

## INF250 Programmation

Durée : 2h45

Aucun document autorisé.

URL autorisées : \*.src, \*.java.com et \*.oracle.com.

### Sommaire

INF250 Programmation .....	1
Première IHM .....	1
Deuxième partie événements .....	2
Annexes .....	6
Proxy .....	6
FTP .....	6
Eclipse .....	7

### Mise en place et restitution

Créer un « workspace » sur un disque public. Le nom du « workspace » et son chemin d'accès doivent être impérativement sans espaces, ni caractères spéciaux.

Importer le projet existant dans le « workspace » en suivant la procédure donnée en annexes.

Vous pouvez tester votre import en lançant « evenements.test.Launcher ».

Si le test fonctionne renommer votre projet avec « nom\_prenom » (sans espaces, ni caractères spéciaux).

A la fin vous devrez exporter votre projet et le déposer par ftp sur ftp-exam.src/jberdjugin/inf250/exam\_tp/Votre\_Groupe, là encore une procédure est donnée en annexes.

## Première IHM

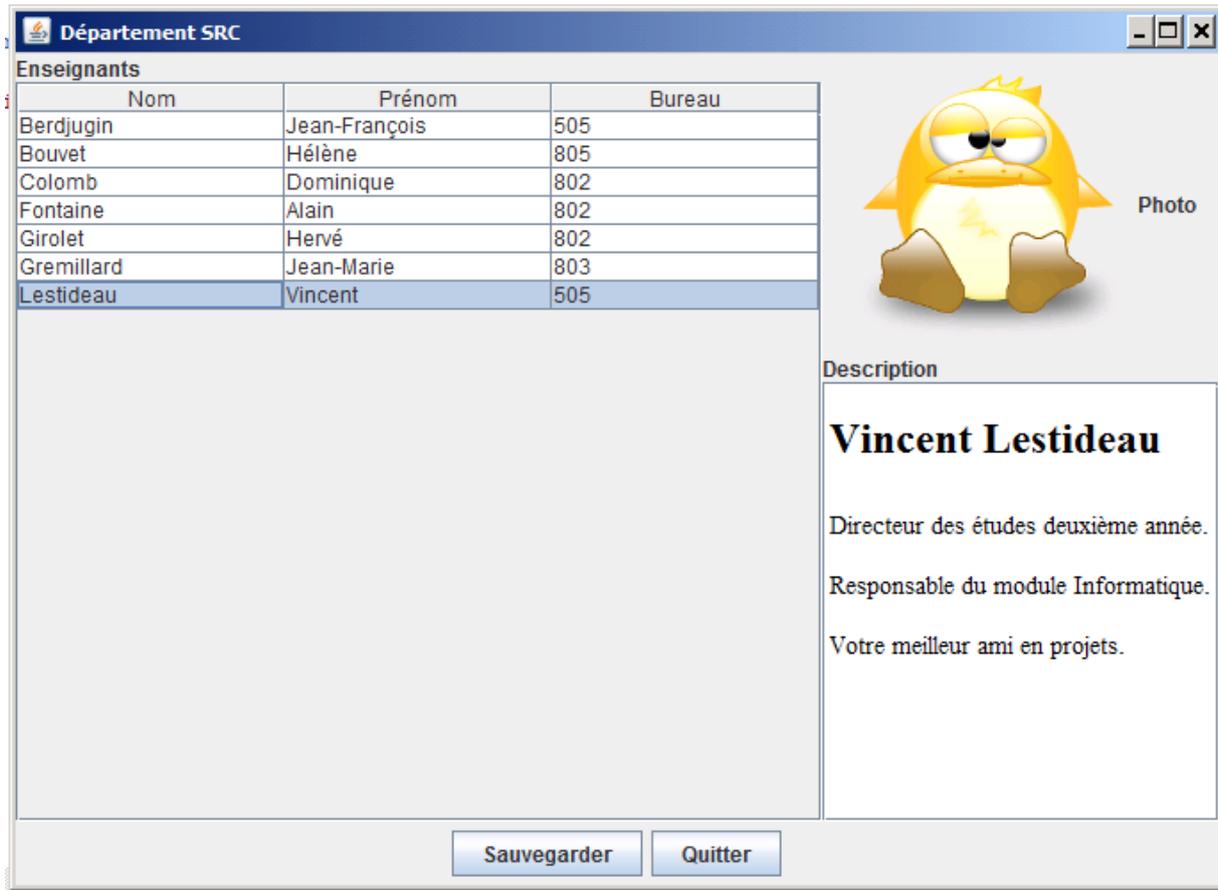


Figure 1 : Design à reproduire

Le « JTable » et le « JEditorPane » sont dans des « JScrollPane ».

La dernière ligne du « JTable » doit-être sélectionnée.

Le « JEditorPane » à une taille de 400 par 400 son type mime est "text/html".

Pour associer une image à un JLabel vous devez :

