

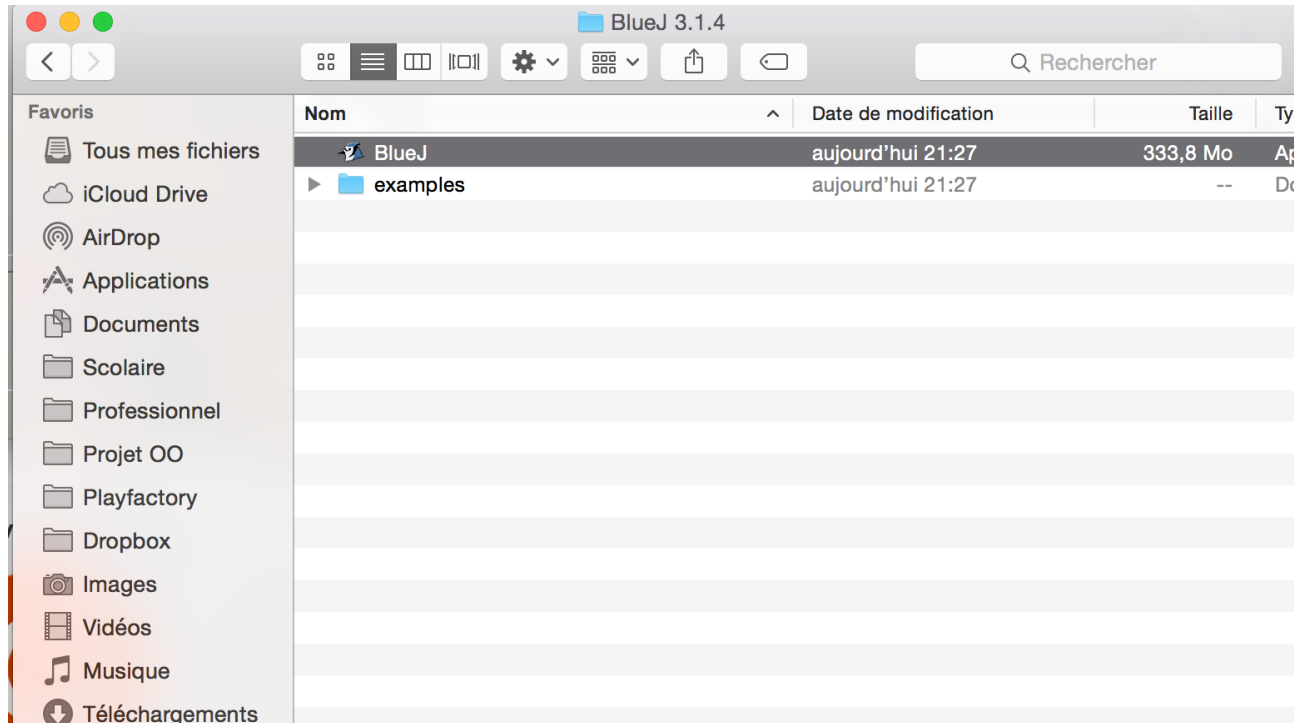
TPOO - Tutoriel d'utilisation de BlueJ

Fronczak Guillaume - Richier Axel

1) Installation de BlueJ

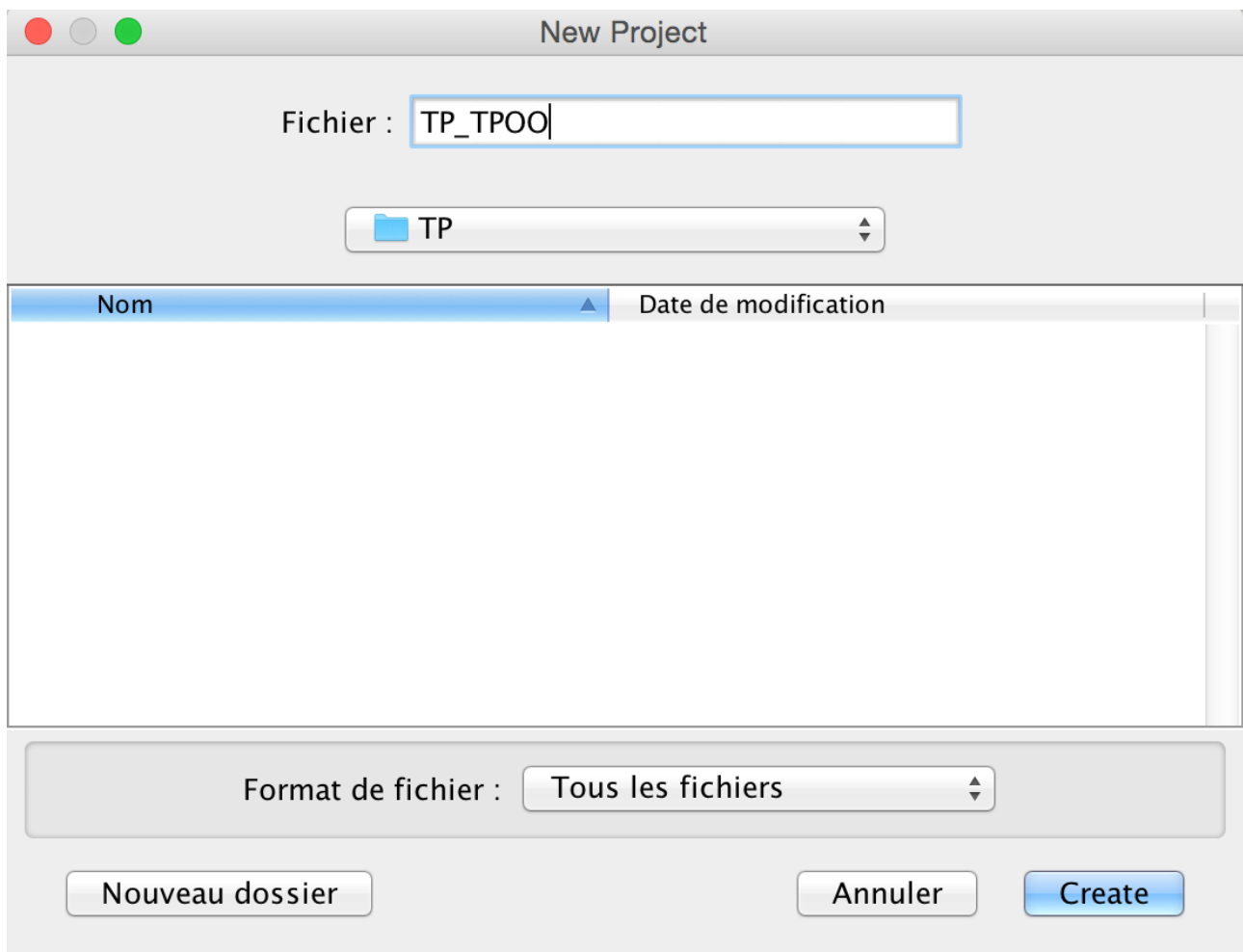


Dans un premier temps, téléchargez la version adéquate de BlueJ sur le site www.bluej.org

