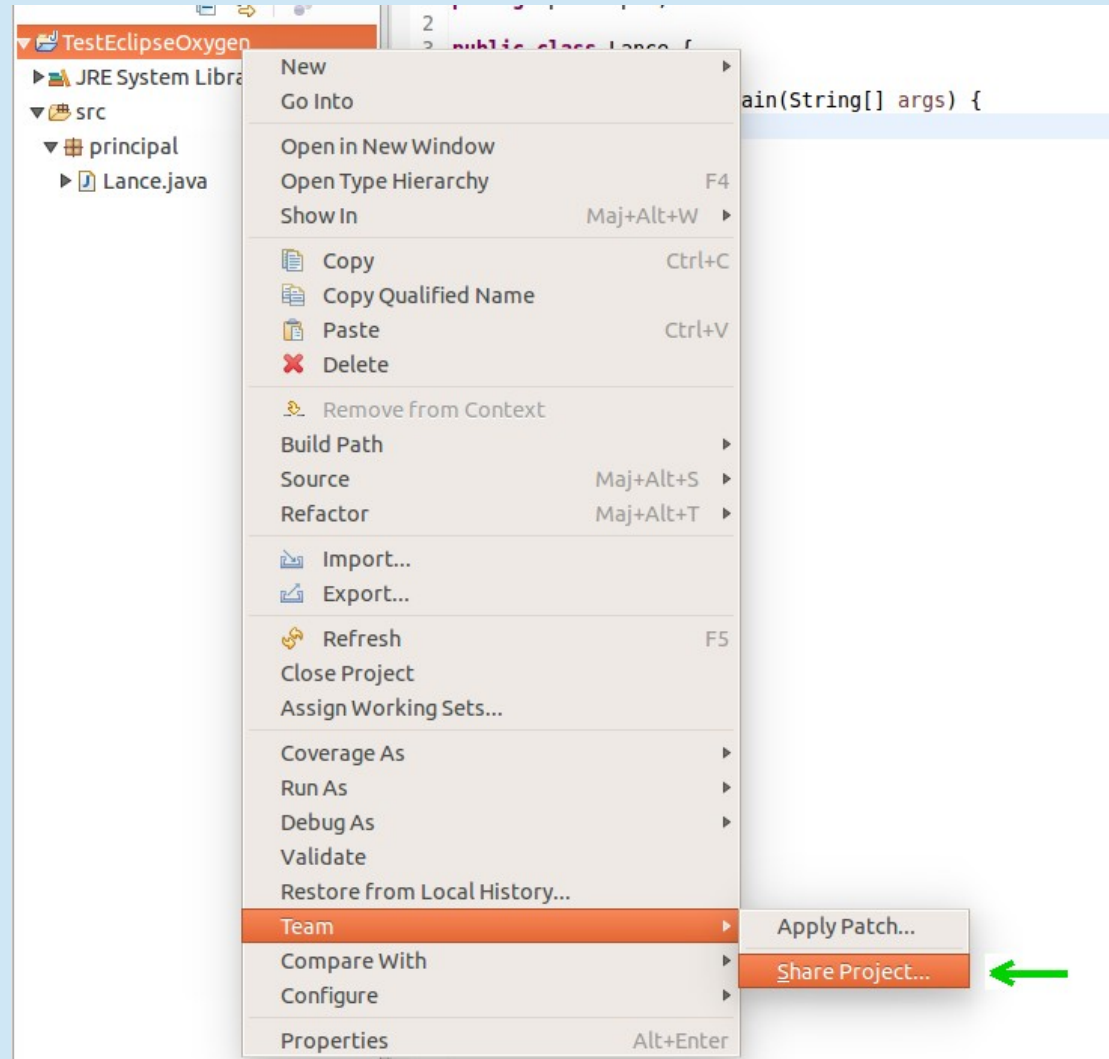
# Création du projet sous Eclipse

- **Pierre** crée le projet de façon classique : menu File, New, Java Project

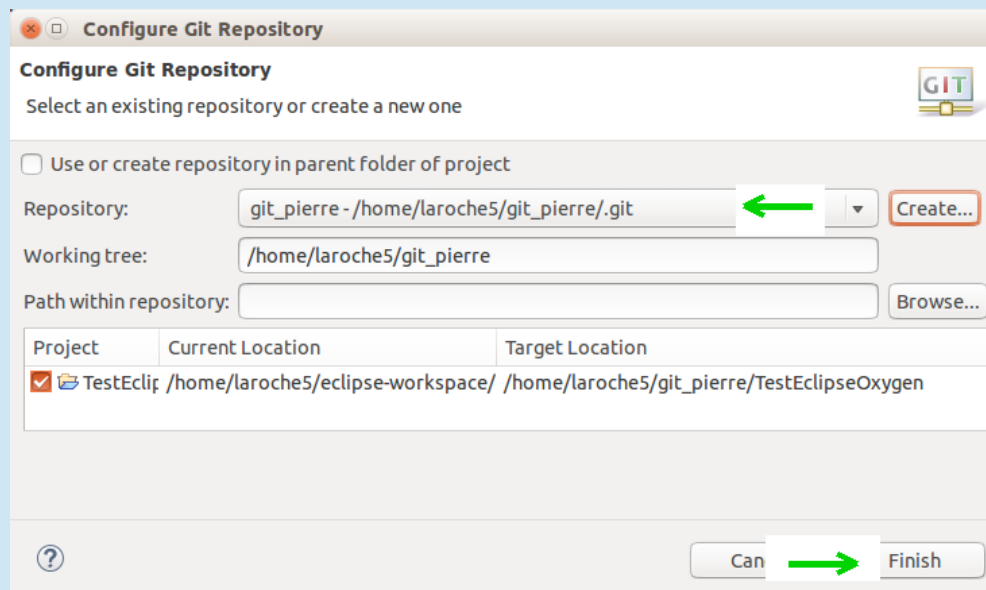
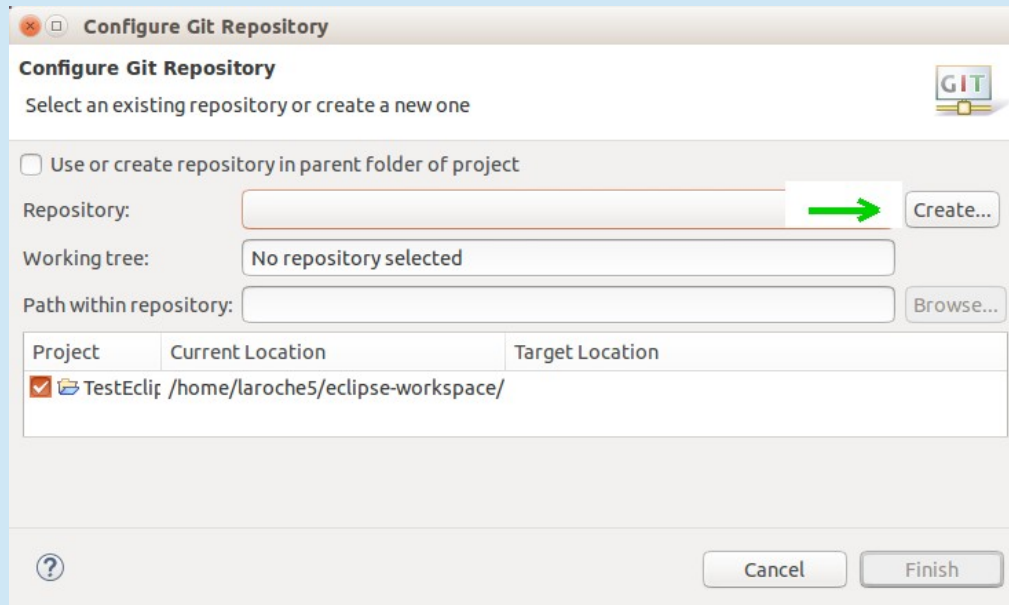