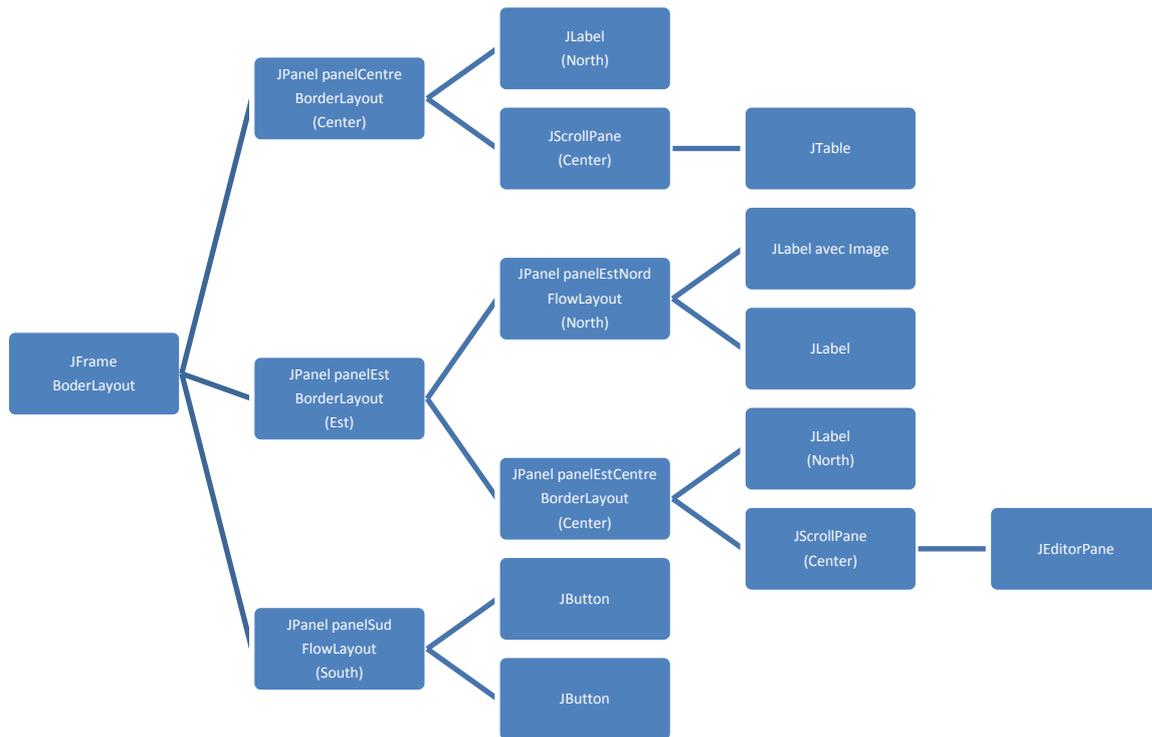
1. Fabriquer une url (URL url =  
`this.getClass().getClassLoader().getResource("ihm/ressources/photos/lestideau.png");`
2. Créer une « ImagemIcon » à partir de l'url,
3. Créer le « JLabel » à partir de l' « ImagemIcon ».

Une info bulle en anglais est un « tool tips » :

<http://docs.oracle.com/javase/tutorial/uiswing/components/tooltip.html>

La classe que vous avez à modifier est « ihm.Gestion.java », vous pouvez tester avec « ihm.test.TestGestion.java ». Le jar « ihm.jar » est fourni pour démonstration.

Je vous conseil l'architecture suivante :



## Deuxième partie événements

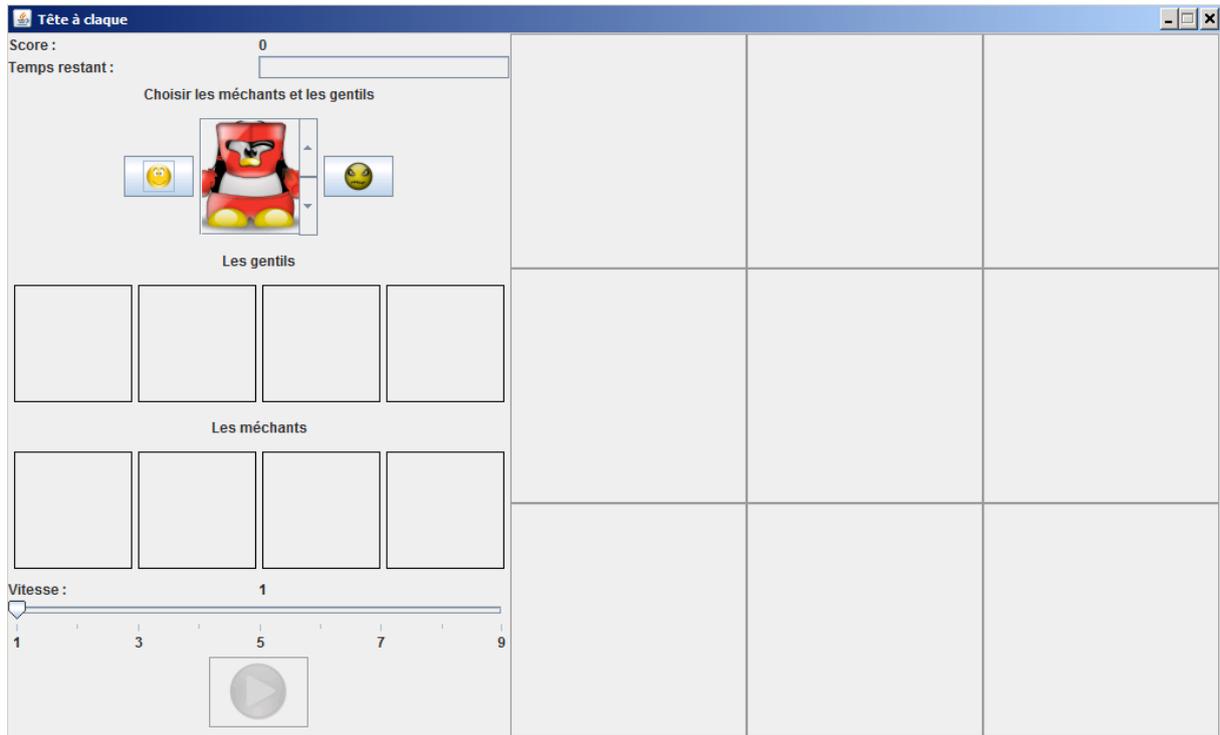


Figure 2 : Gestion des événements

La classe à modifier est « `evenements.GamaPanel.java` », vous pouvez la tester avec « `evenements.test.Launcher.java` ». Un « jar » de démonstration « `evenements.jar` » vous est fourni, il est indispensable que vous l'exécutiez pour tenter d'en reproduire le comportement.

La lecture de la méthode « `initValues()` » peut vous aider.

Les événements suivants doivent-être gérés :

1. Un clic sur le bouton « `buttonAddGentil` »  doit si le nombre de gentils sélectionnés (`nbGentils`) est inférieur à 4 :

- a. Rajouter l'icône sélectionnée par le « `JSpinner` » aux « `JLabel` » « `labelsGentils` », la méthode « `GamePanel.this.addKindIcon()` » réalise cette tâche.
- b. Augmenter le nombre de gentils sélectionnés (« `nbGentils` ») ;

- c. Activer le bouton « `GamePanel.this.buttonStart` » .
- d. Si le nombre de gentils est maintenant égal à 4 alors le bouton « `buttonAddGentil` » doit-être désactivé.