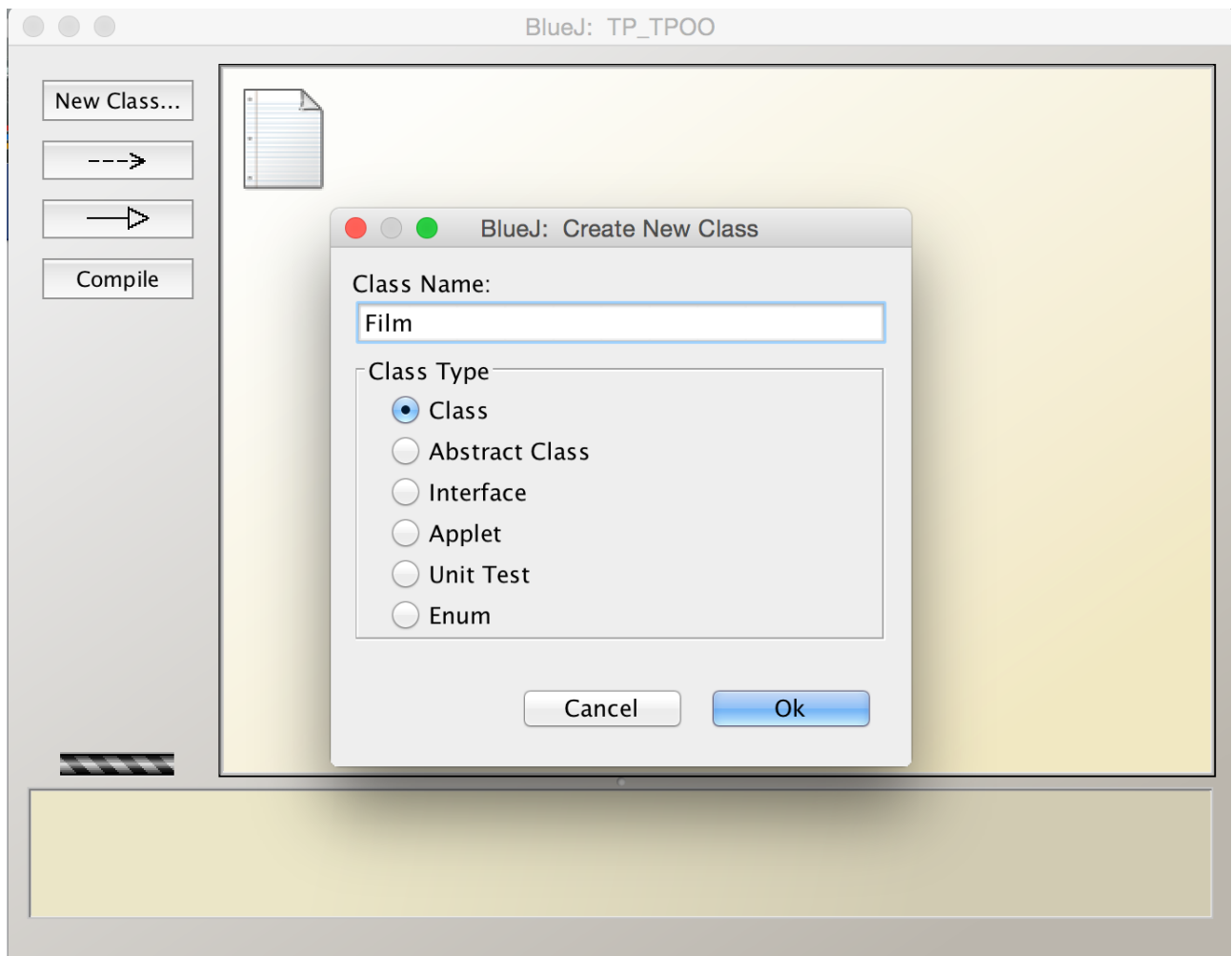


BlueJ s'exécute directement sans installation, en double-cliquant simplement sur l'icône de l'application