# Lien du projet avec un dépôt local

- Clic bouton droit sur le projet, Team, Share Project

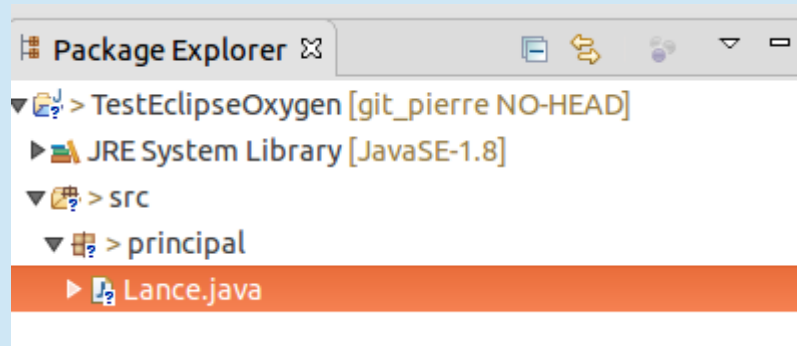


# Création du dépôt local



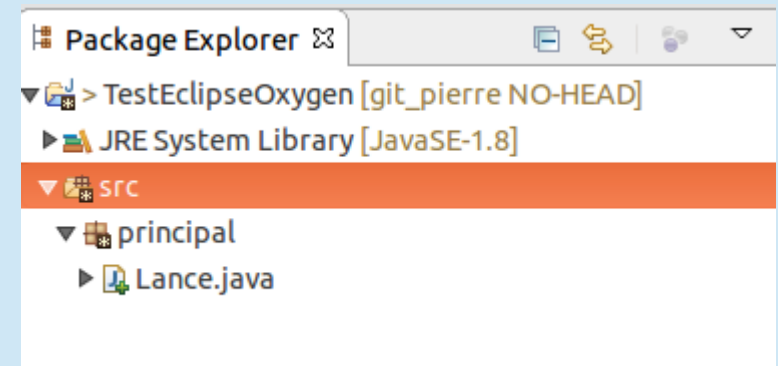
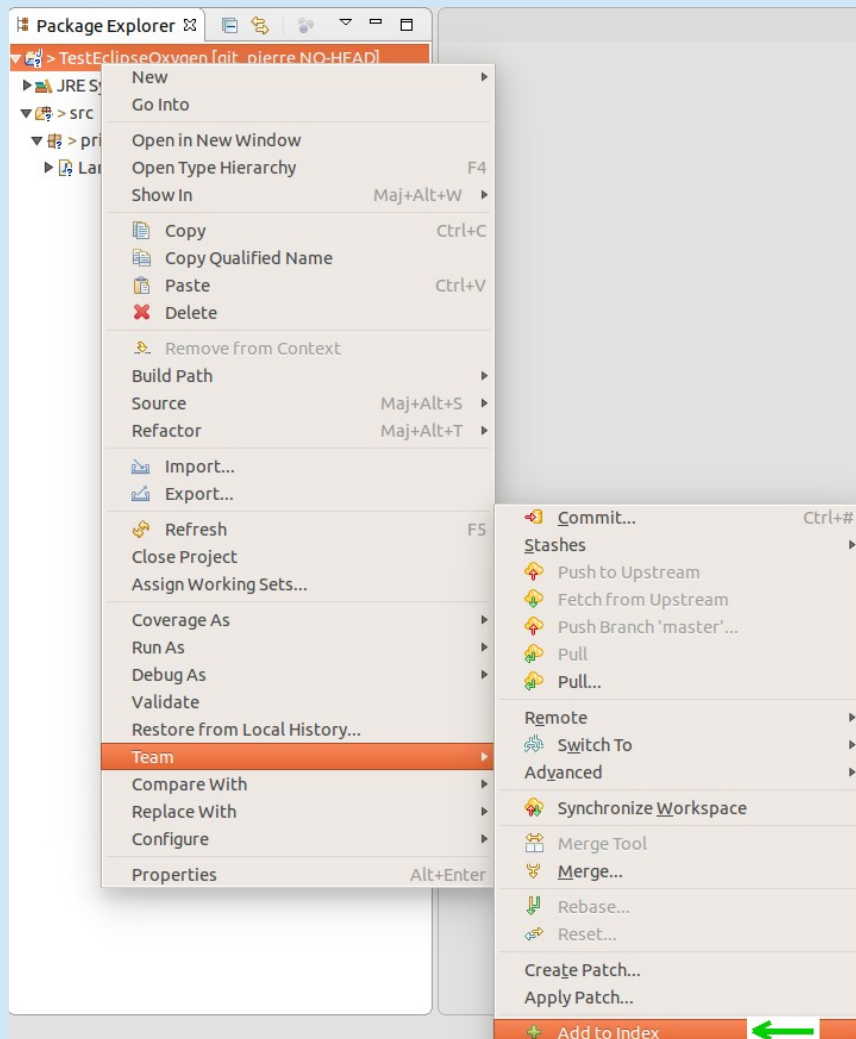
# Le projet passe en mode collaboratif

- Mention [nom du dépôt NO-HEAD]  
(pas encore de commit donc pas de branche)
- ‘ ? ’ sur les éléments



# Désigner ce qui doit être mis dans le dépôt

- Clic droit sur le projet, Team, Add to Index



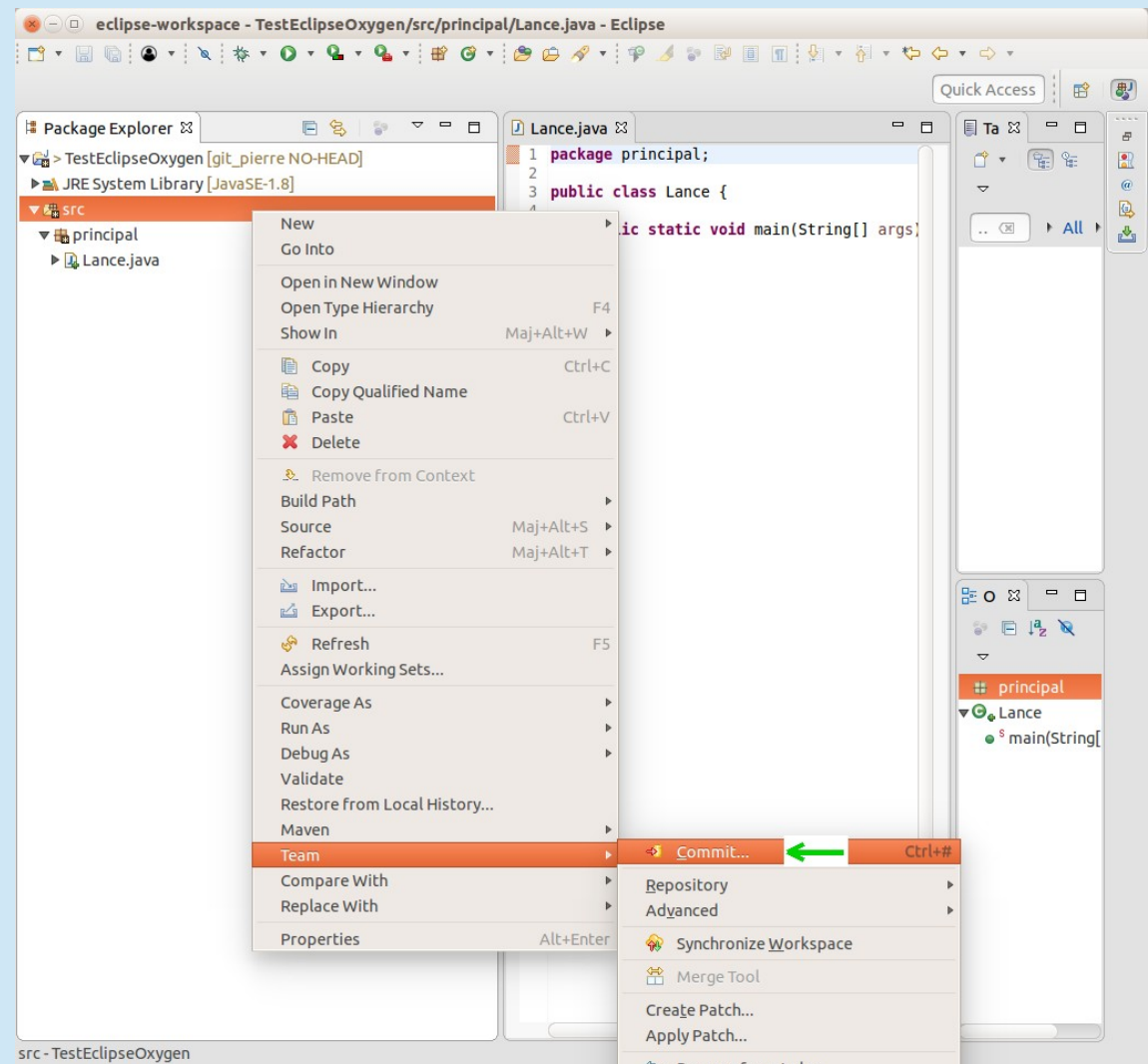
‘\*’ sur les éléments qui contiennent des nouveautés

‘+’ sur les éléments ajoutés



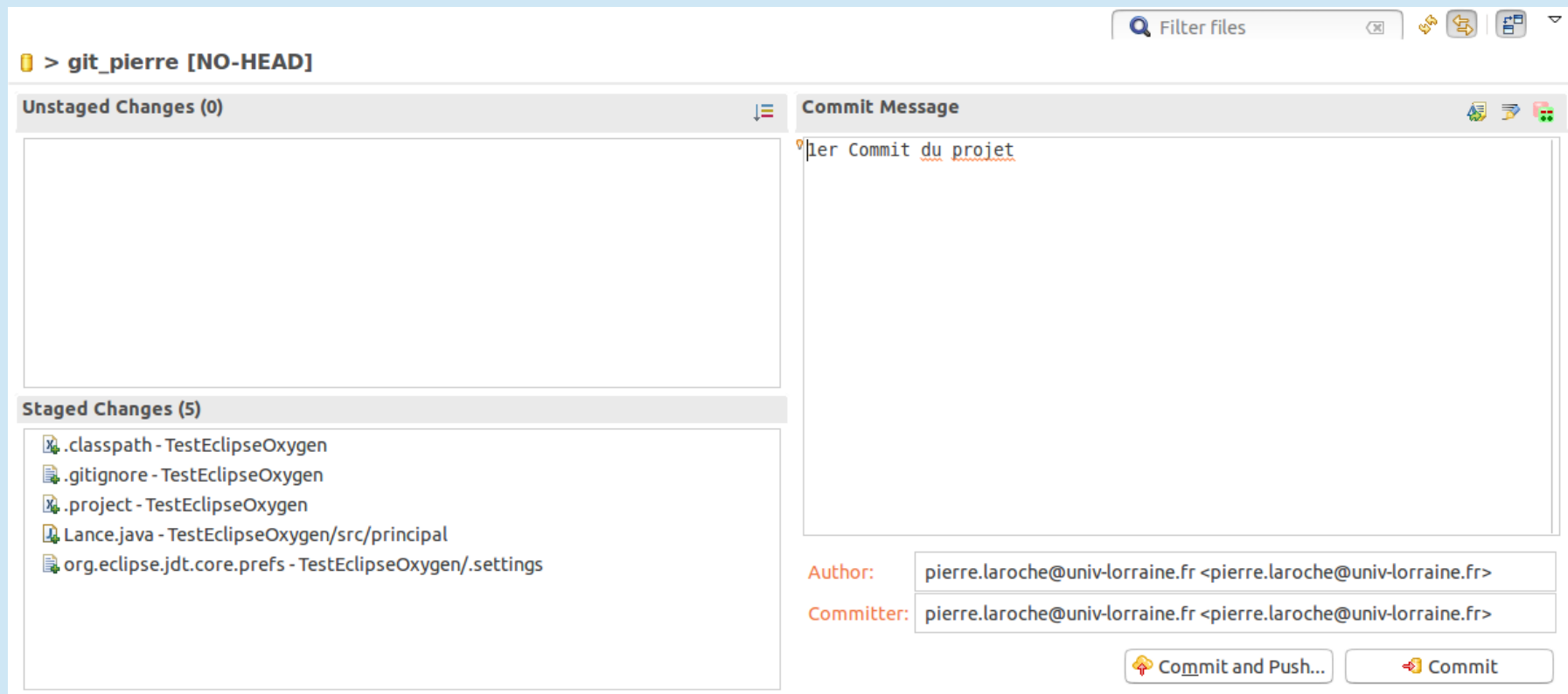
# Envoi sur les dépôts locaux et distants (1)

- Clic bouton droit dans le projet, Team, Commit...



