

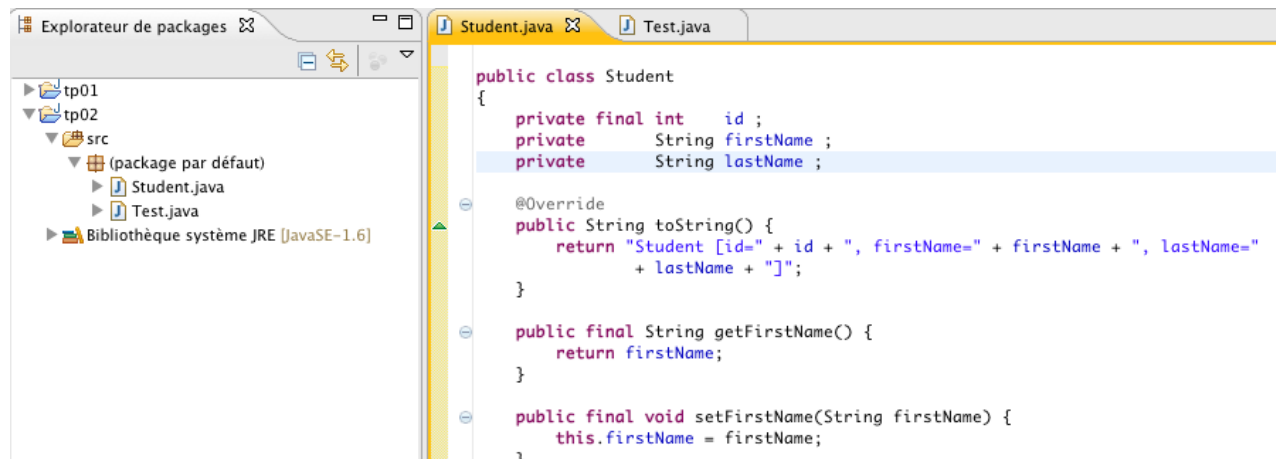
## TP Modéliser et trier une promotion d'étudiants

Pour ce TP, vous allez continuer à travailler dans un même projet. Vous allez :

- Rajouter une nouvelle classe nommée **Promotion** pour modéliser une promo d'élèves.
- Rajouter de nouvelles fonctionnalités à la classe **Student**.

### Partie 1 : Modéliser une promotion d'étudiants

Lancez Eclipse à partir des menus de votre environnement (e.g., Linux, Mac ou Windows). Retravaillez sur le projet tp02.



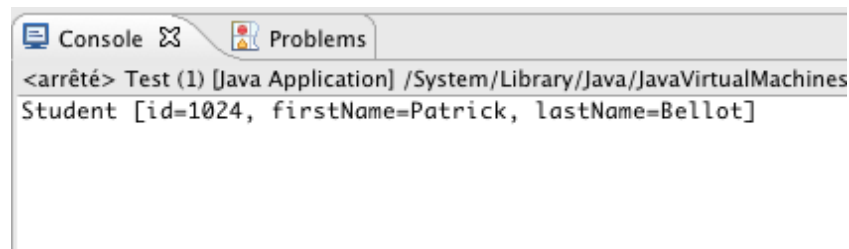
```
public class Student
{
    private final int id ;
    private String firstName ;
    private String lastName ;

    @Override
    public String toString() {
        return "Student [id=" + id + ", firstName=" + firstName + ", lastName="
            + lastName + "]";
    }

    public final String getFirstName() {
        return firstName;
    }

    public final void setFirstName(String firstName) {
        this.firstName = firstName;
    }
}
```

Lancez l'exécution pour vérifier que tout fonctionne encore bien. Vous devriez obtenir ceci dans la vue *Console* :



```
<arrêté> Test (1) [Java Application] /System/Library/Java/JavaVirtualMachines
Student [id=1024, firstName=Patrick, lastName=Bellot]
```