2. Un clic sur le bouton « `buttonAddMechant` »  doit si le nombre de méchants sélectionnés (`nbMechants`) est inférieur à 4 :

- a. Rajouter l'icône sélectionnée par le « `JSpinner` » aux « `JLabel` » « `labelsMechants` », la méthode « `GamePanel.this.addEvillIcon()` » réalise cette tâche.
- b. Augmenter le nombre de méchants sélectionnés (« `nbMechants` ») ;

- c. Si le nombre de méchants est maintenant égal à 4 alors le bouton « `buttonAddMechant` » doit-être désactivé.
3. Un changement d'état du « `JSpinner` » « `sliderVitesse` » doit :
  - a. Affecter à l'attribut « `GamePanel.this.vitesse` » la valeur du slider.
  - b. Modifier le texte du label « `GamePanel.this.labelVitesse` » en fonction de l'attribut `vitesse`.
4. Un clic sur le bouton « `buttonStart` » doit :
  - a. Désactiver : le « `JSlider` » « `sliderVitesse` », les « `JButton` » « `buttonAddMechant` », « `buttonAddGentil` », « `buttonStart` » et le « `JSpinner` » `spinnerTux`.
  - b. Les « `JButton` » `buttonsJeu` doivent-être activé. L'attribut `buttonsJeu` étant un tableau à deux dimensions, il vous faudra utiliser une double boucle imbriquée.

```
for (int x=0; x < 3; x++){
    for (int y=0; y < 3; y++){
        GamePanel.this.buttonsJeu[x][y].xxx; }}
```
  - c. Fabriquer un objet de type « `PasseLeTemps` » et utiliser la méthode `execute()`.

## **Annexes**

### **Proxy**

Host : proxy.src

Port : 3128

Pas de proxy pour .src (mozilla), \*.src (Internet Explorer).

### **FTP**

Host : ftp-exam.src

Répertoire : /jberdjugin/inf250/exam\_tp/votre groupe

## Eclipse

### *Importer un projet existant dans le workspace*

L'importation est celle d'un projet existant dans le workspace.

Le projet fourni est une archive que vous n'avez pas besoin de décompresser.