# Envoi sur les dépôts locaux et distants (2)

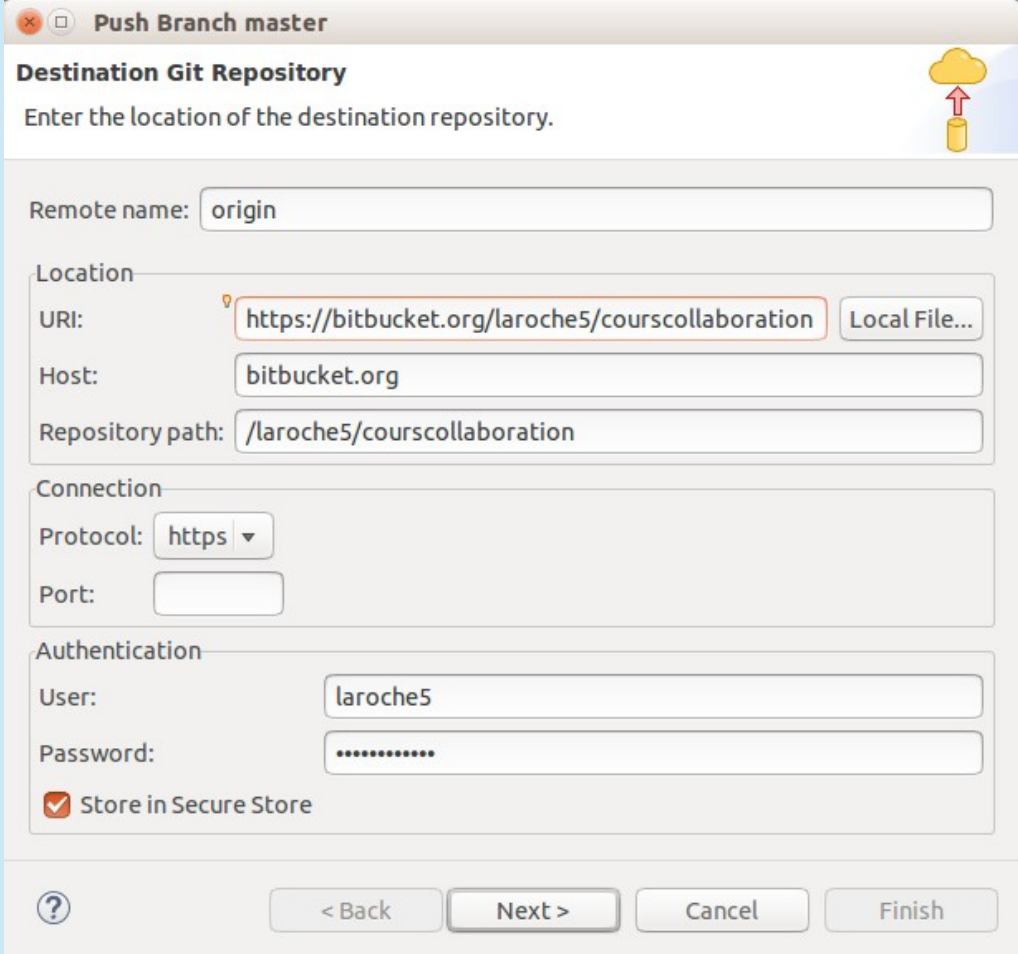
- Unstaged changes : ce qui ne sera pas copié
- Commit message : obligatoire, permet de documenter le dépôt



10/29

# Envoi sur les dépôts locaux et distants (3)

- Après clic sur Commit and Push...
- Saisie des données d'accès au dépôt distant



**Push Branch master**

**Destination Git Repository**  
Enter the location of the destination repository.

Remote name:

Location

URI:

Host:

Repository path:

Connection

Protocol:

Port:

Authentication

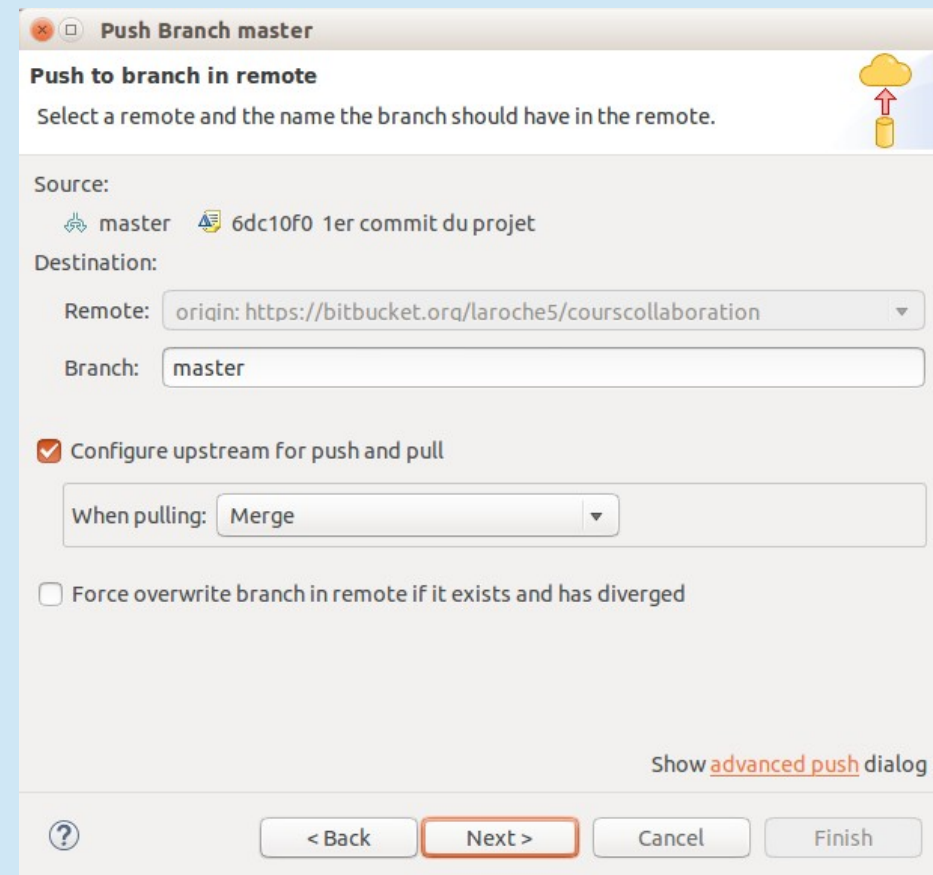
User:

Password:

☒ Store in Secure Store

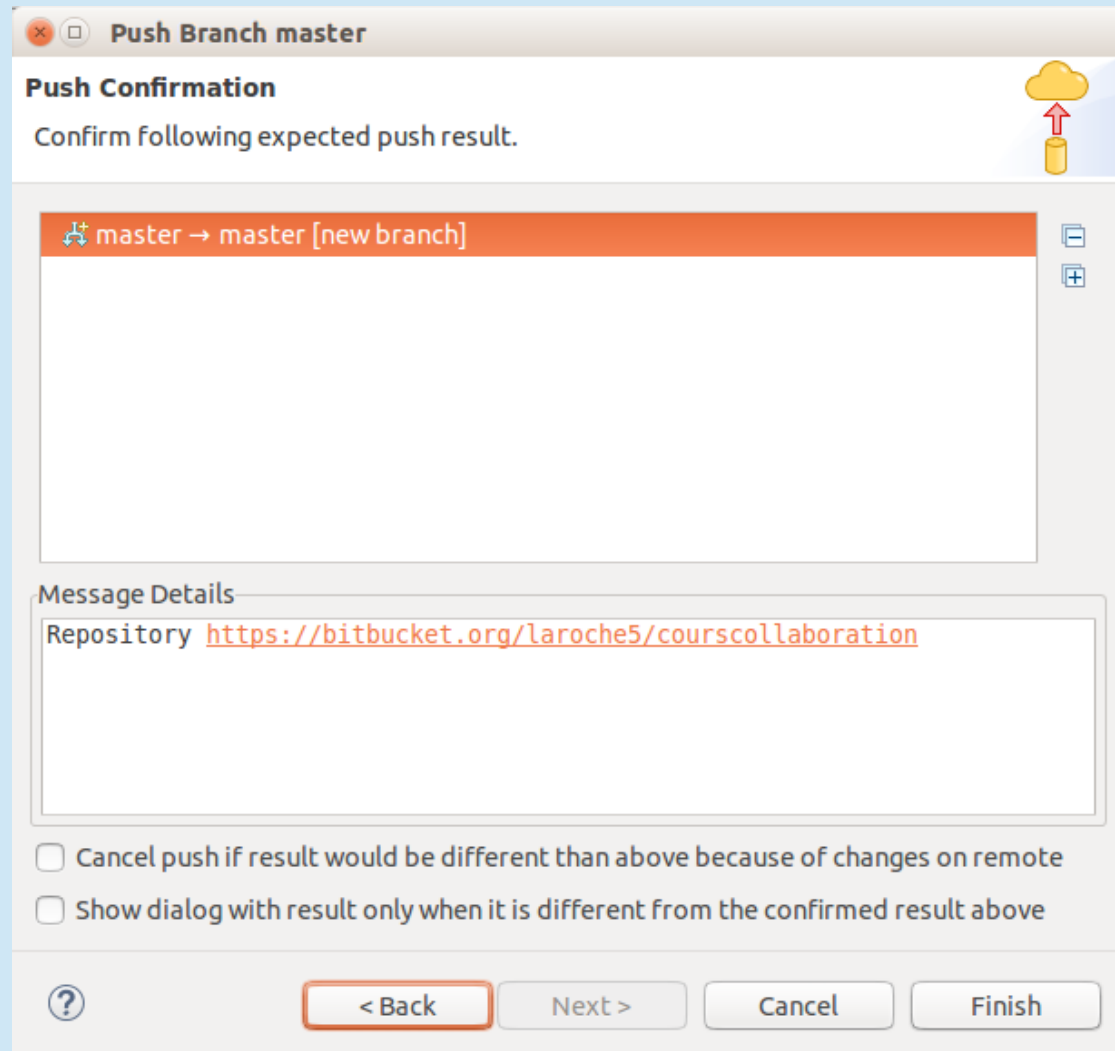
# Envoi sur les dépôts locaux et distants (4)