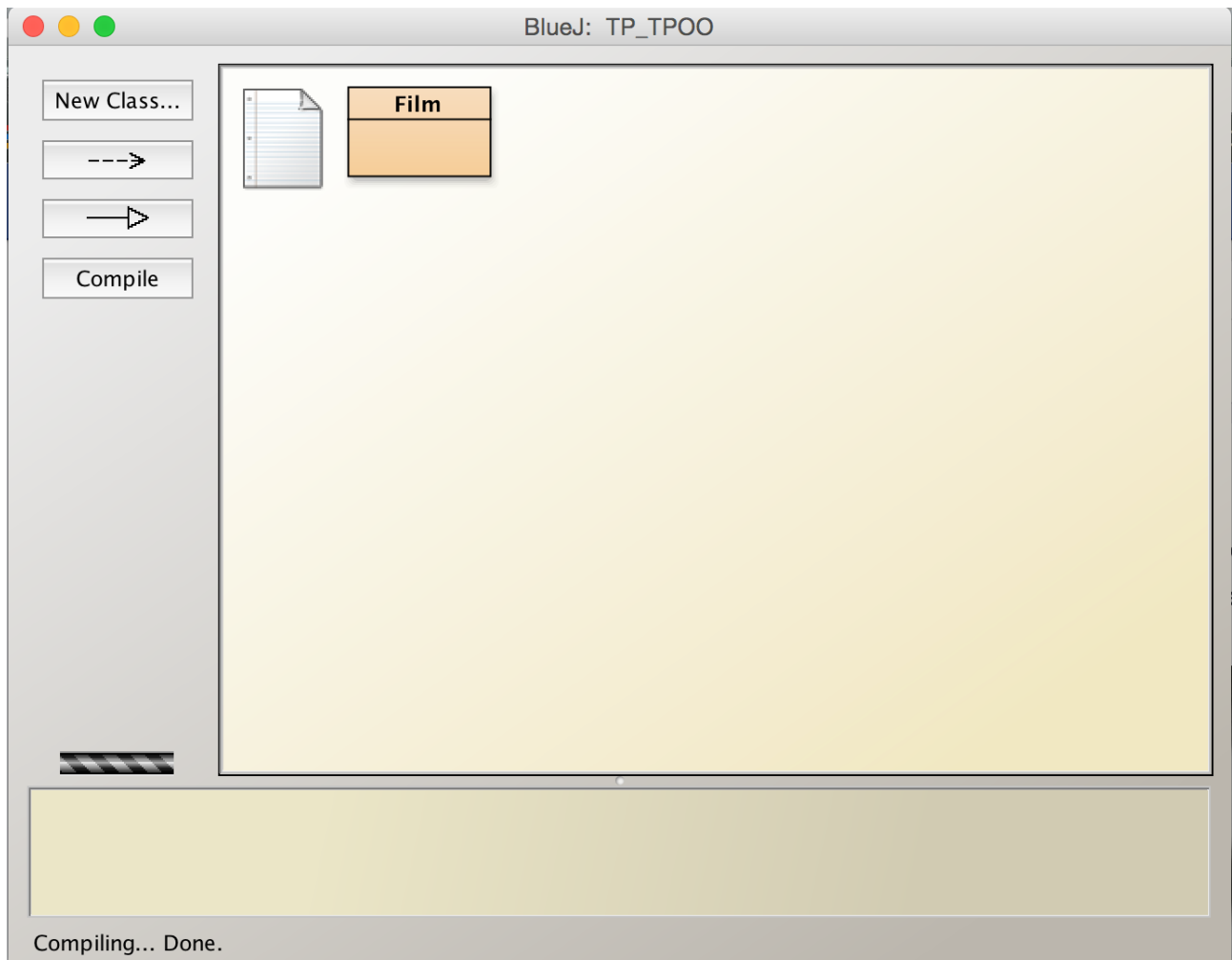
2) Pour démarrer



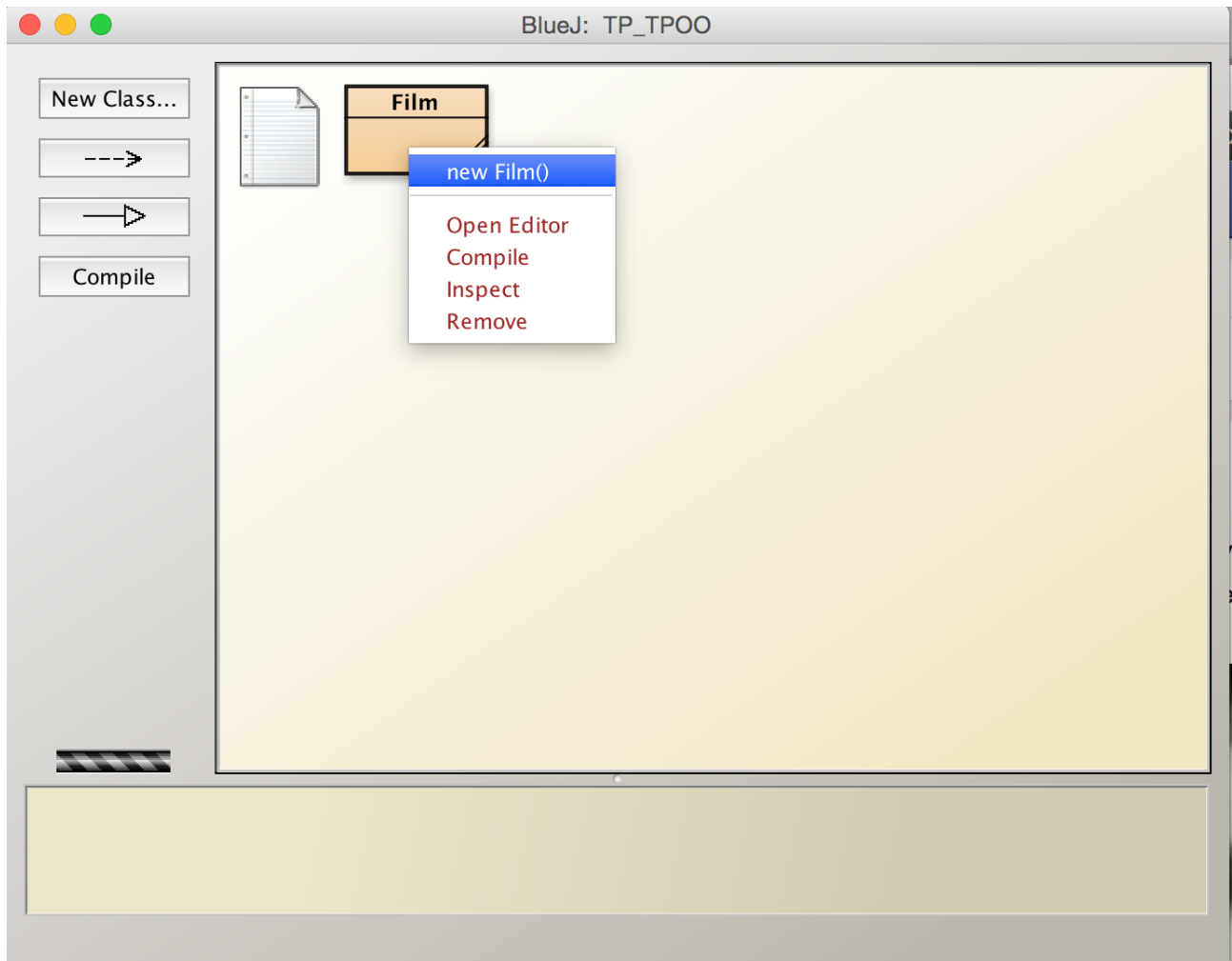
Pour créer un nouveau projet, cliquez sur project -> new Project dans la barre de tâches. Ensuite, une fenêtre apparaît, où il est possible d'indiquer l'emplacement et le nom du projet.