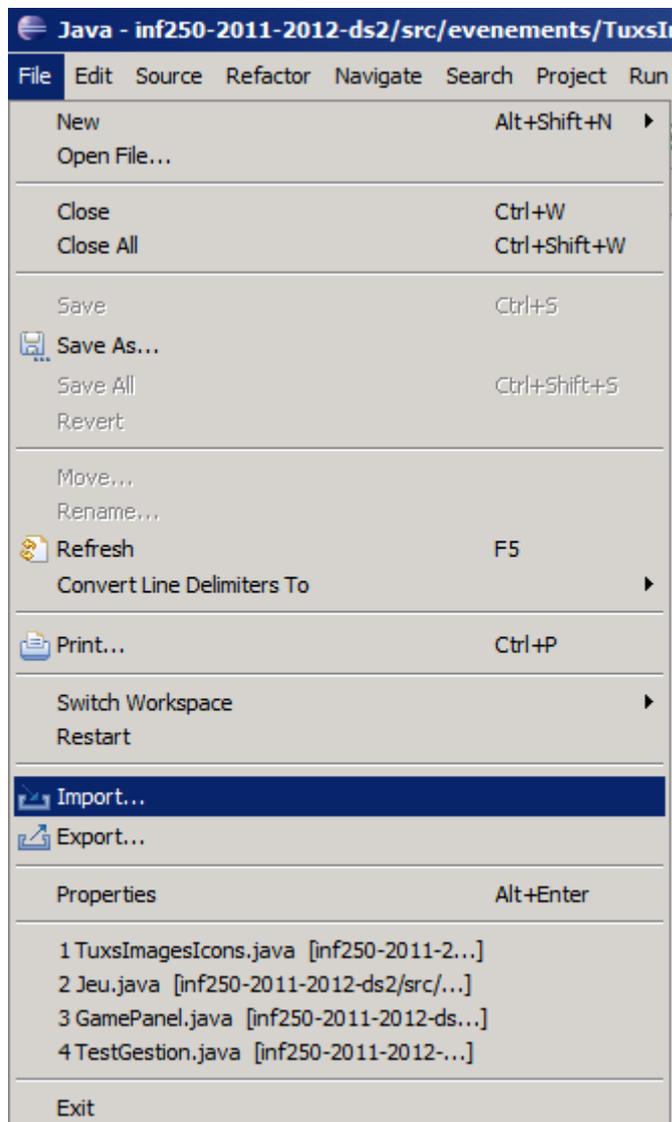


Figure 3 : File->Import

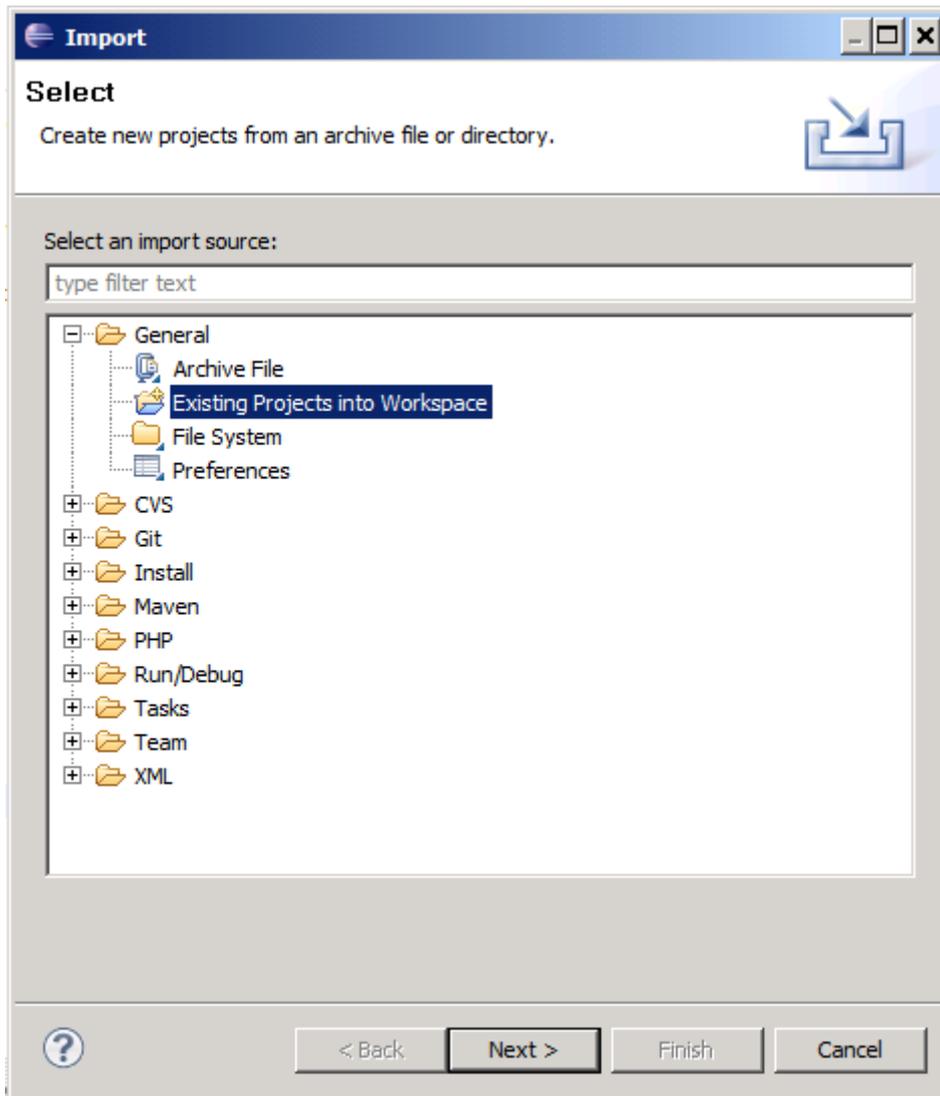


Figure 4 : Existing Projects into Workspace

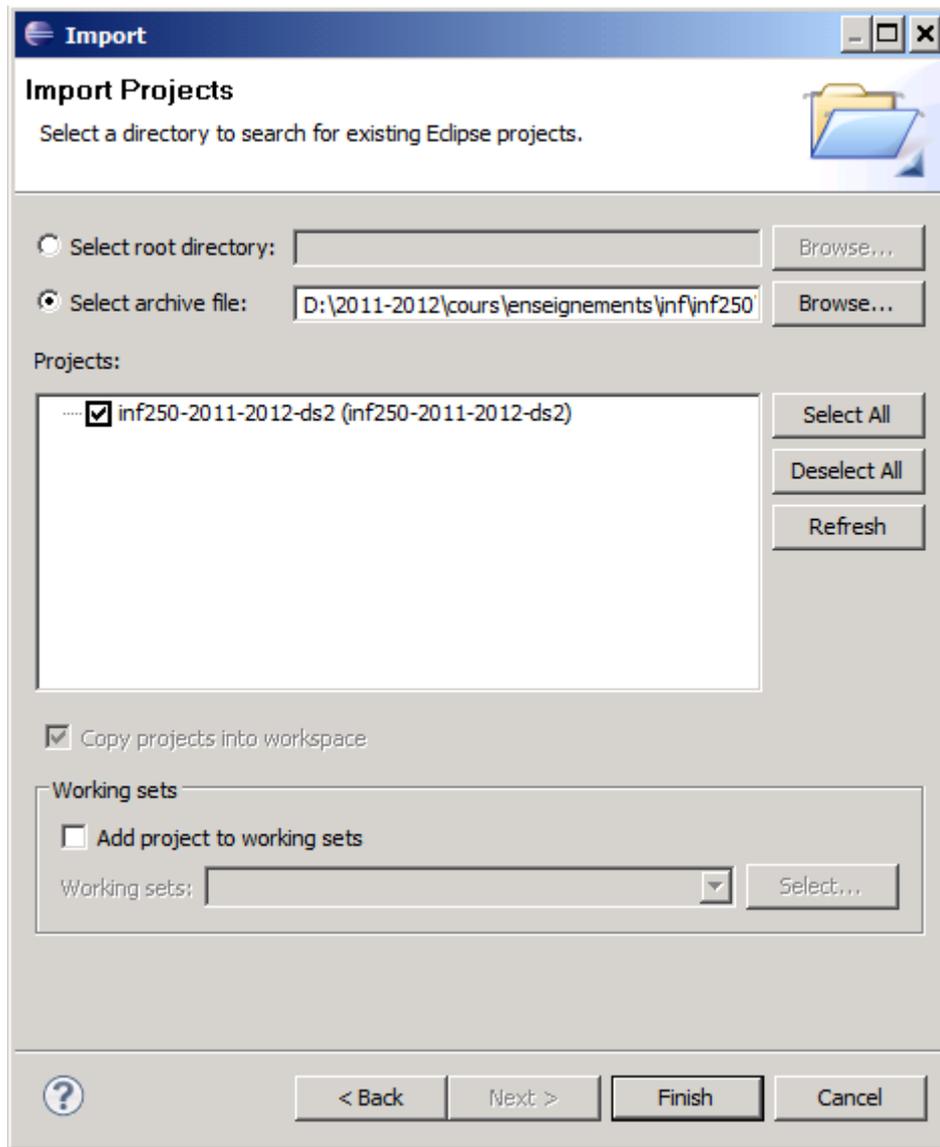


Figure 5 : Select archive file

### Configuration de client web

Pour utiliser le raccourci rapide d'accès à l'aide SHIFT+F2 vous devez d'abord indiquer votre client web.