- On pousse vers la branche « master »
- (nous verrons plus tard ce qu'est une branche)



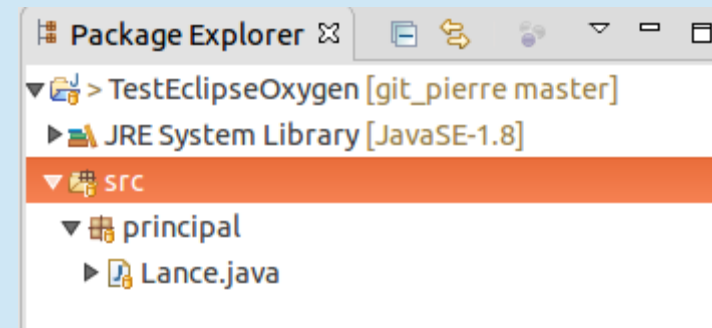
# Envoi sur les dépôts locaux et distants (5)

- Demande de confirmation



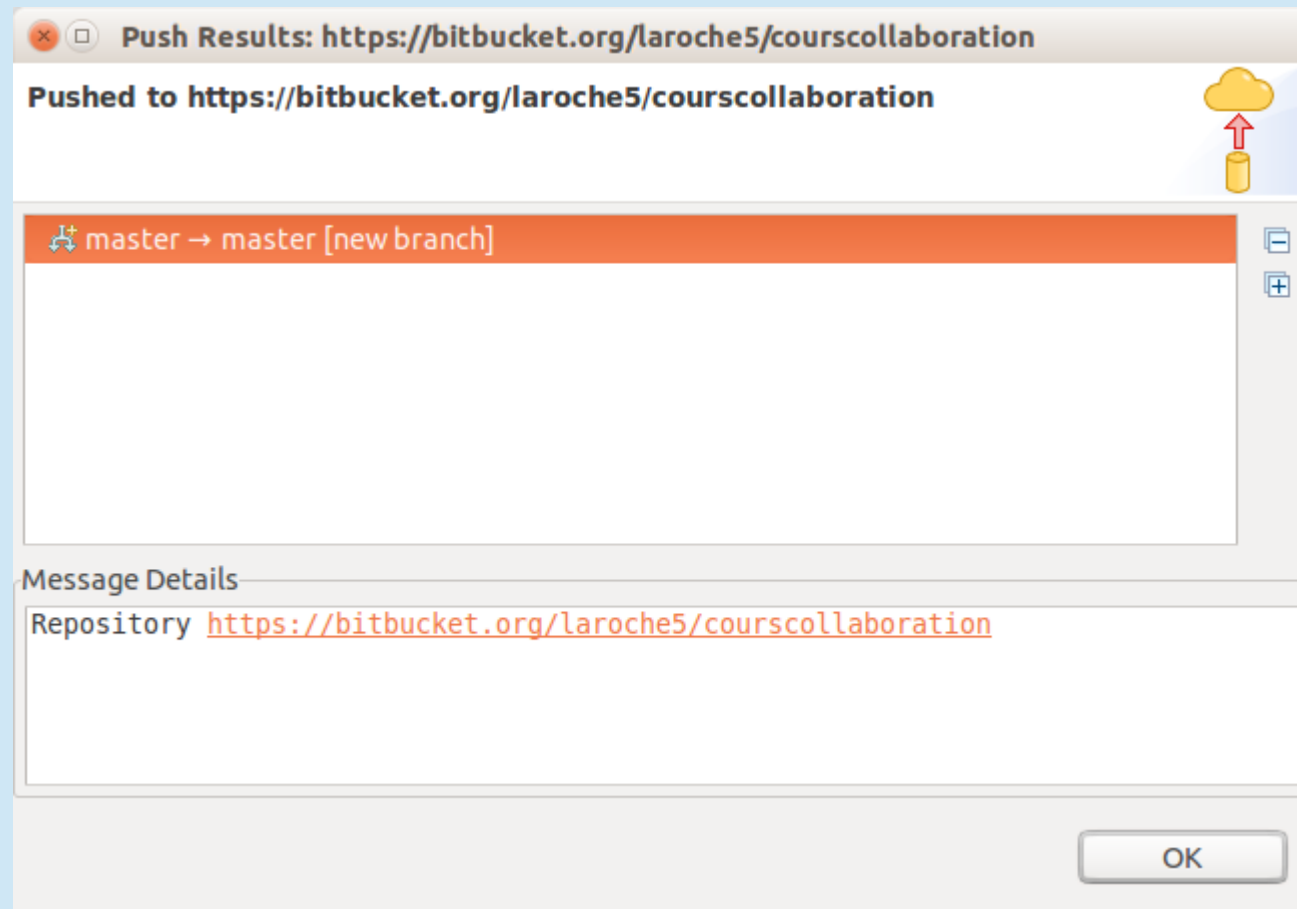
# Envoi sur les dépôts locaux et distants (6)

- Dans l'explorateur, on voit que tout est synchrone avec le dépôt  
« \* » et « + » remplacés par une icône « dépôt »



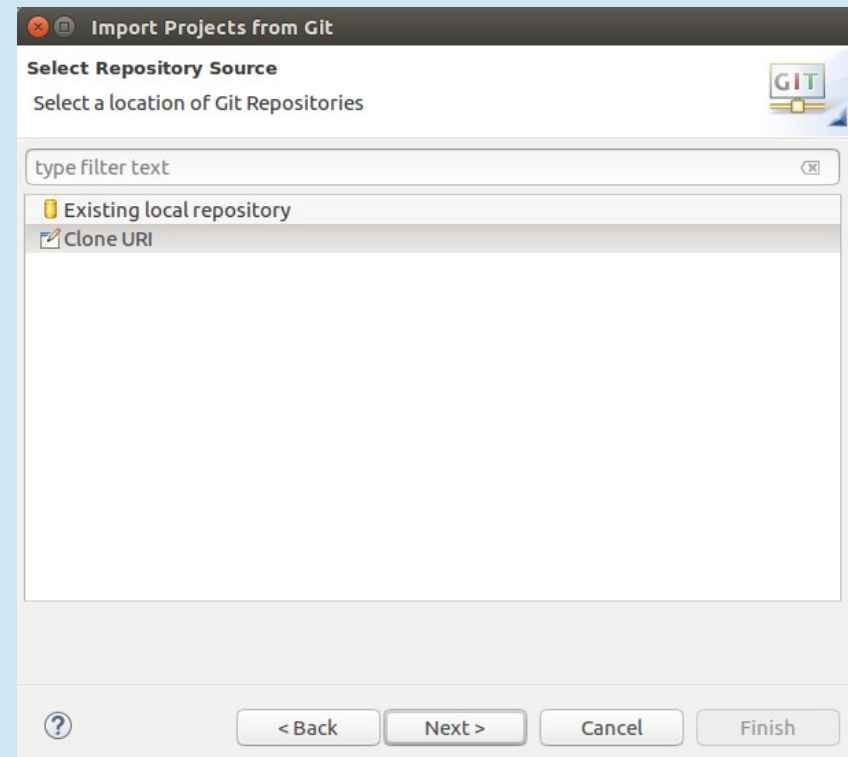
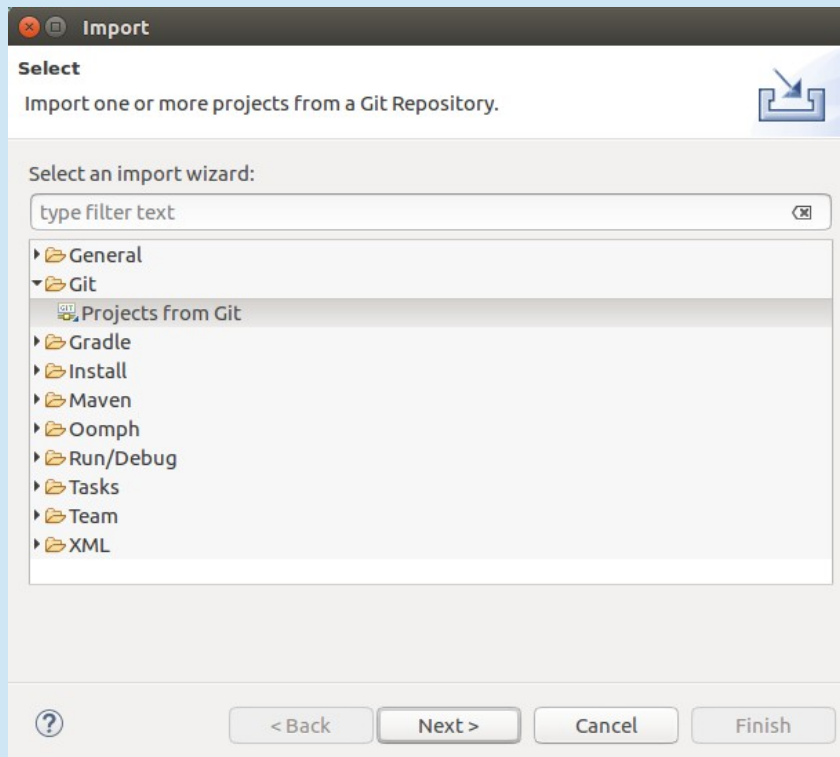
# Envoi sur les dépôts locaux et distants (7)

- Résultat : OK



# A Stéphane de travailler

- Stéphane importe le projet Git dans son workspace Eclipse
- Menu File, Import, Git, Projects from Git, Clone URI





# Import depuis le dépôt distant (1)

- Stéphane se connecte au dépôt distant commun
- Avec son identifiant à lui

Import Projects from Git

Source Git Repository

Enter the location of the source repository.