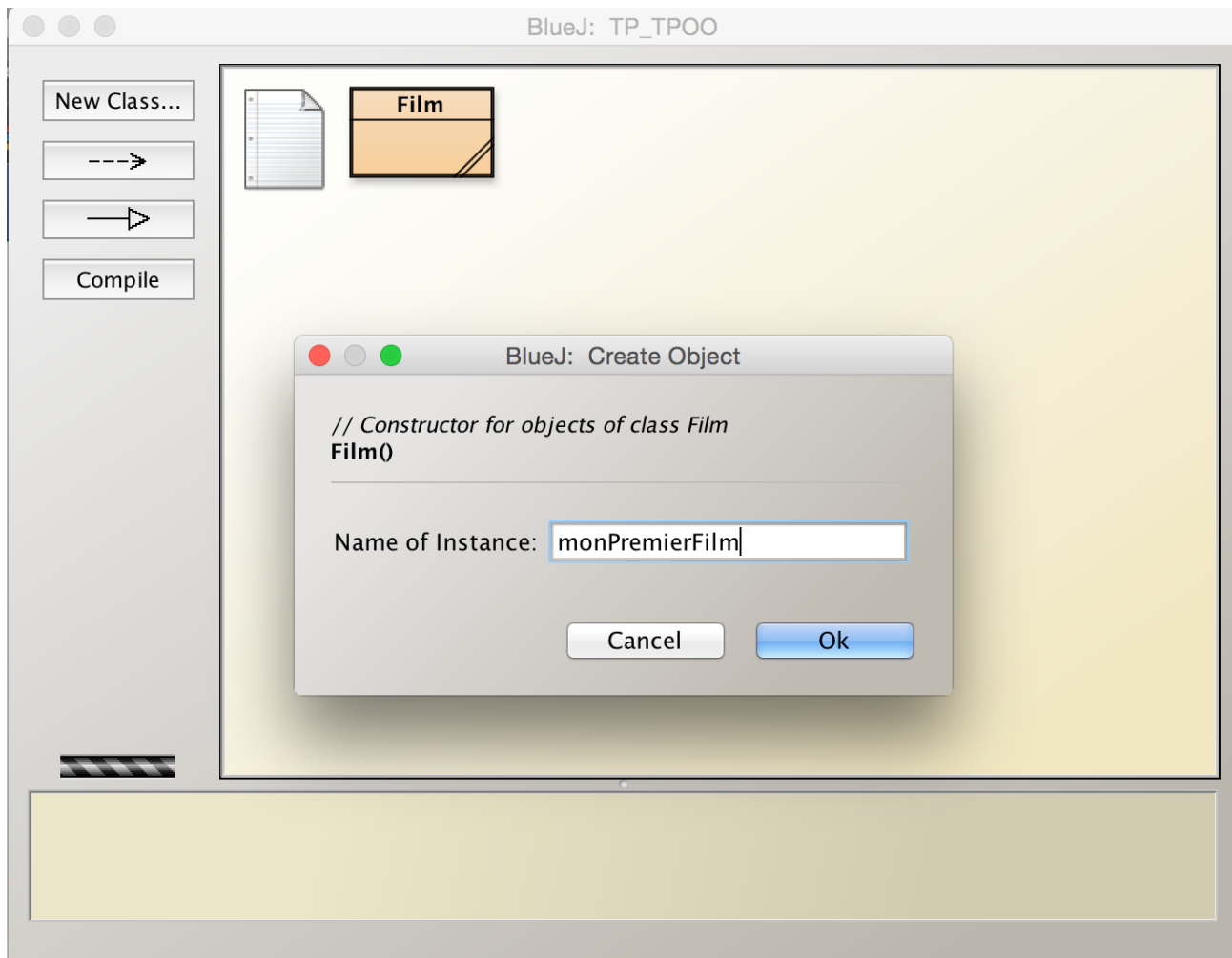
Pour créer une classe, cliquez sur le bouton new Class et indiquer le nom de la classe dans la fenêtre qui apparaît. Cliquez sur OK.



Cliquez sur le bouton Compile pour compiler la classe.
Compiling...Done s'affiche en bas de la fenêtre

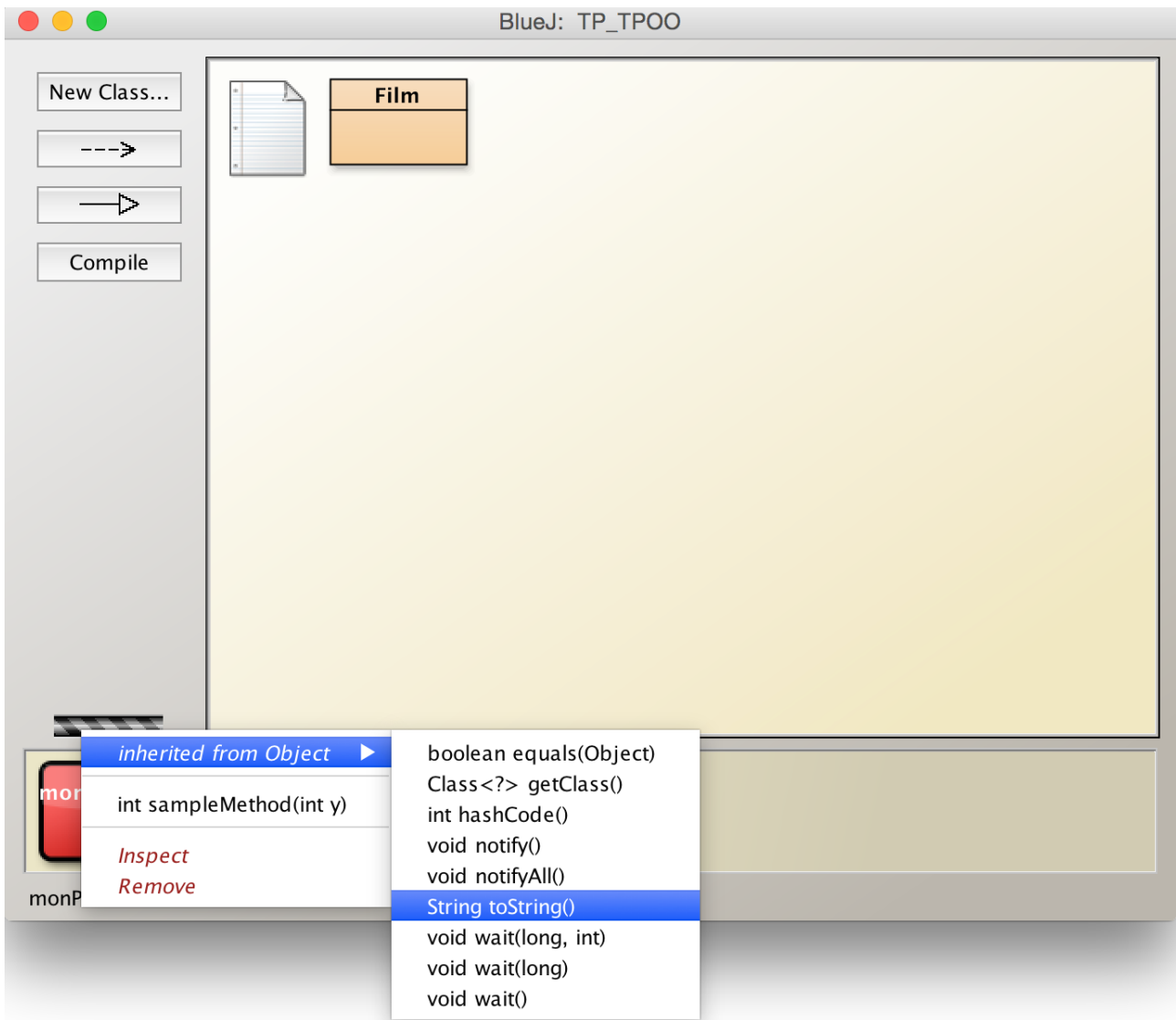


Pour instancier la classe, faites un clique droit sur la classe et cliquez sur new nomDeVotreClasse.