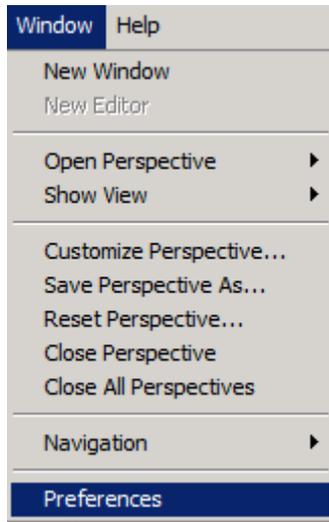


Figure 6 : Window->Preferences

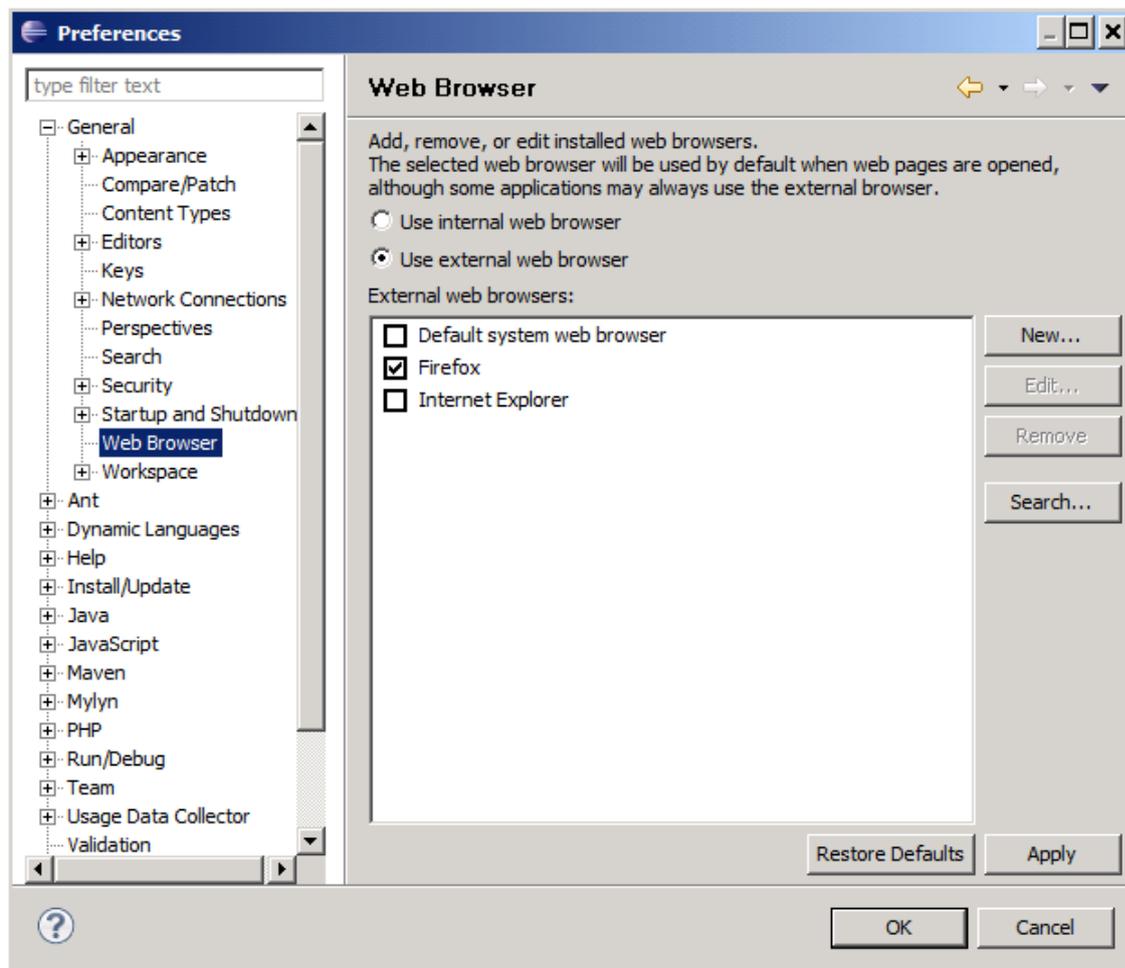


Figure 7 : General - > Web Browser

### Renommer un projet

Avant de rendre votre projet vous devrez d'abord le renommer.