Location

URI:  Local File...

Host:

Repository path:

Connection

Protocol:

Port:

Authentication

User:

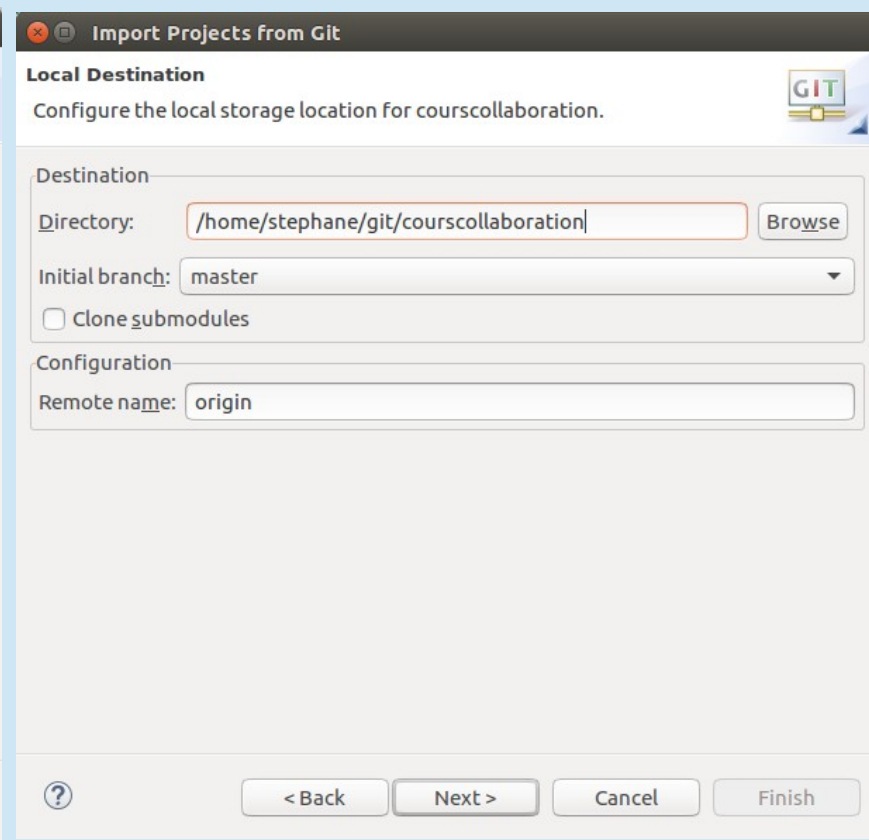
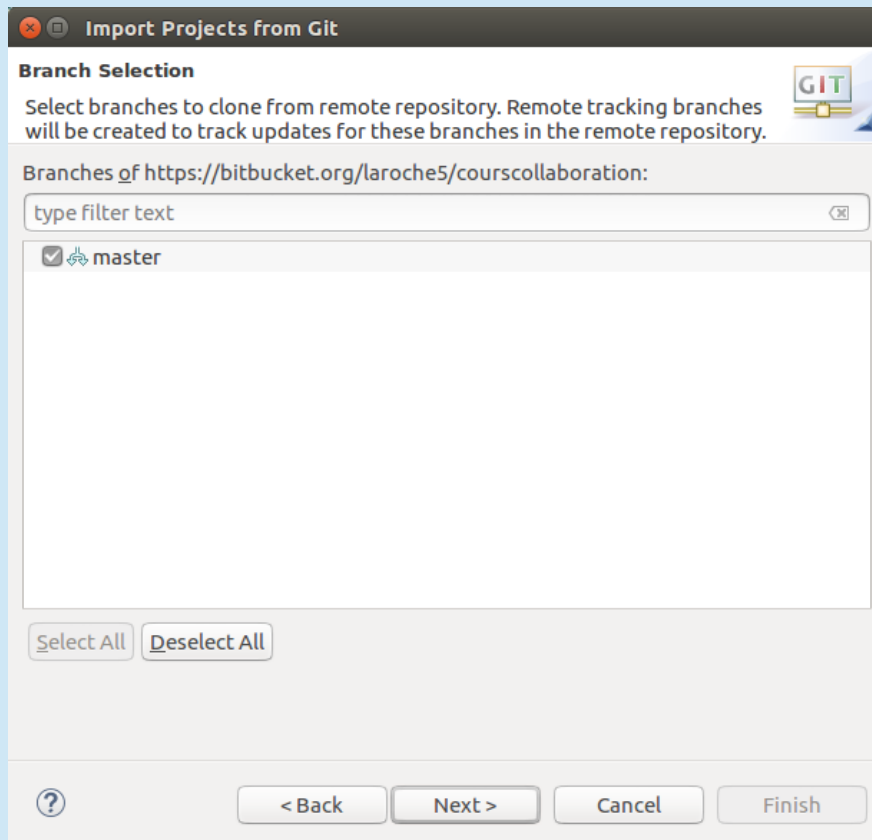
Password:

☒ Store in Secure Store

? < Back Next > Cancel Finish

# Import depuis le dépôt distant (2)

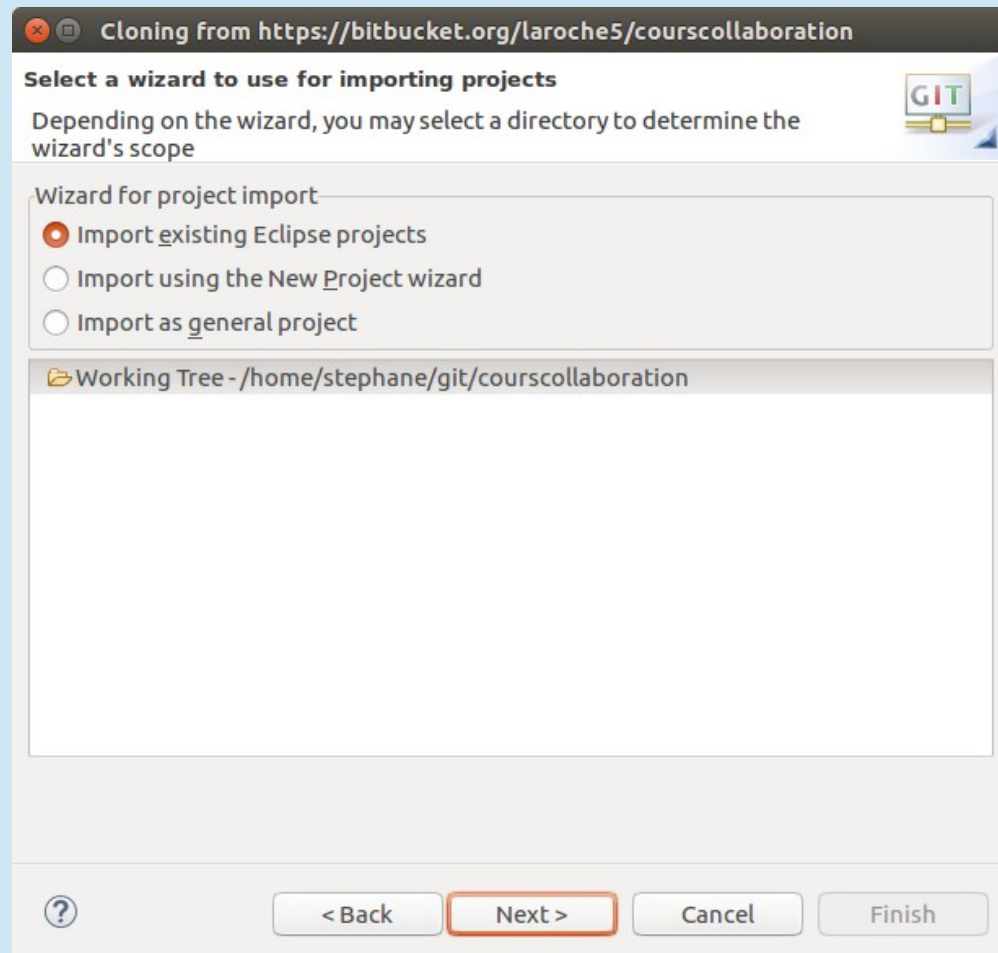
- Stéphane importe la branche « master » dans son dépôt local