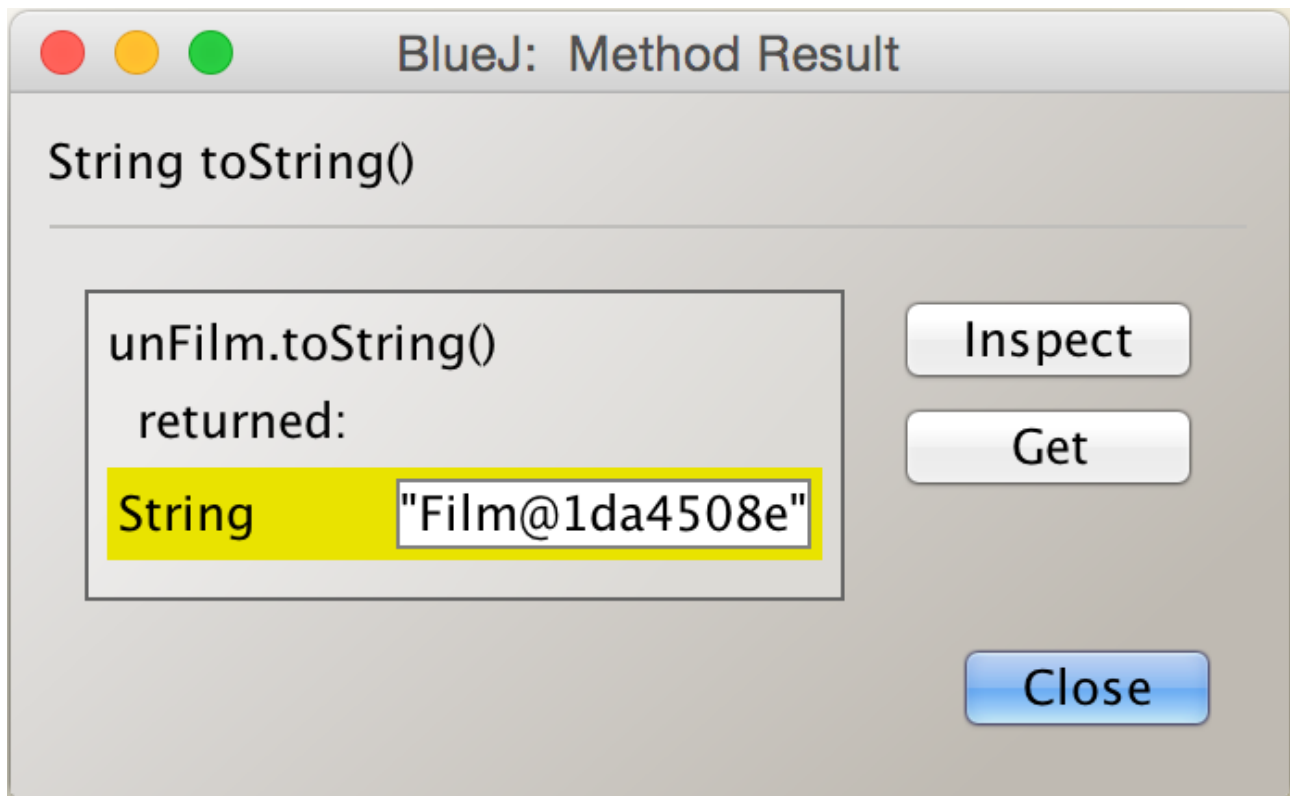


Indiquez ensuite le nom de l'objet dans la fenêtre qui apparaît.

3) Les premiers tests



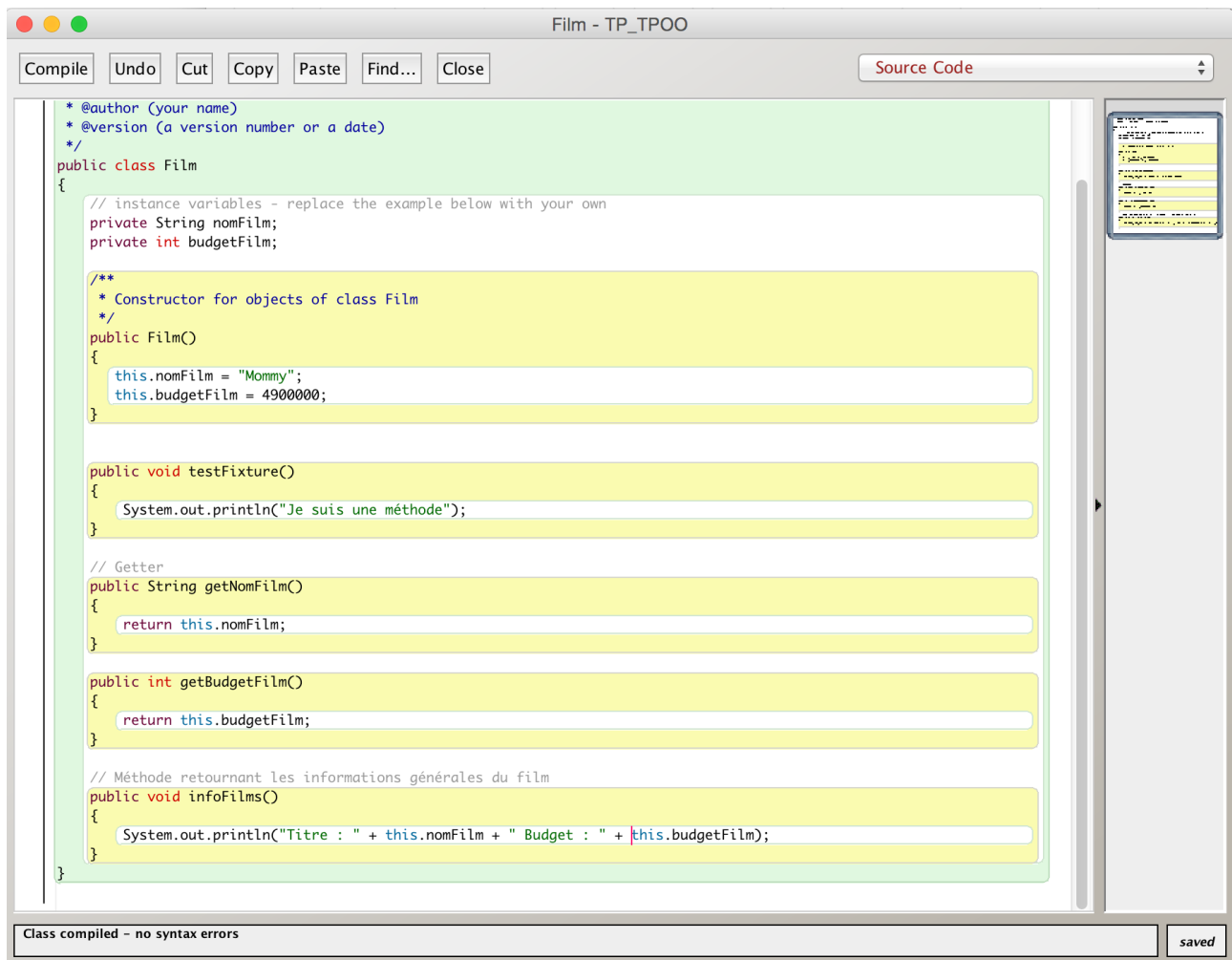
Pour créer une méthode de test, faites un clique droit sur l'objet, et sélectionnez votre méthode. Si cette dernière comprend un paramètre, il vous faudra le renseigner dans une autre fenêtre.