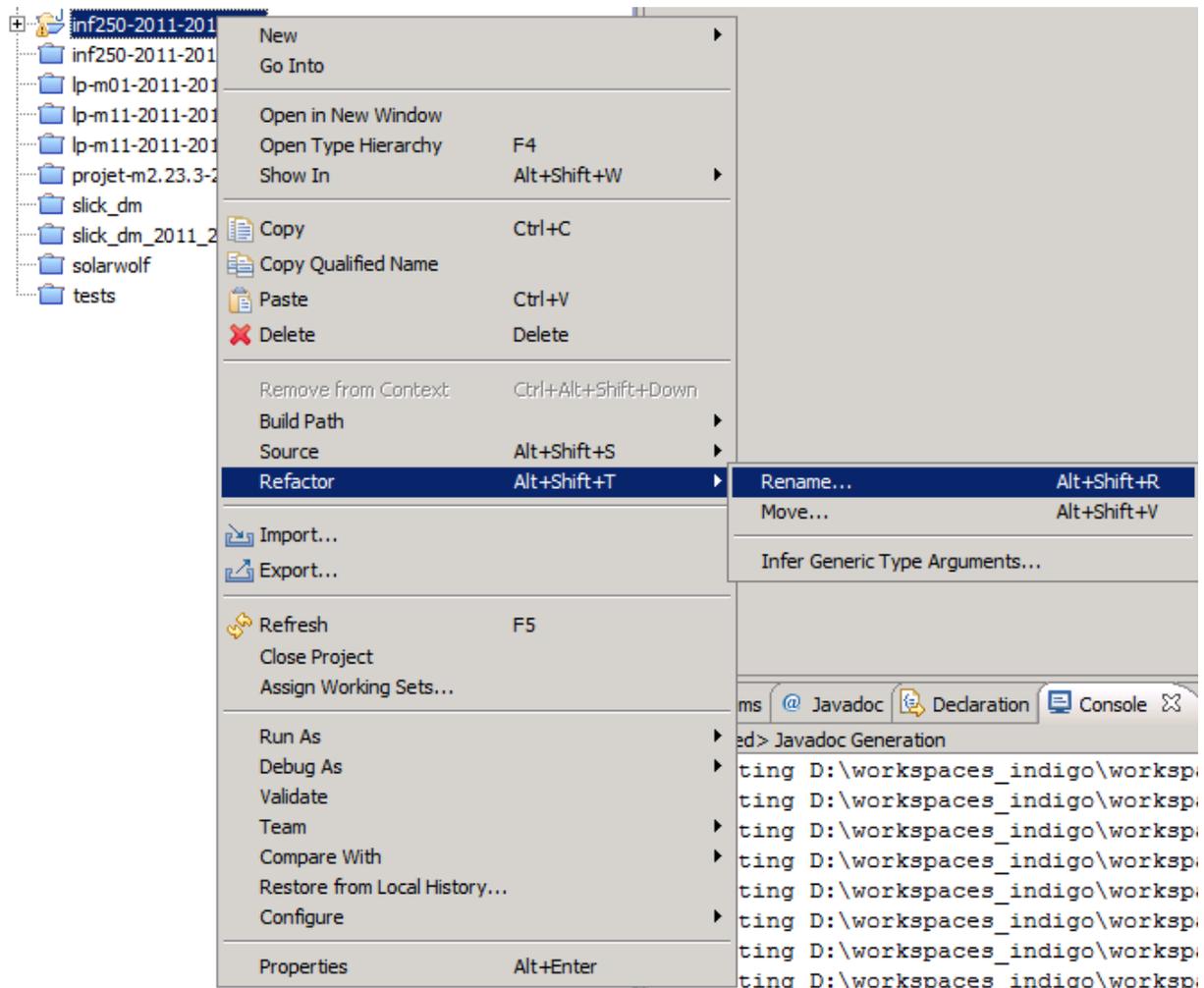


Figure 8 : click droit -> Refactor -> Rename

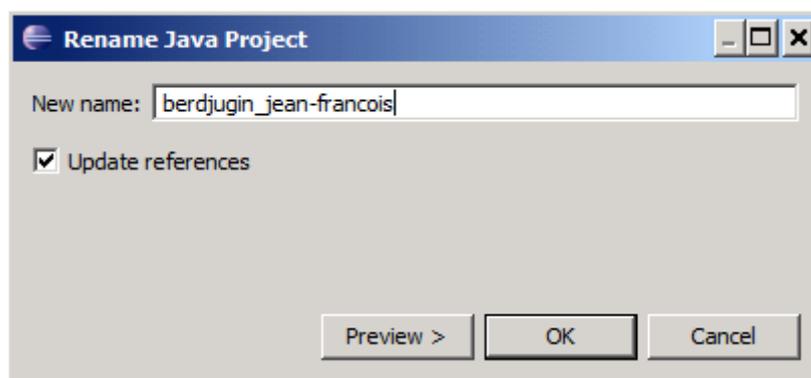


Figure 9 : Mettre votre nom et prénom

### Exporter votre projet

Pour exporter votre projet, il est préférable de commencer par le rafraichir puis de l'exporter sous forme d'archive. L'archive ne doit pas être placée dans le projet.