# Import depuis le dépôt distant (3)

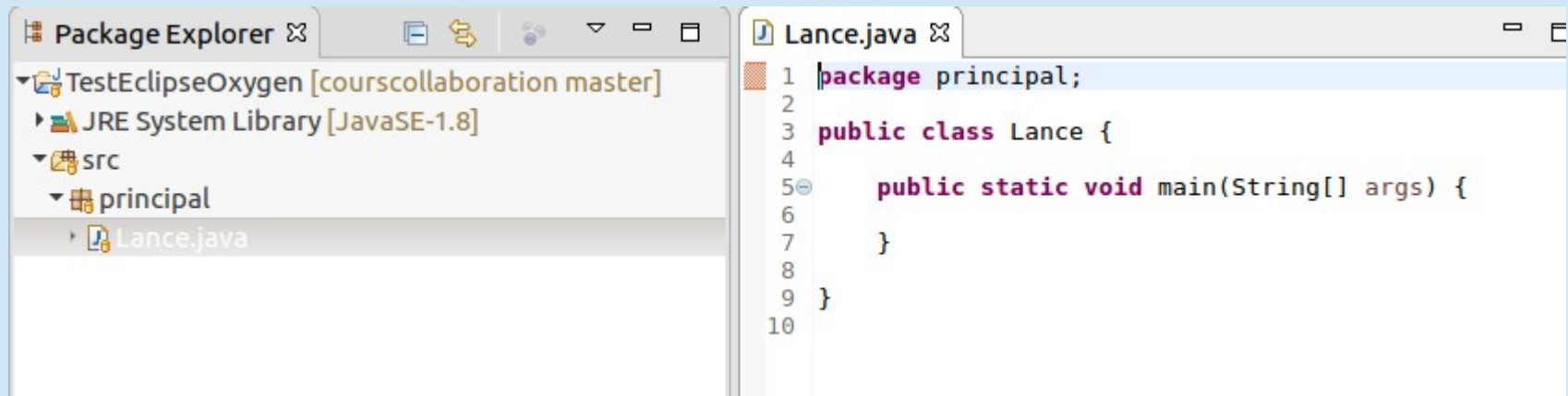
- Stéphane choisit d'importer

le projet  
du dépôt  
distant dans  
son dépôt local  
et son  
workspace



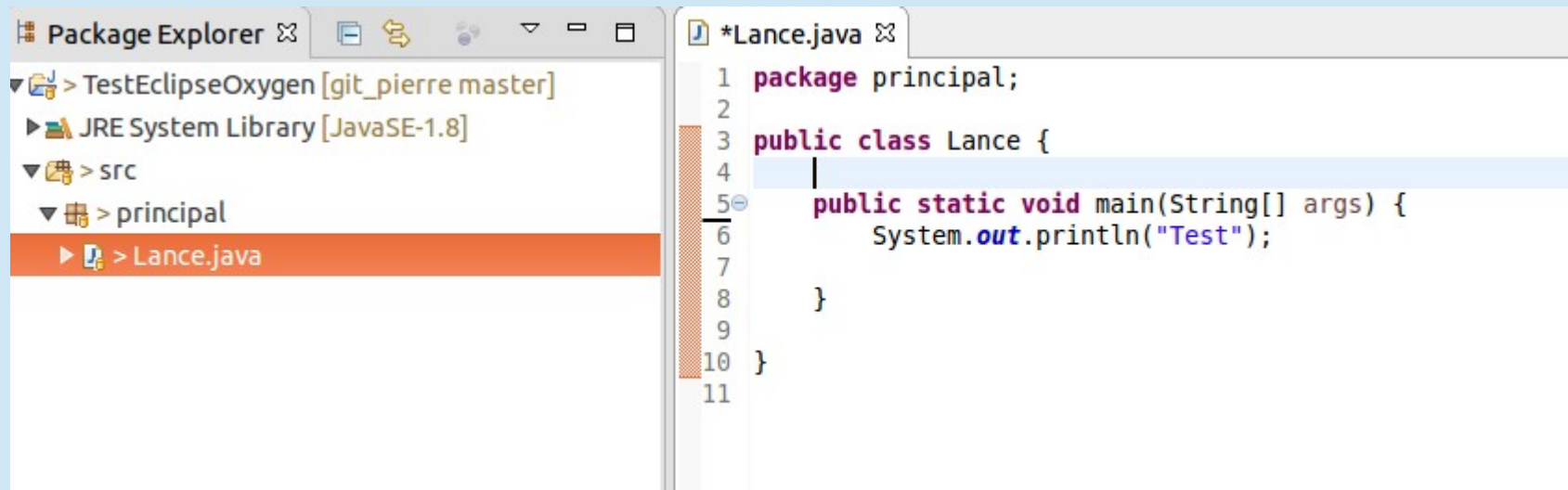
# Import depuis le dépôt distant (4)

- Stéphane a récupéré le projet à jour



# Gestion des conflits (1)

- **Pierre** modifie le fichier Lance.java
- Commit and Push

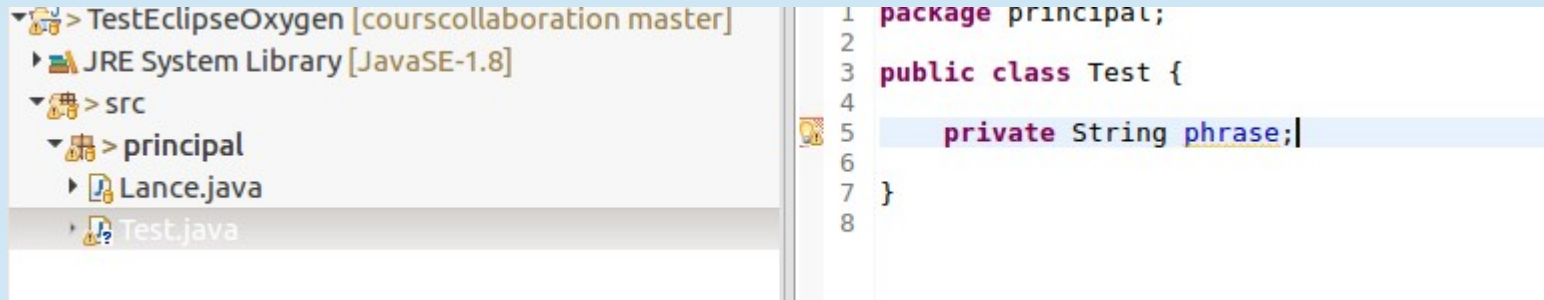


The screenshot shows the Eclipse IDE interface. On the left, the Package Explorer displays the project structure: TestEclipseOxygen [git\_pierre master] > JRE System Library [JavaSE-1.8] > src > principal > Lance.java. The file Lance.java is selected and highlighted in orange. On the right, the editor window shows the code for Lance.java. The code is as follows:

```
1 package principal;
2
3 public class Lance {
4     |
5     public static void main(String[] args) {
6         System.out.println("Test");
7     }
8 }
9
10
11
```

# Gestion des conflits (2)

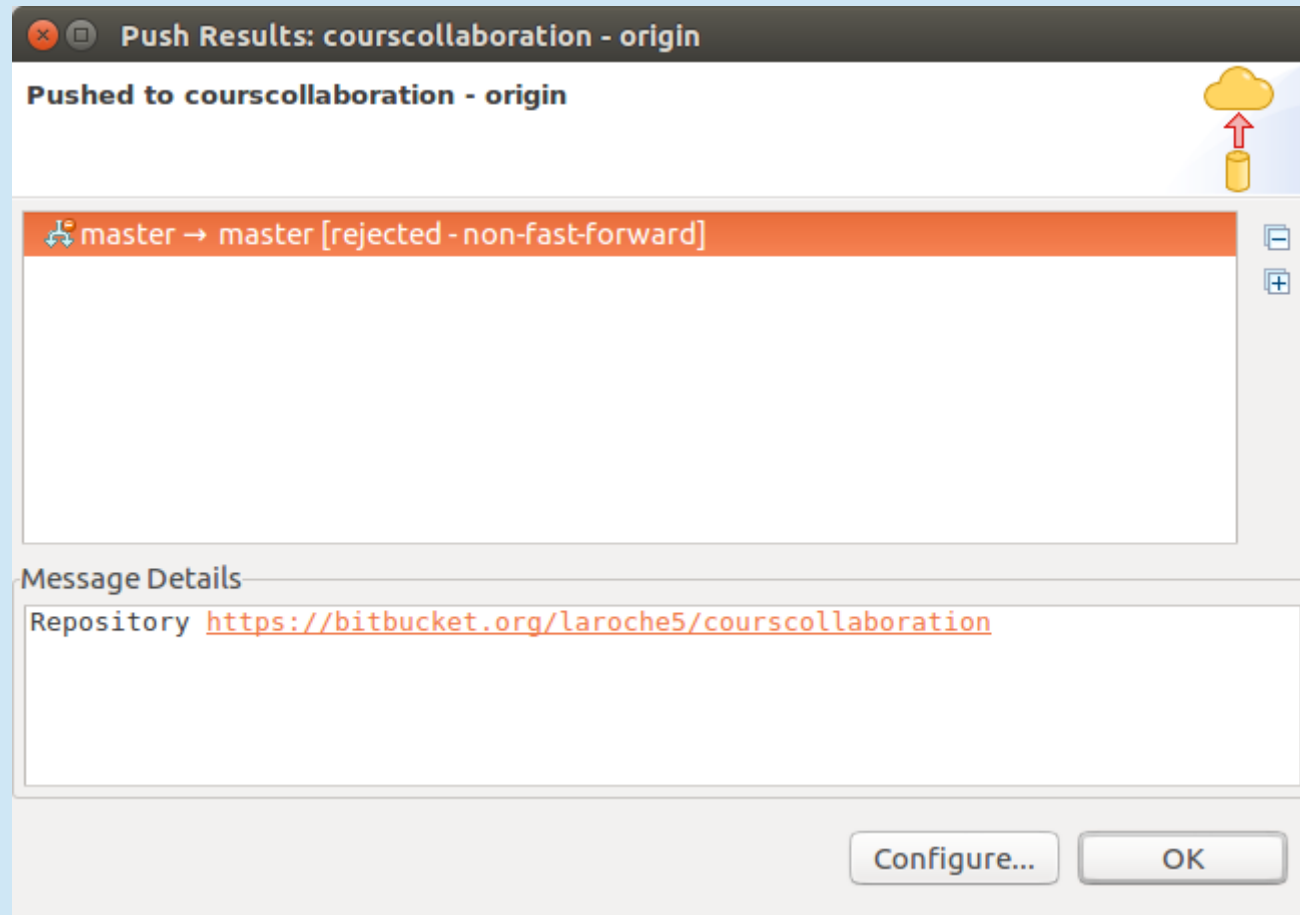
- Stéphane ajoute le fichier Test.java



- Add to Index, Commit and Push
- → Erreur : le dépôt distant a un fichier (Lance.java) plus récent que celui de Stéphane