Voici un exemple de rendu pour la méthode String toString().

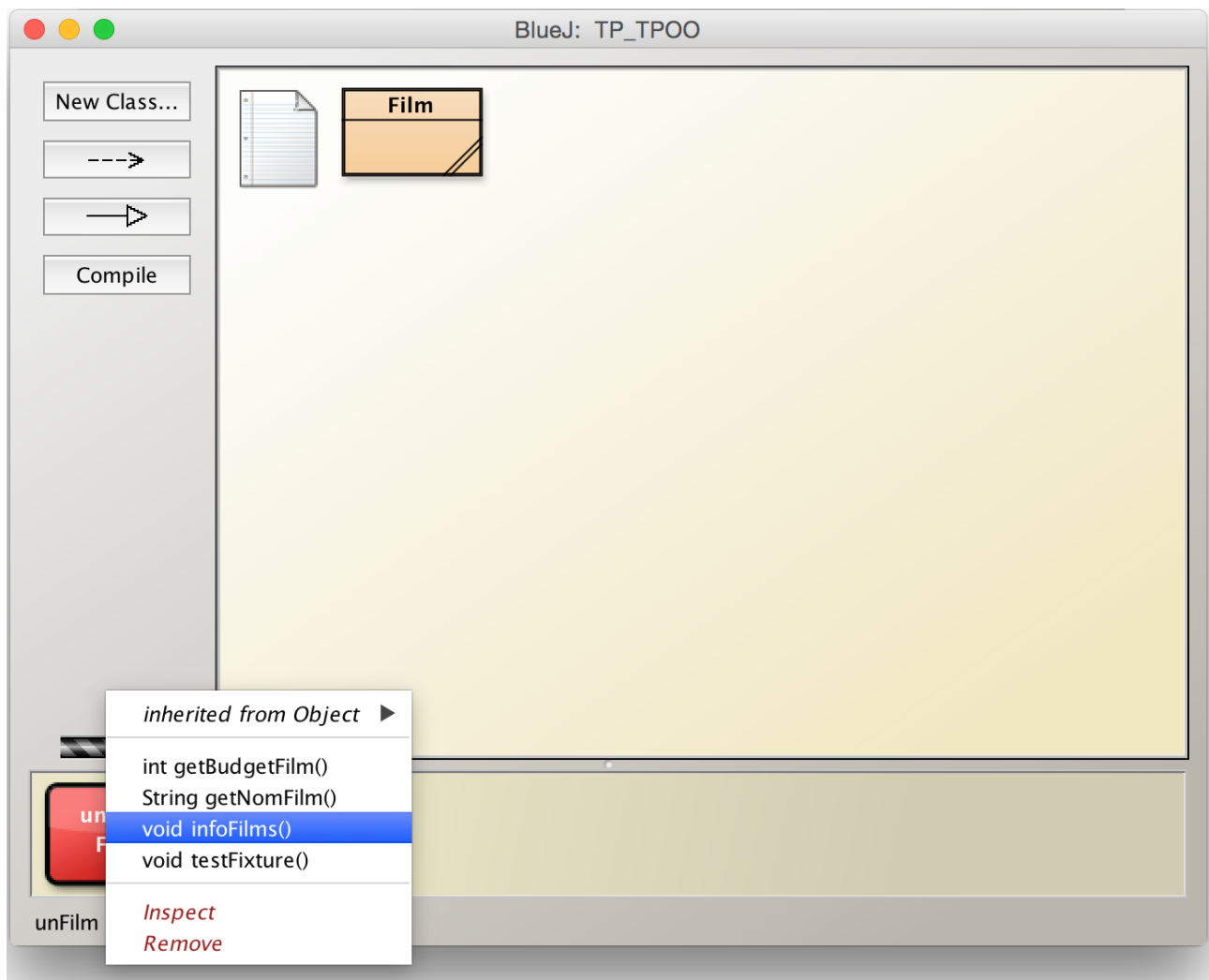


Pour modifier la classe, il suffit de double cliquer dessus.