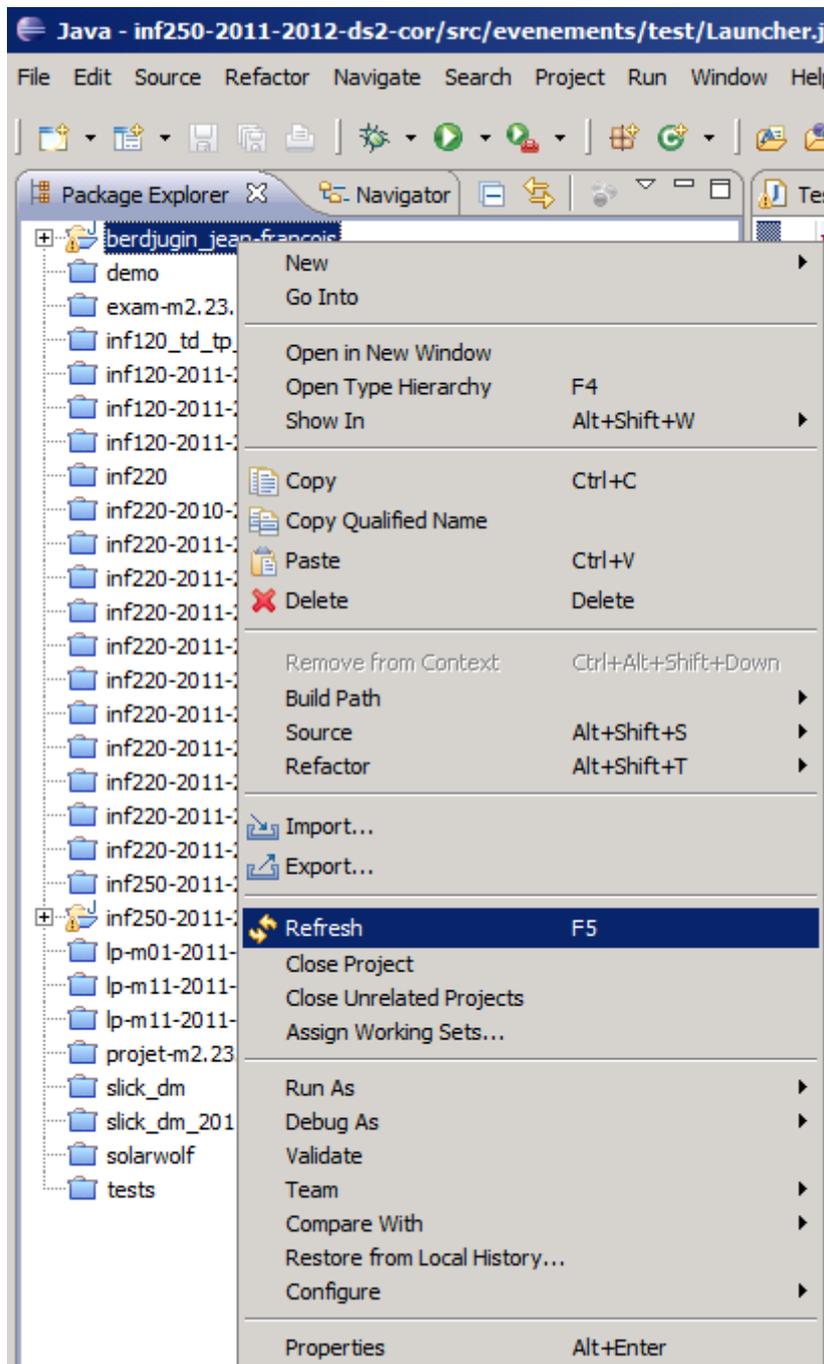


Figure 10 : click droit -> Refresh



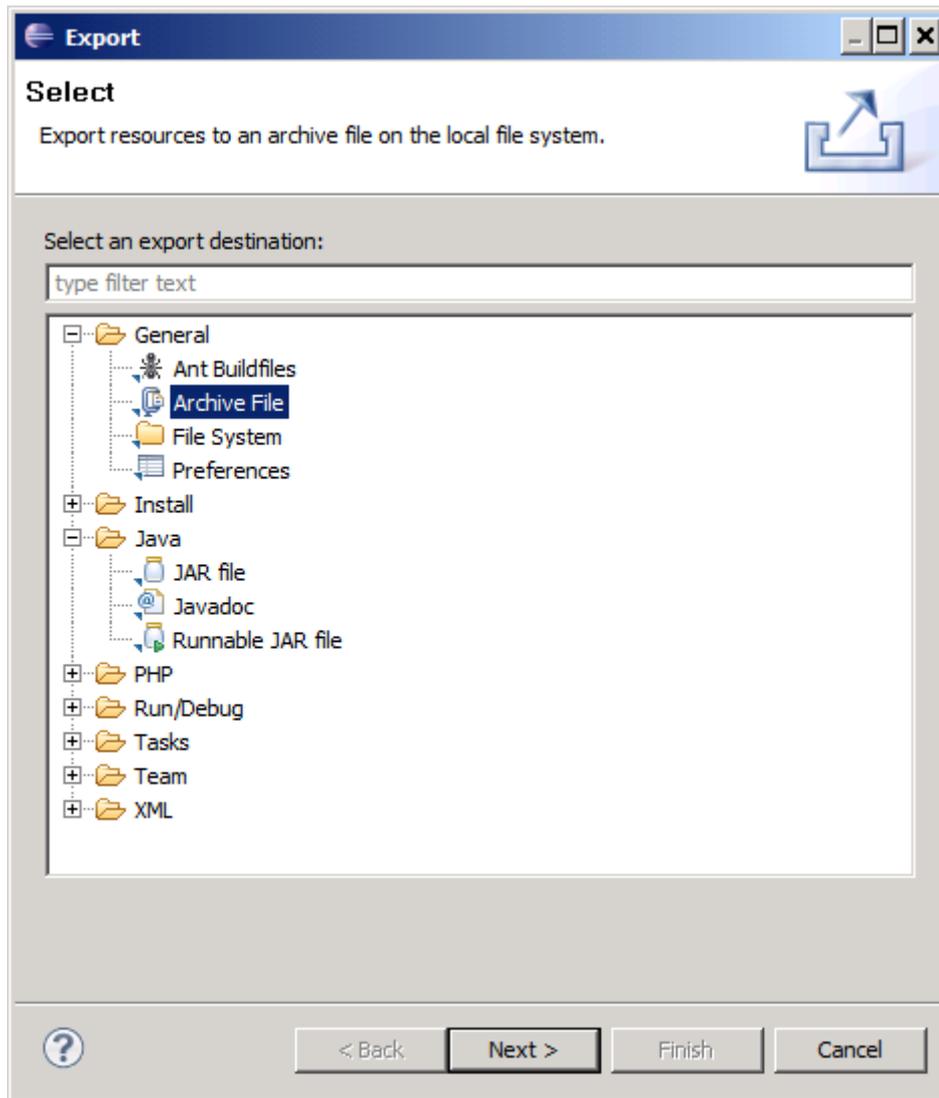


Figure 12 : Archive File