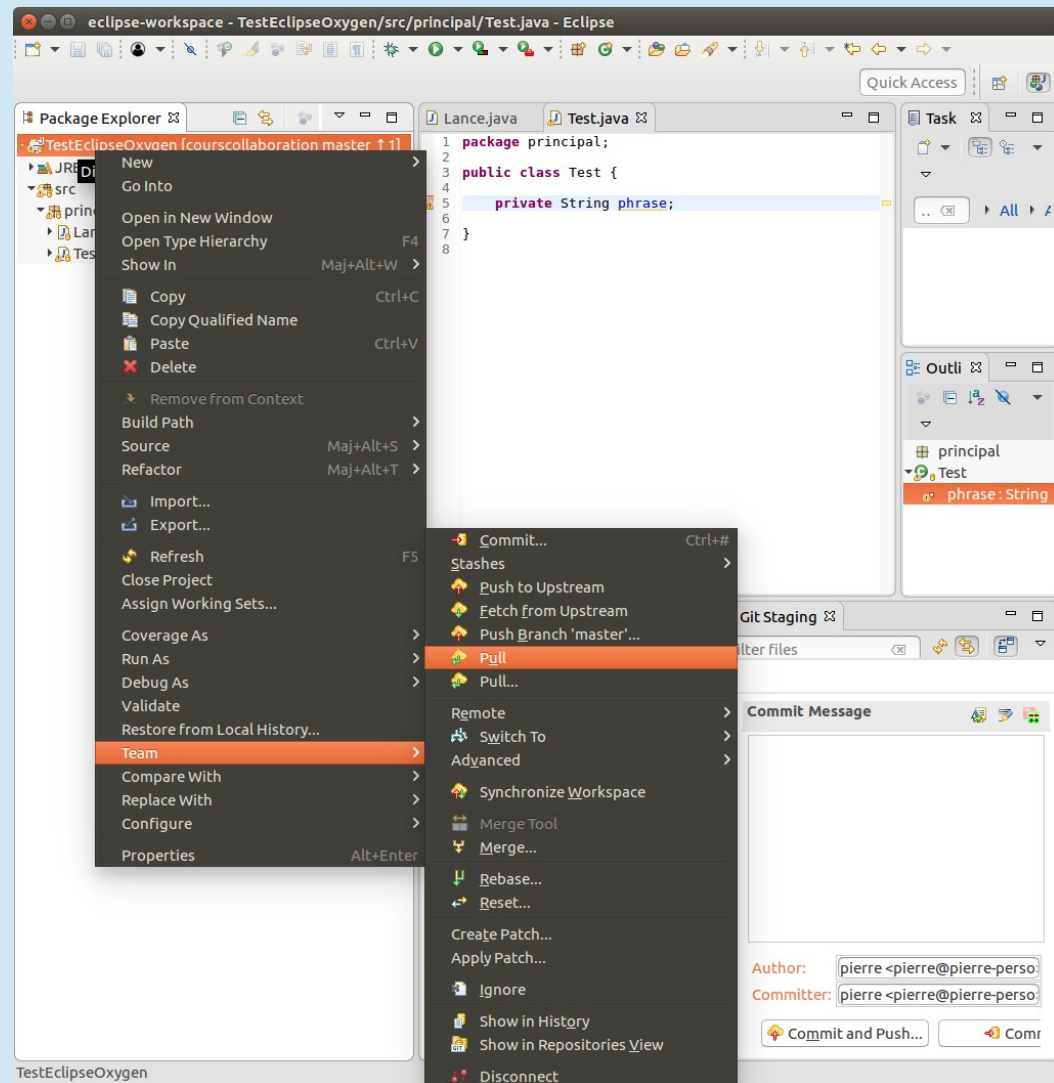
# Gestion des conflits (3)

- Fenêtre d'erreur



# Gestion des conflits (4)

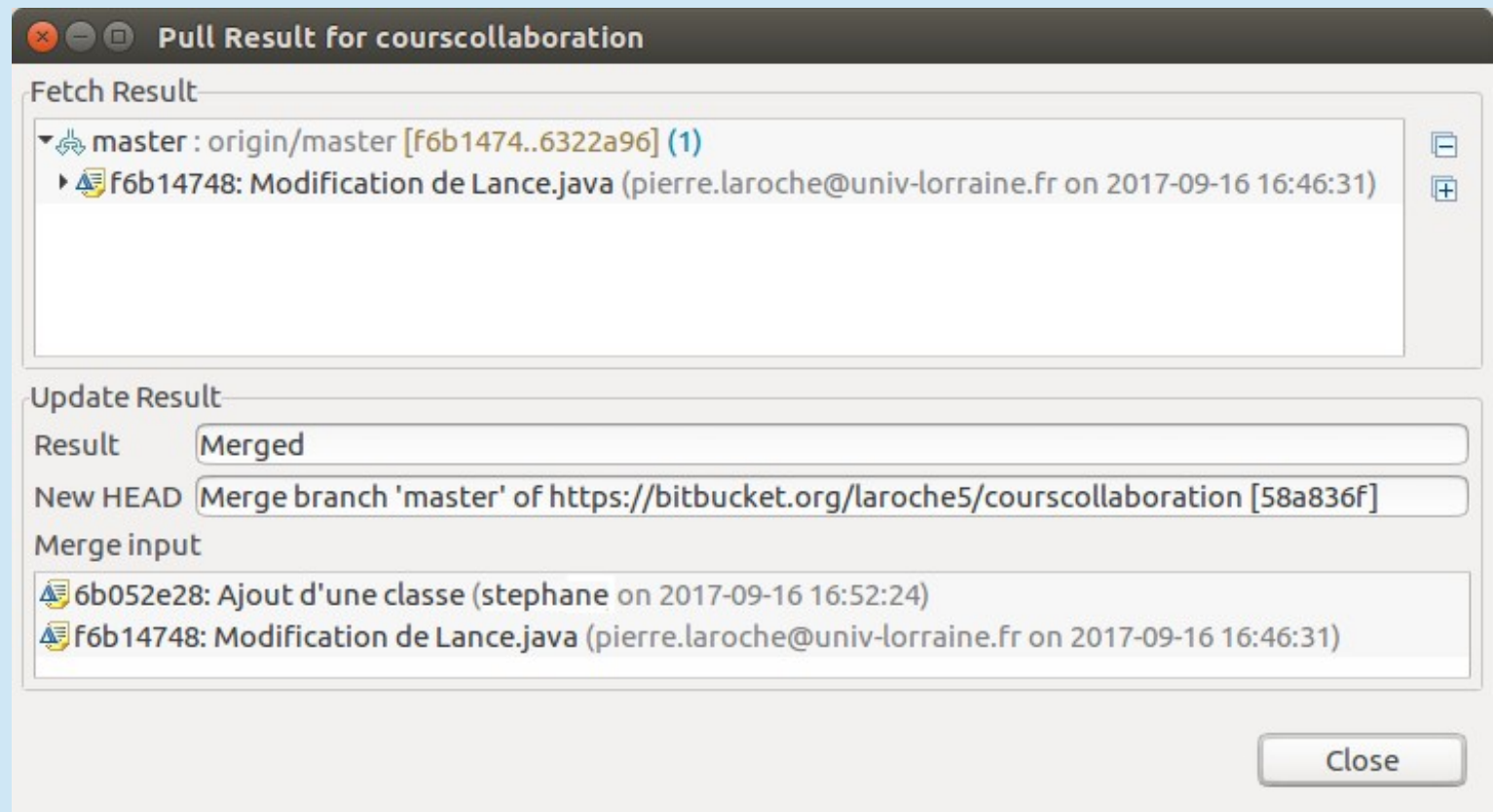
- Solution : Stéphane doit d'abord faire un Pull pour récupérer les dernières nouveautés du serveur





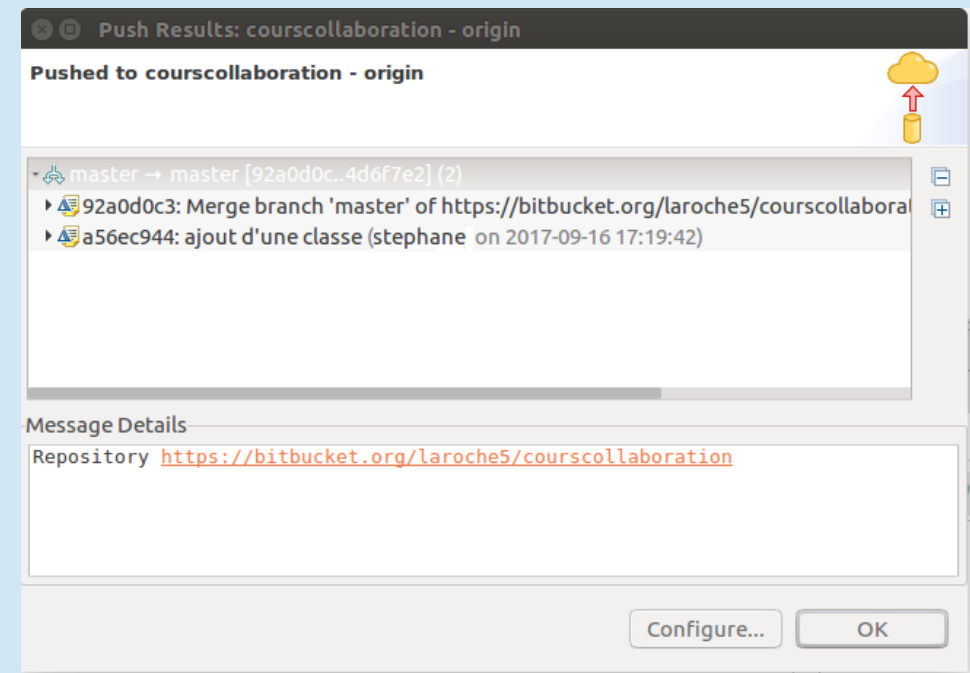
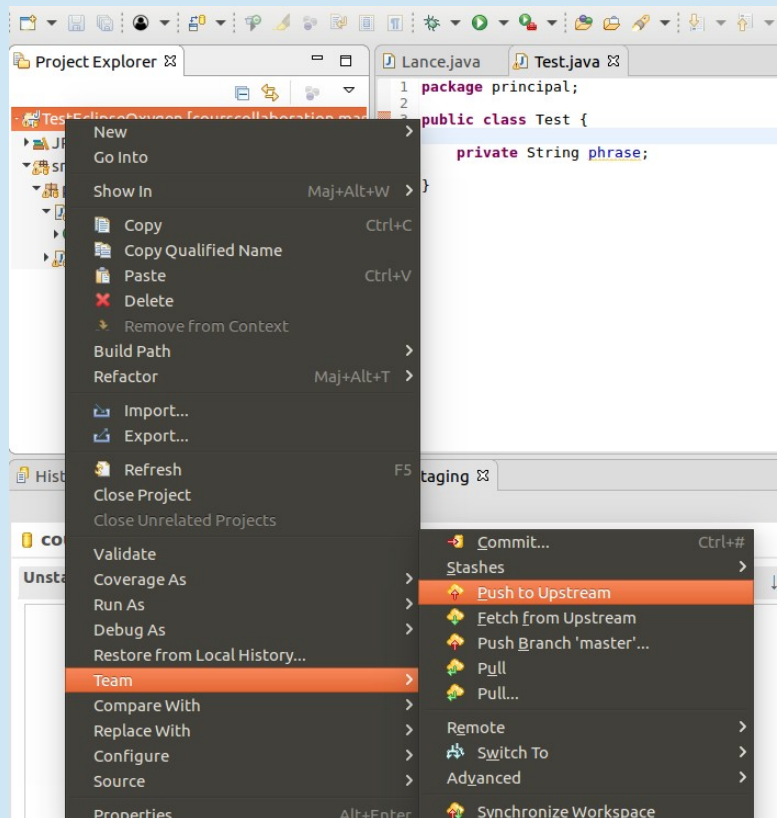
# Gestion des conflits (5)