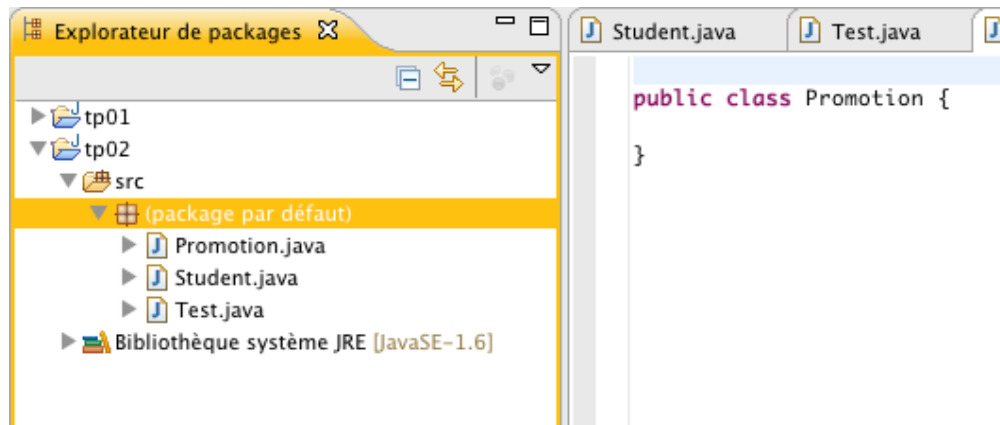
### Redéfinir l'affichage des élèves

Modifiez la méthode **toString()** de la classe pour qu'elle renvoie la chaîne de caractères ainsi formée :

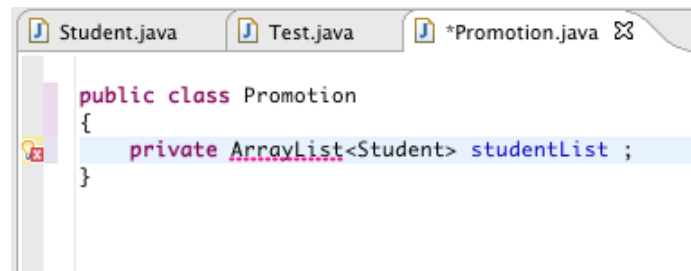
*<Prénom Nom (identifiant)>*

Exécutez puis vérifiez votre travail [ici](#).

Par clic droit sur le package par défaut, puis *Nouveau* puis *Classe*, créez une classe nommée **Promotion**.

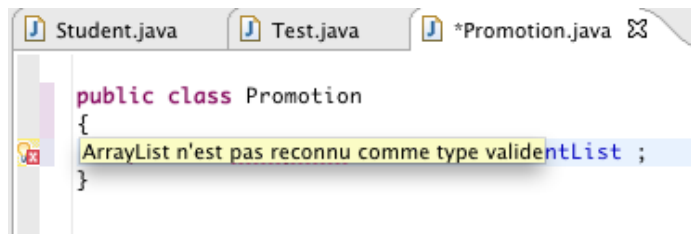


Une promotion étant une liste d'élèves, déclarez dans la classe **Promotion** un attribut **studentList** de type **ArrayList<Student>**. N'oubliez pas le qualificateur **private**. Vous devriez obtenir ceci :



### Déclaration d'importation

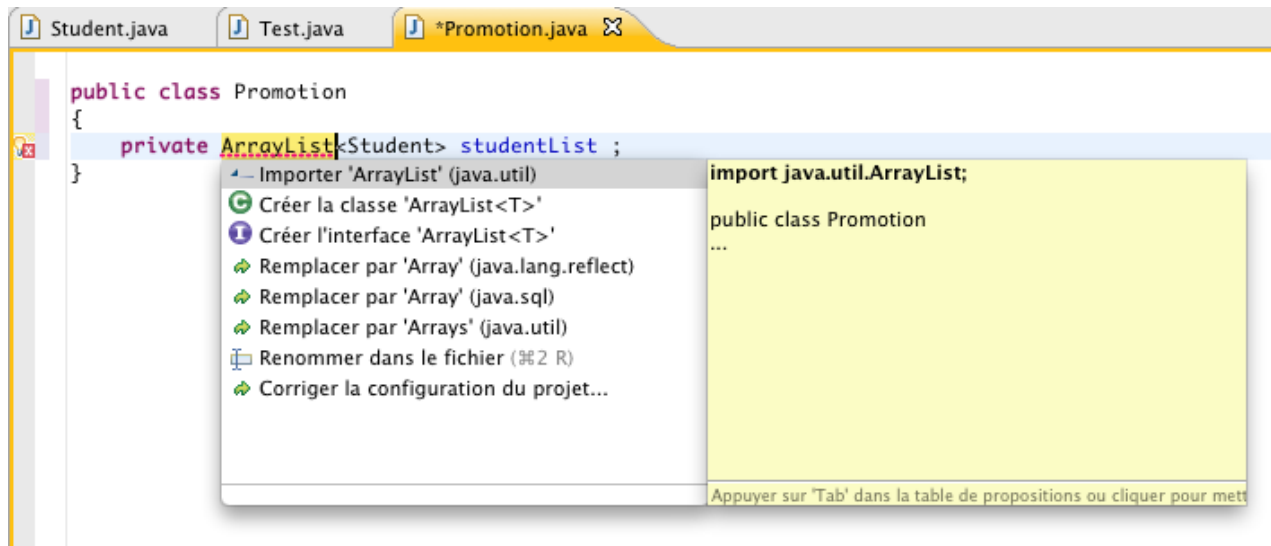
Une marque rouge indiquant une erreur est apparue dans la marge gauche de l'éditeur de code Java. En survolant ce marqueur avec la souris, l'erreur s'affiche :