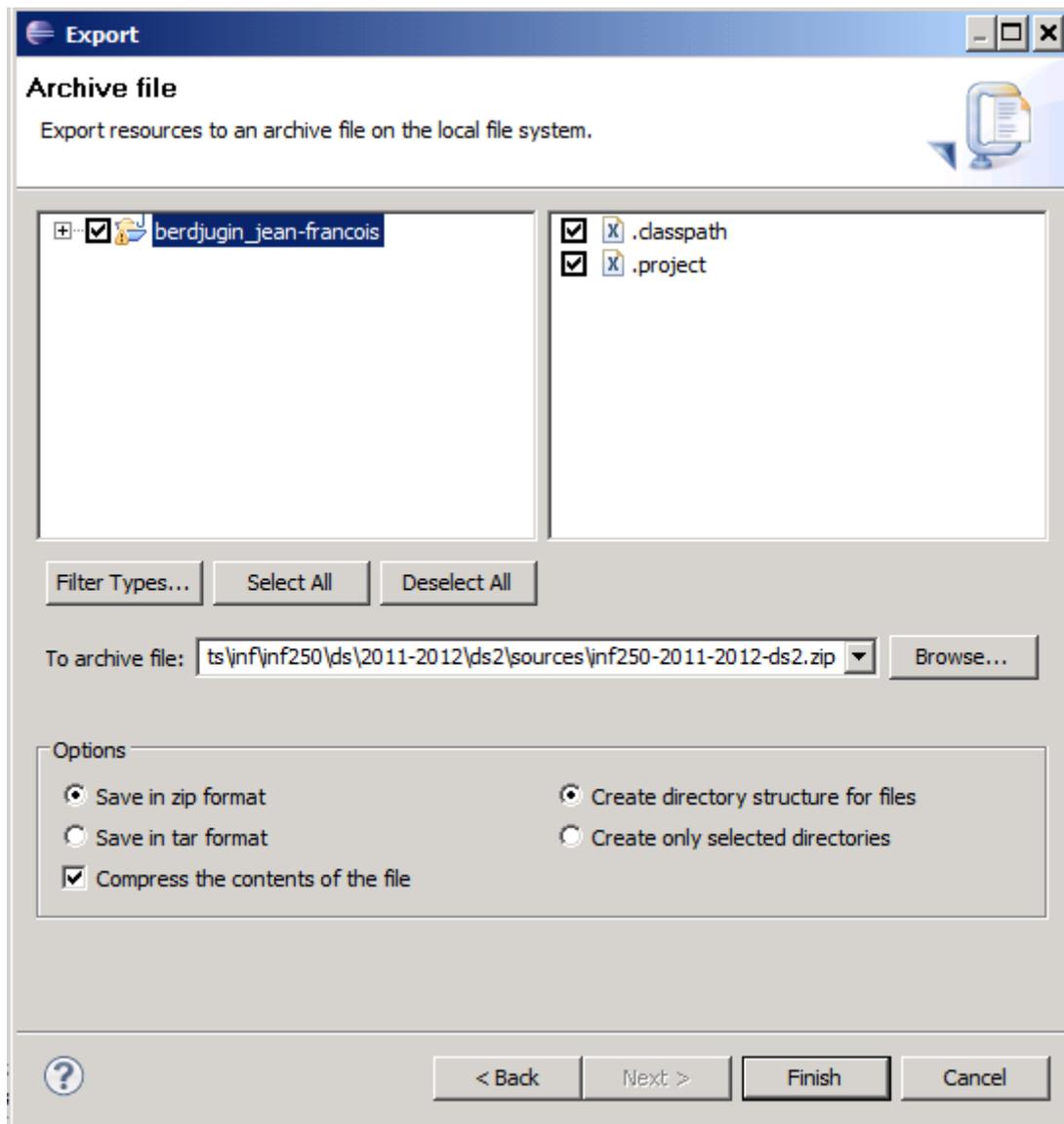


Figure 13 : Browse

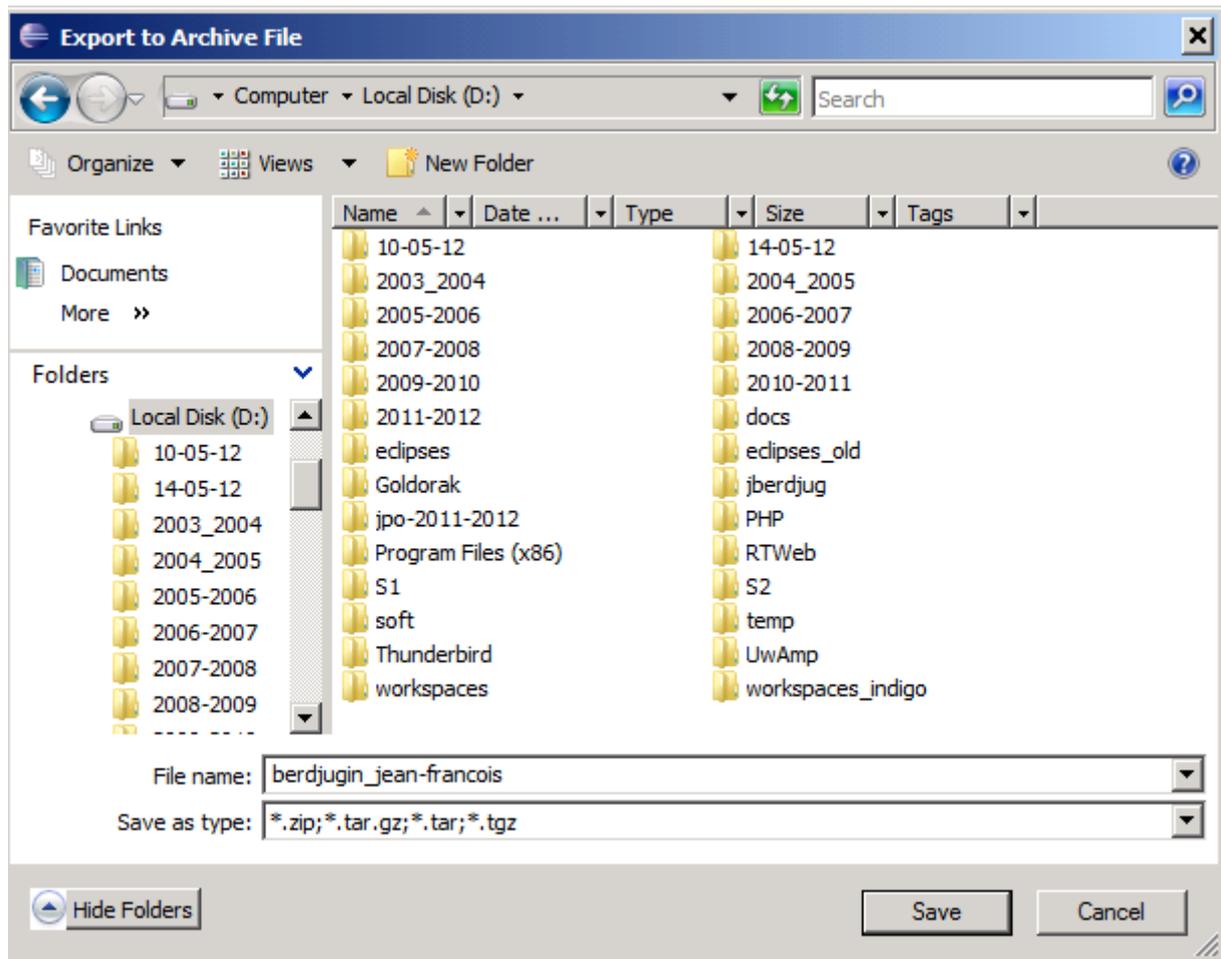


Figure 14 : Choix du nom du fichier et du répertoire de dépôt.