- Résultat du pull : la modification de **Pierre** est prise en compte et la classe de **Stéphane** est ajoutée



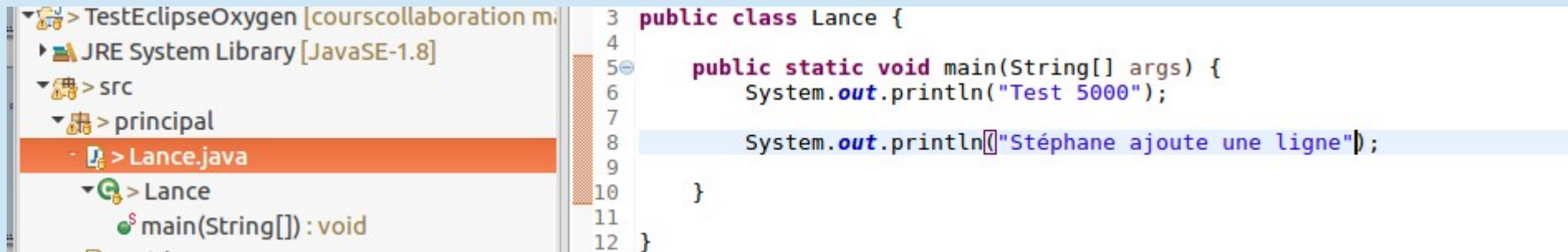
# Gestion des conflits (6)

- Le dépôt local de **Stéphane** est à jour
- Il lui reste à pousser sa classe sur le serveur, en faisant un Team, Push to Upstream



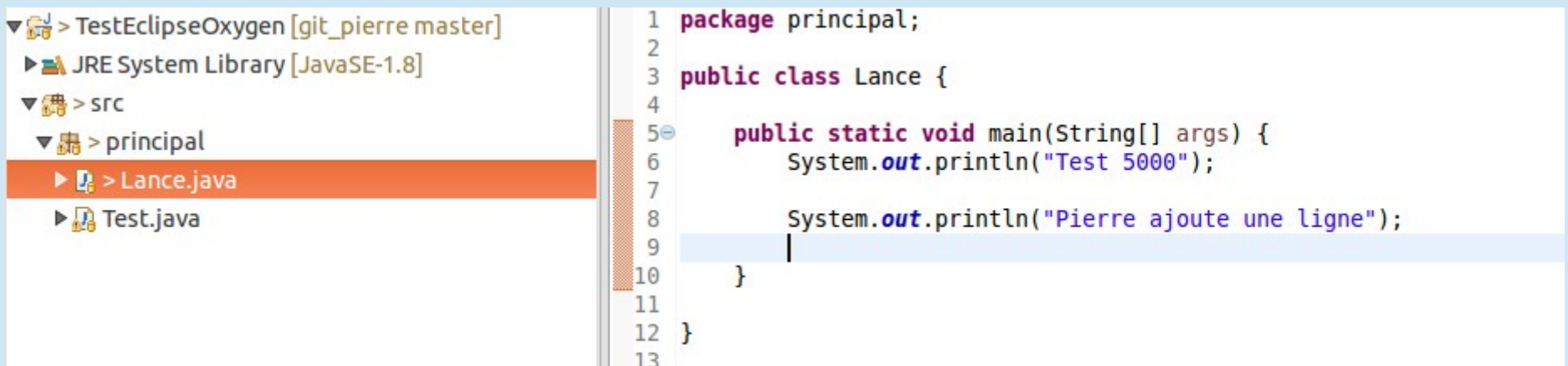
# Conflit dans le même fichier (1)

- Stéphane modifie Lance.java et fait un Commit and Push... → OK



```
3 public class Lance {
4
5     public static void main(String[] args) {
6         System.out.println("Test 5000");
7
8         System.out.println("Stéphane ajoute une ligne");
9     }
10 }
11
12 }
```

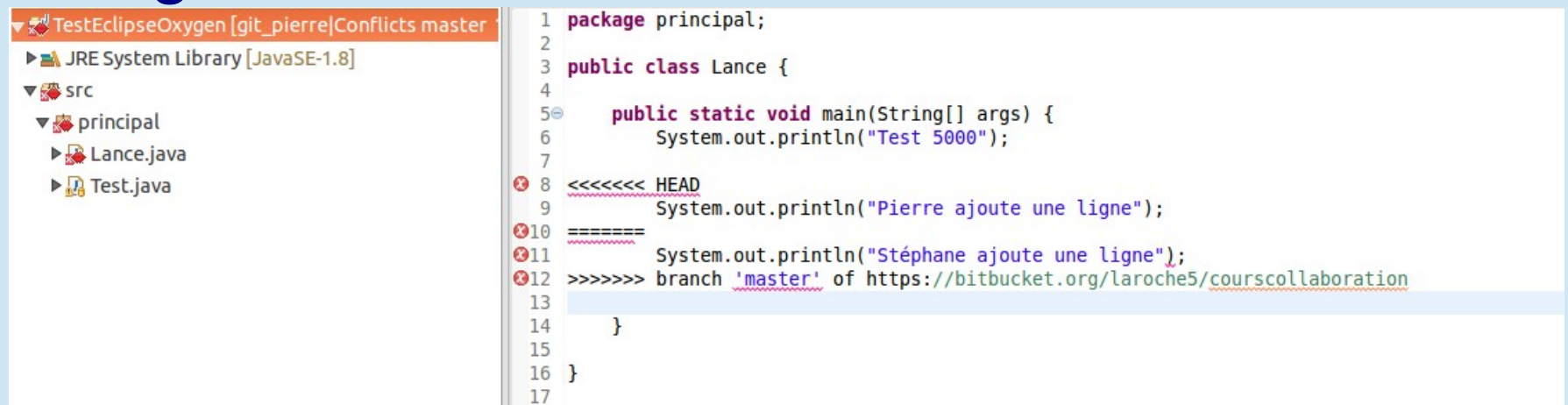
- Pierre modifie aussi et fait un Commit and Push... → échec [rejected-non-fast-forward]



```
1 package principal;
2
3 public class Lance {
4
5     public static void main(String[] args) {
6         System.out.println("Test 5000");
7
8         System.out.println("Pierre ajoute une ligne");
9     }
10 }
11
12 }
13 }
```

# Conflit dans le même fichier (2)

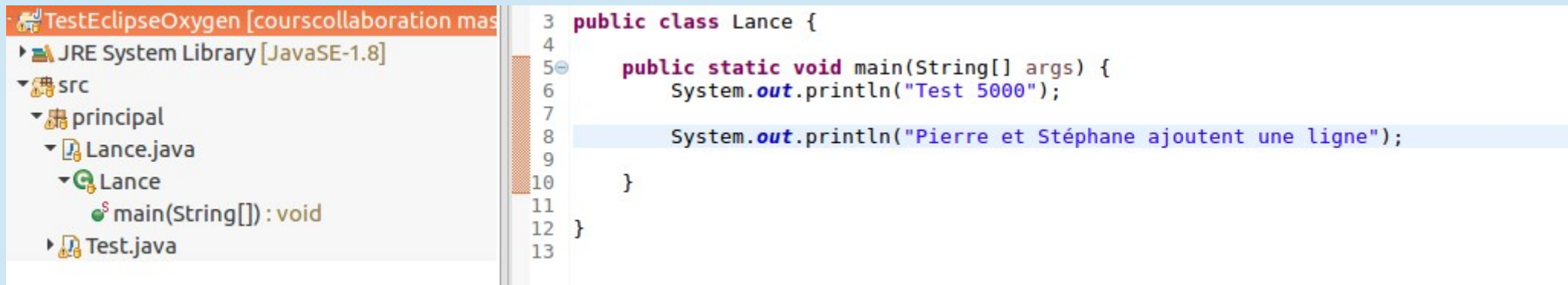
- Solution : **Pierre** fait un Pull pour avoir les dernières nouveautés du serveur
- Les deux fichiers sont fusionnés en montrant les lignes des deux versions



```
1 package principal;
2
3 public class Lance {
4
5     public static void main(String[] args) {
6         System.out.println("Test 5000");
7
8         <<<<<<< HEAD
9             System.out.println("Pierre ajoute une ligne");
10        =====
11        System.out.println("Stéphane ajoute une ligne");
12        >>>>>>> branch 'master' of https://bitbucket.org/laroche5/courscollaboration
13
14    }
15
16 }
17
```

# Conflit dans le même fichier (2)

- **Pierre** règle le conflit manuellement en choisissant le code à garder



The screenshot shows the Eclipse IDE interface. On the left, the Project Explorer displays the project structure: 'TestEclipseOxygen' (courscollaboration mas) containing 'JRE System Library [JavaSE-1.8]', 'src' folder, 'principal' folder, 'Lance.java' file, 'Lance' package, 'main(String[]): void' method, and 'Test.java' file. The main editor window shows the code for 'Lance.java'. The code is as follows:

```
3 public class Lance {  
4  
5     public static void main(String[] args) {  
6         System.out.println("Test 5000");  
7  
8         System.out.println("Pierre et Stéphane ajoutent une ligne");  
9  
10    }  
11  
12 }  
13
```

- Puis **Pierre** fait un Commit and Push...
- **Stephane**, lui, devra faire un Pull pour avoir la dernière version.