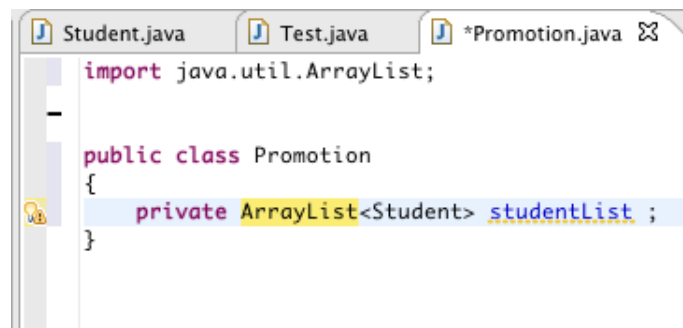
En effet, la classe **ArrayList** n'est pas connue par défaut. Il faut alors, tel que vu en cours, importer cette classe grâce à la déclaration :

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

Cette déclaration devra être placée au début du fichier de la classe. Ici encore, Eclipse vous aide. En cliquant sur la marque rouge d'erreur, l'IDE vous expose différentes solutions possibles :



La première solution est bien sûr la bonne puisque nous voulons utiliser la classe **ArrayList** fournie avec Java. Déplacez la souris et cliquer sur *Importer ArrayList*. La déclaration d'importation est alors créée par Eclipse.



En passant la souris sur la marque jaune-orange qui est apparue, vous verrez un avertissement que nous avons déjà rencontré.

### Écrire le constructeur

Comme nous l'avons vu en cours, par défaut l'attribut **studentList** est initialisé avec la valeur **null**. Nous avons également vu qu'il faut spécifier un constructeur afin d'initialiser les attributs avec des valeurs convenables. Consultez la Javadoc de la classe **ArrayList** sur le Web pour connaître les constructeurs de cette classe. Écrire un constructeur pour la classe **Promotion**. Contrôlez votre travail [ici](#).

On désire maintenant écrire une méthode dans la classe **Promotion** ayant la signature :

```
public int addStudent(String firstName, String lastName)
```

La fonction de cette méthode sera d'ajouter un élève de prénom **firstName** et de nom **lastName** à la promotion. Le résultat retourné par cette méthode sera l'identifiant numérique de l'élève.

### Trouver un identifiant libre

Pour cela, nous allons d'abord écrire une méthode de la classe **Promotion** ayant la signature :

```
public int newId()
```

Cette méthode calculera un identifiant numérique unique et donc non utilisé dans la promotion. La méthode **newId()** devra parcourir la liste des élèves pour trouver l'identifiant numérique le plus grand parmi ceux utilisés. Puis elle ajoutera **1** pour obtenir un nouvel identifiant non utilisé. Si la liste des élèves est vide, la fonction devra retourner **0**.

Ecrivez la méthode **newId()** dans la classe **Promotion**. Contrôlez votre travail [ici](#).

### Ajouter un élève à la promotion

Écrivez la méthode :

```
public int addStudent(String firstName, String lastName)
```

Contrôlez votre travail [ici](#).

### Afficher les élèves d'une promotion

Écrivez une méthode de la classe **Promotion** ayant la signature :

```
public void printToConsole()
```

Cette méthode affichera la liste des élèves de la promotion à la *Console*. Contrôlez votre travail [ici](#).

### Testons nos méthodes

Dans la fonction main de la classe **Test**, ajoutez les instructions qui permettront de :

1. Créer un objet de la classe **Promotion**.
2. Ajouter une dizaine d'élèves dans cet objet.
3. Afficher la liste des élèves dans la *Console*.

Exécutez votre programme et vérifiez votre travail [ici](#).