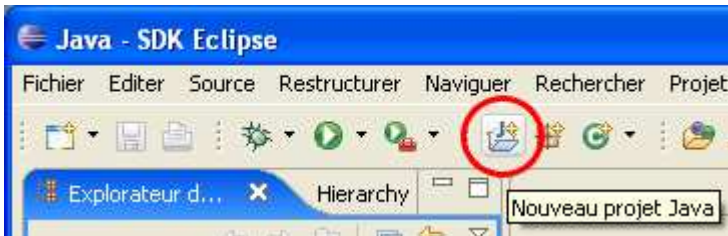


**Inspiré/Copié du tutoriel Eclipse :**  
<http://www.ecliptotale.com/articles/premierPas.html>

## - Création d'un projet Java

La première étape pour développer une application Java avec Eclipse est de créer un 'Projet Java'. L'ouverture de l'assistant se fait en utilisant le bouton correspondant dans la barre de boutons (ou *Fichier/Nouveau/Projet Java*):



La seule information nécessaire pour l'assistant est le nom du projet. Une fois le nom saisi, vérifiez que la JRE sélectionnée est bien la JRE6. Si tel n'est pas le cas, sélectionnez-la via la ligne « Utiliser un environnement JRE propre au projet ». Le bouton **Terminer** est activé, cliquez sur ce bouton pour demander la création du projet.



Après la création, le projet apparaît dans la vue 'Explorateur de projets' :



## - Création d'une classe

Pour ouvrir l'assistant de création de classes Java, faire afficher le menu contextuel du projet tests (**bouton droit** puis *Nouveau/Classe*).

Les informations attendues par cet assistant sont le nom de la classe et le nom de son package. Entrez ces informations puis cochez les cases indiquant à l'assistant de générer une méthode main et les commentaires.

The screenshot shows the 'Nouvelle classe Java' dialog box. The title bar reads 'Nouvelle classe Java'. Below the title bar, it says 'Classe Java' and 'Créez une classe Java'. The dialog is divided into several sections:

- Dossier source :** tests/src (with a 'Parcourir...' button)
- Package :** principal (with a 'Parcourir...' button)
- Type englobant :** (empty, with a 'Parcourir...' button)
- Nom :** VO
- Modificateurs :**  public,  Valeur par défaut,  private,  protected,  abstract,  final,  static
- Superclasse :** java.lang.Object (with a 'Parcourir...' button)
- Interfaces :** (empty, with an 'Ajouter...' button and a 'Supprimer' button)
- Quels raccords de méthode voulez-vous créer ?**
  - public static void main(String[] args)
  - Constructeurs de la superclasse
  - Méthodes abstraites héritées
- Voulez-vous ajouter des commentaires ? (Configurez les modèles et la valeur par défaut ici)**
  - Générer les commentaires

Après avoir cliqué sur le bouton **Terminer**, la classe est créée et ouverte en édition :

The screenshot shows the Eclipse IDE with the 'VO.java' file open in the editor. The package explorer on the left shows the project structure: SAPP > test > tests > src > principal > VO.java. The editor displays the following code:

```
package principal;

/**
 * @author CEDRIC HERPSON
 */
public class VO {


    /**
     * @param args
     */
    public static void main(String[] args) {
        // TODO Auto-generated method stub
    }
}
```

## - Exécution

Dans l'éditeur, complétez la méthode 'main' avec les lignes de code suivante :

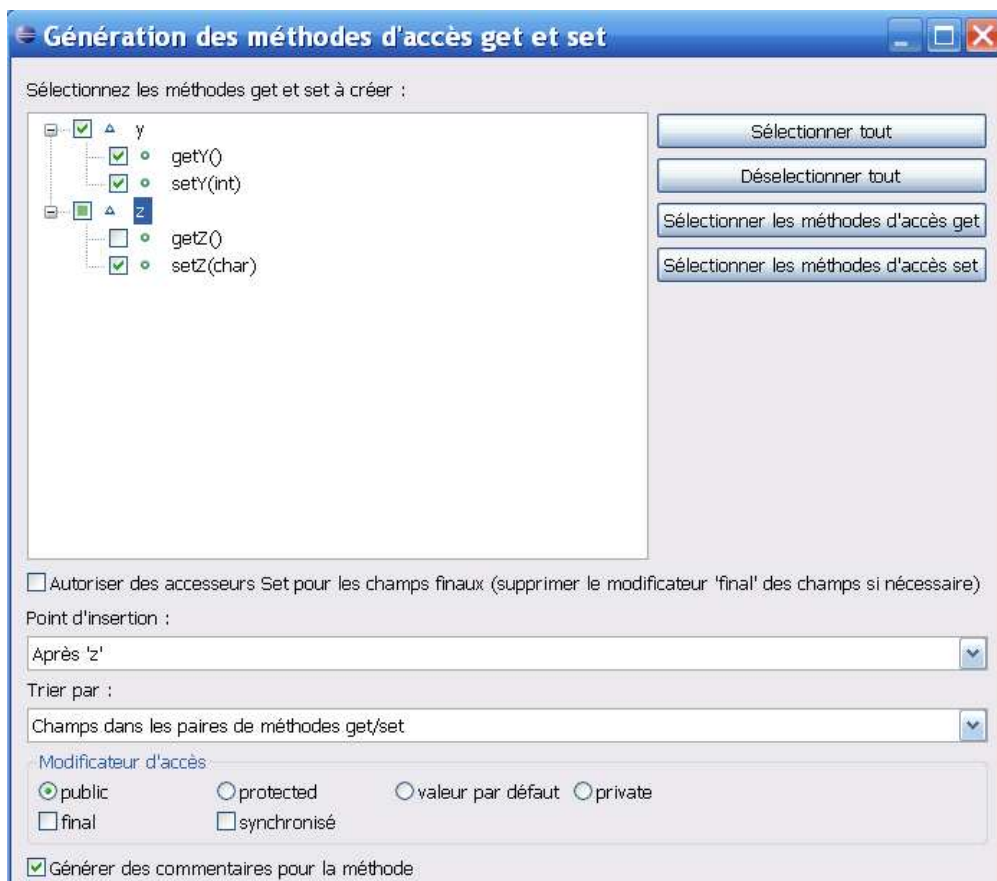
```
String x=" ";
String y="103683";
x=y;
x="longue";
System.out.println("La route est "+x+" mais la voie est libre");
```