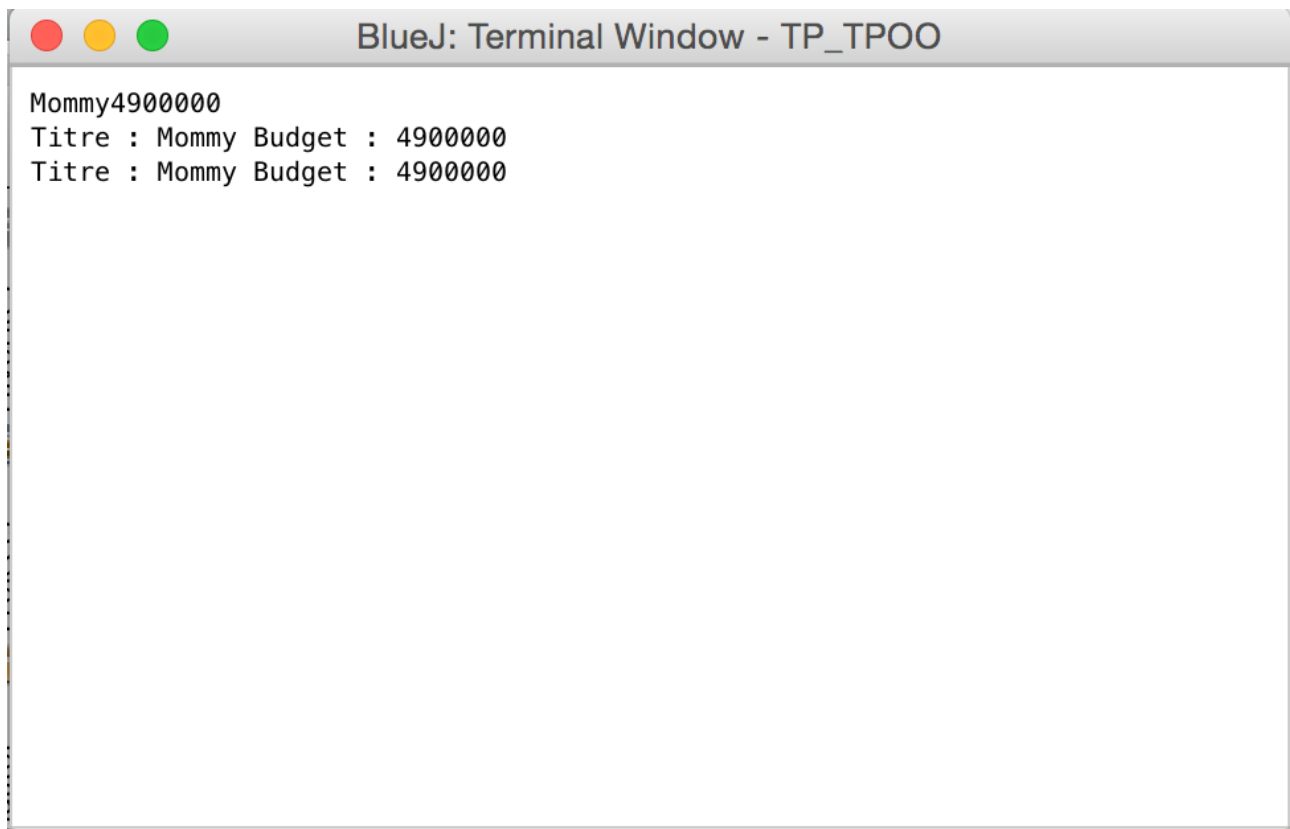


Ajoutez ensuite deux attributs ainsi que leur getters, et une méthode utilisant ces attributs.

Pour ré-instancier l'objet, on procède de la même façon que précédemment



Pour essayer notre méthode, faites un clique droit sur l'objet instancié et sélectionnez la méthode.



Le résultat de la méthode apparaît dans un terminal, car on appelle `System.out.println()`;

4) Et avec deux classes ?

```
import java.util.List;
import java.util.ArrayList;

public class Acteur
{
    private String prenomActeur;
    private String nomActeur;
    private List<Film> listeFilms;

    /**
     * Constructor for objects of class Acteur
     */
    public Acteur()
    {
        this.prenomActeur = "Anne";
        this.nomActeur = "Dorval";
        this.listeFilms = new ArrayList<Film>();
    }

    // Getter
    public String getNomActeur()
    {
        return this.nomActeur;
    }

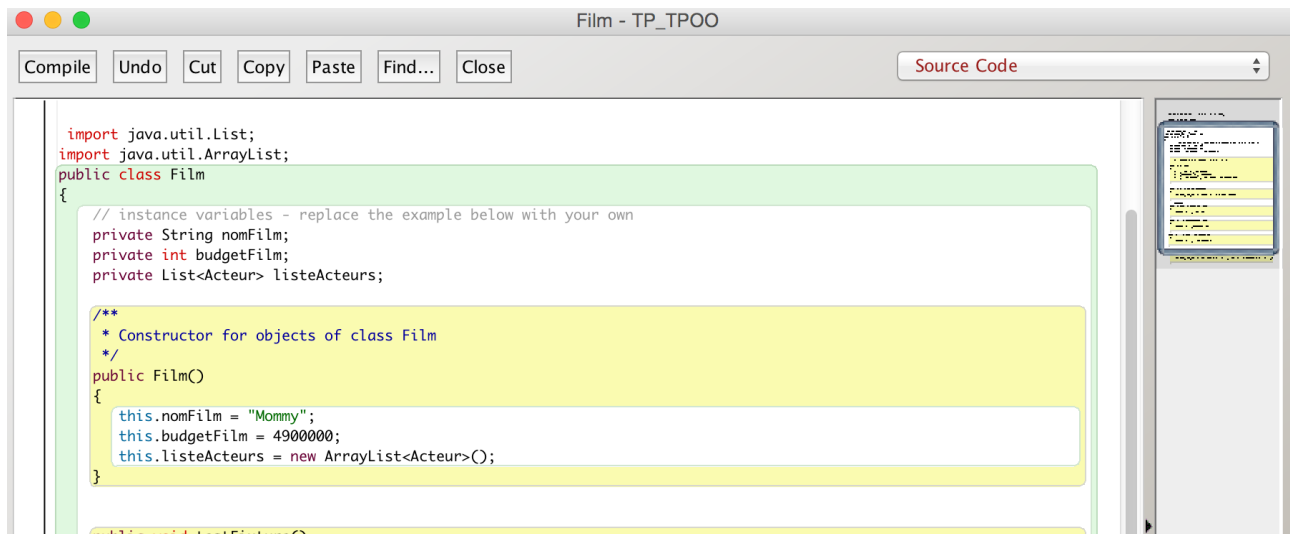
    public String getPrenomActeur()
    {
        return this.prenomActeur;
    }

    public List getListeFilms()
    {
        return this.listeFilms;
    }

    // Méthode pour ajouter un film à la liste de films
    public void ajouterFilm(Film unFilm)
    {
        this.listeFilms.add(unFilm);
        System.out.println("Le film " + unFilm.getNomFilm() + " a été ajouté ");
    }
}
```

Créez une seconde classe. Ici nous utiliserons la classe Acteur. Cette classe doit être reliée à votre première classe par une liste, ici listeFilms.

De la même façon, votre première classe doit comporter une liste de la seconde classe afin d'assurer la symétrie entre les objets.