Une fois le code saisi, demandez l'enregistrement (Ctrl+S ou menu Fichier). La classe est enregistrée **et précompilée**.

Pour demander l'exécution d'une classe, dépliez le menu associé au bouton  et sélectionnez *Exécuter en tant que.../Application Java*. La vue Console affiche le résultat

## - Génération automatique de code

Dans l'éditeur, ajoutez une variable globale de type 'int' t et une variable globale privée de type 'char' z à la classe. Ensuite, faites **bouton droit** puis *source/génération des méthodes d'accès get et set* .



### Raccourcis fondamentaux :

CTRL+A : Sélection de tous les éléments de la fenêtre active (non spécifique à Eclipse)

CTRL+I : indentation automatique de la sélection

CTRL+S : sauvegarde du projet (non spécifique à Eclipse)

CTRL+ESPACE : complétion automatique

Bonus : « syso » puis CTRL+ESPACE --> System.out.println() ;

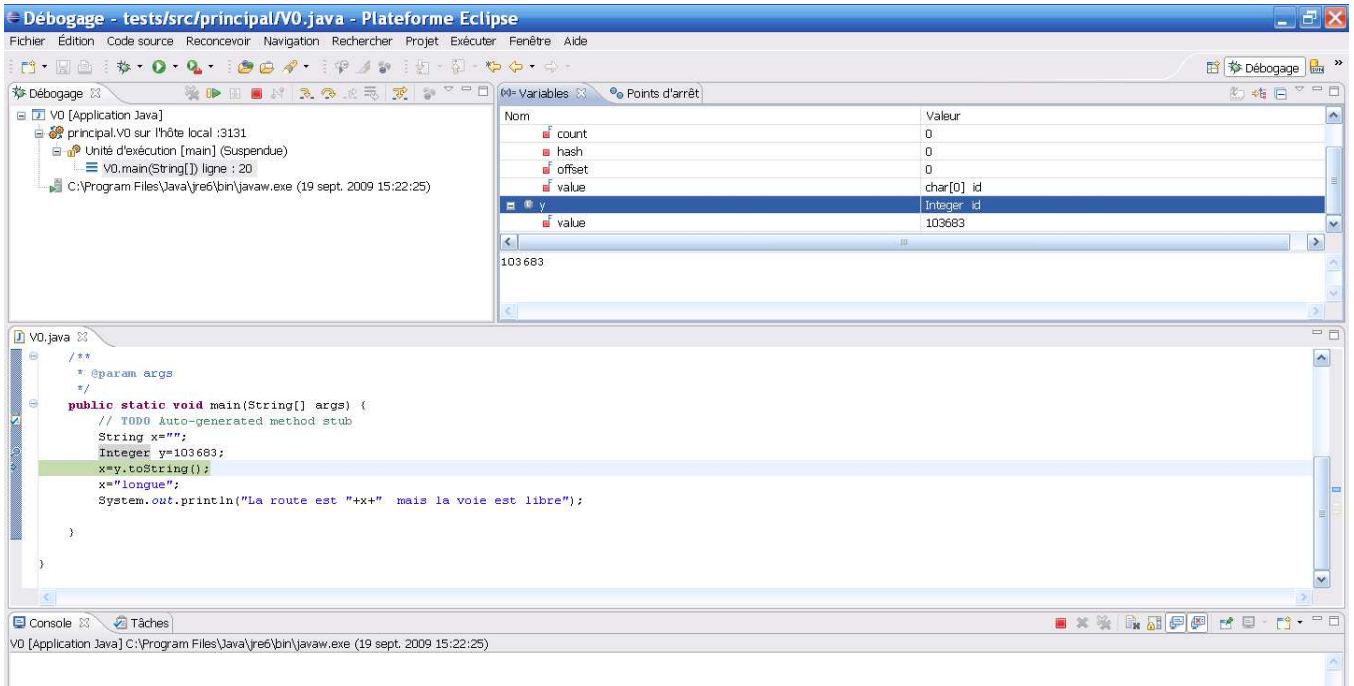
## - Débogage




Placer un point d'arrêt en double-cliquant dans la marge ou en utilisant le menu contextuel dans la marge (**bouton droit** et *Ajouter/Supprimer un point d'arrêt*) en face de ligne `String y="103683"` ;



Lancer l'application en mode Débogage via le bouton:

Lorsque que le point d'arrêt est rencontré, Eclipse propose d'afficher le débogueur.



- Les boutons  permettent d'avancer en pas à pas, le premier en rentrant dans le code de la méthode appelée, le second en passant directement à l'instruction suivante.
- Le bouton  permet de reprendre l'exécution normale (jusqu'au prochain point d'arrêt).
- Le bouton  permet de stopper l'exécution du programme.

L'exécution en mode pas à pas permet de suivre facilement l'évolution des variables.