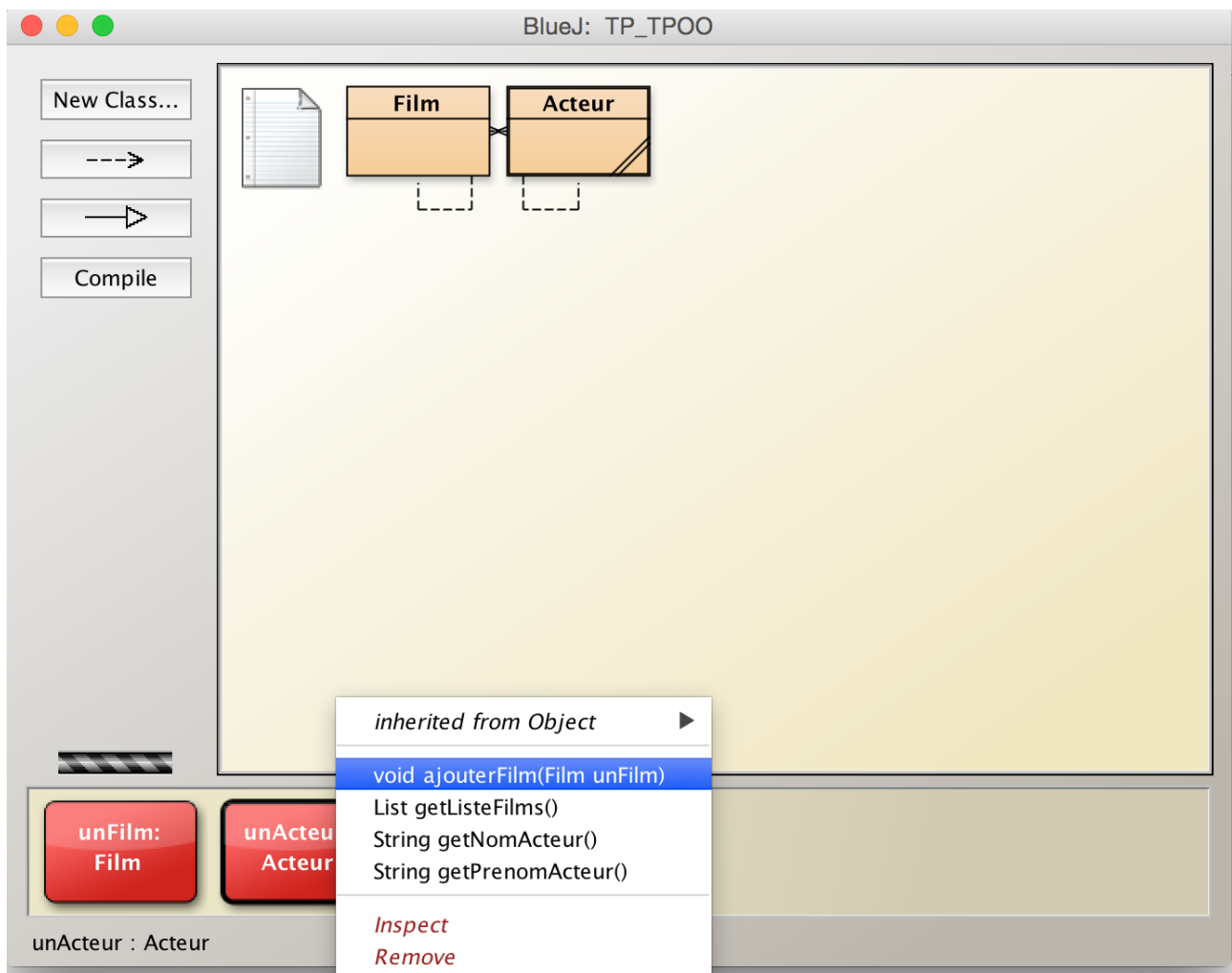
```
// Méthode pour ajouter un film à la liste de films
public void ajouterFilm(Film unFilm)
{
    this.listeFilms.add(unFilm);

    // On s'assure de la symétrie des objets
    unFilm.getListeActeurs().add(this);
    System.out.println("Le film a été ajouté");
    System.out.println("Liste des films :");

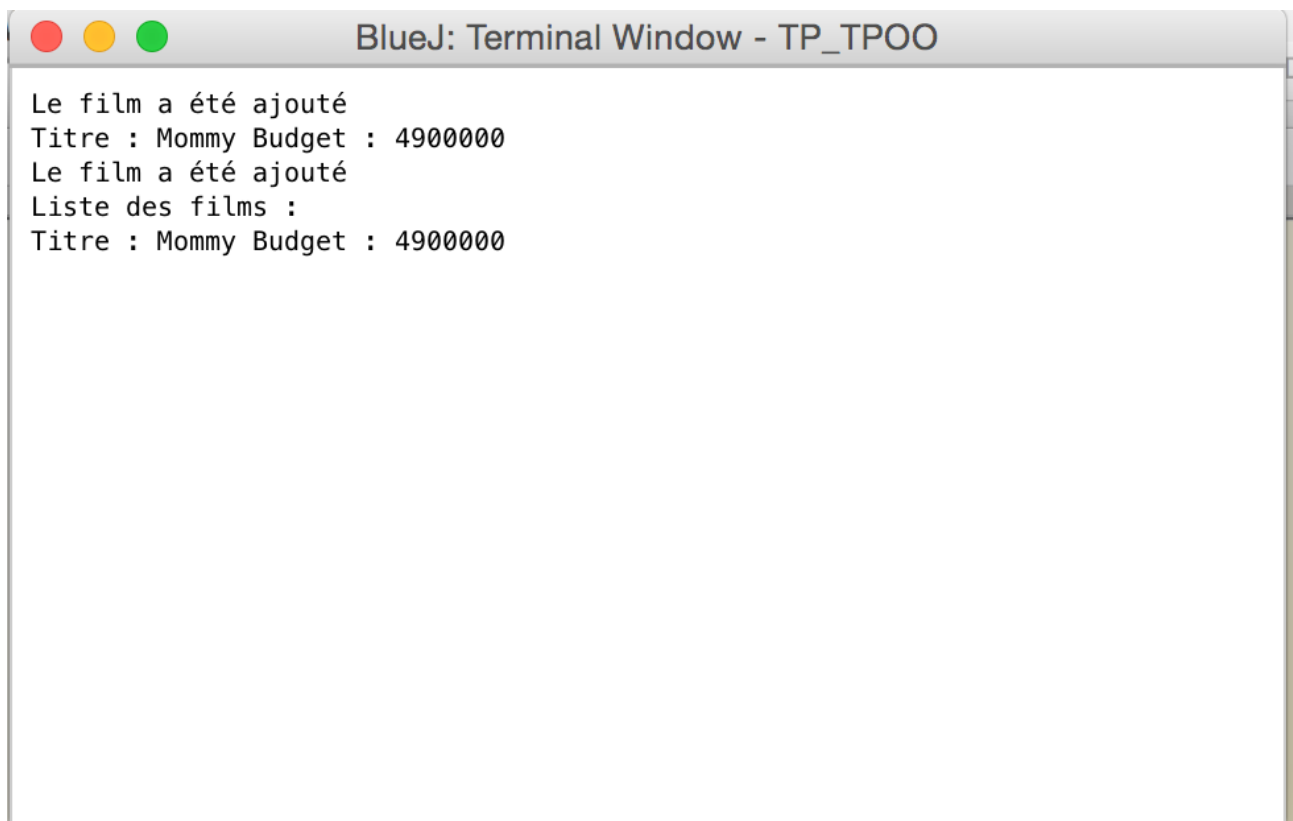
    // On affiche tous les films de l'acteur
    for (Film leFilm : this.listeFilms)
    {
        leFilm.infoFilms();
    }
}
```

Ajoutez ensuite une méthode utilisant la méthode de votre première classe dans cette nouvelle classe.

Pour instancier les classes, le procédé est le même qu'auparavant.



On remarque sur le schéma que les classes sont liées.
Pour tester la nouvelle méthode, on l'exécute en faisant un clique droit sur l'objet Acteur.



On remarque alors que le résultat est le même que le résultat attendu.