

# Programmation avec ECLIPSE

Claude Duvallet

Université du Havre  
UFR Sciences et Techniques  
25 rue Philippe Lebon - BP 540  
76058 LE HAVRE CEDEX  
Claude.Duvallet@gmail.com  
<http://litis.univ-lehavre.fr/~duvallet/>

Session de formation au MTP

## Présentation de la plateforme Eclipse

- 1 Objectifs
- 2 Création d'un projet JAVA avec Eclipse
- 3 Création de la JavaDOC

## Introduction à Eclipse

- Prévues pour fournir une plateforme ouverte de développement :
  - Fonctionne sur un grand nombre de systèmes d'exploitation.
  - Interface graphique très performante et facilitant le développement d'applications.
- Indépendance du langage de programmation :
  - Permet sans restriction l'utilisation plusieurs types de contenus.
  - HTML, Java, C, JSP, EJB, XML, GIF,...
- Facilite l'intégration de nouveaux outils :
  - Au niveau de l'interface et en profondeur.
  - Ajout de nouveaux outils pour les produits installés.
- Attire une grande communauté de développeurs :
  - Y compris des éditeurs de logiciels indépendants.
  - Capitalise la popularité de Java pour l'écriture de nouveaux Outils.

## Genèse d'Eclipse

- Eclipse créé par l'OTI et les équipes d'IBM chargé pour les produits IDE :
  - IBM VisualAge / Smalltalk (Smalltalk IDE)
  - VisualAge IBM / Java (Java IDE)
  - VisualAge IBM / Micro Edition (Java IDE)
- Initialement composé de 40 développeurs à plein temps.
- Des équipes dispersées géographiquement de développement.
  - Ottawa OTI, Minneapolis OTI, Zurich OTI, IBM Toronto, OTI Raleigh, RTP, IBM Saint-Nazaire (France).
- Effort transition en projet open source
  - IBM a donnée le code de base d'Eclipse : Plate-forme, JDT, PDE.

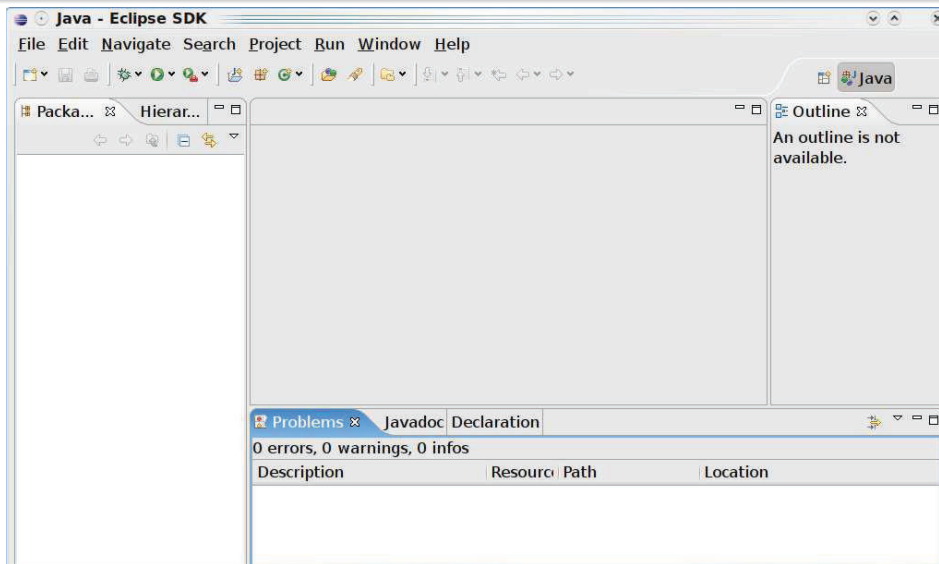
## Historique

- 1999
  - Mars Début des travaux sur Eclipse au sein de OTI/IBM.
- 2000
  - Mars Premiers composants Eclipse.
- 2001
  - Mars Ouverture du site <http://www.eclipsecorner.org/>.
  - Juin Version Eclipse 0.9.
  - Octobre Version Eclipse 1.0.
  - Novembre IBM fait don du code source d'eclipse.  
Ouverture du site <http://www.eclipse.org/>
- 2002
  - Juin Eclipse 2.0
  - Septembre Eclipse 2.0.1
  - Novembre Eclipse 2.0.2
- 2003
  - Mars Eclipse 2.1

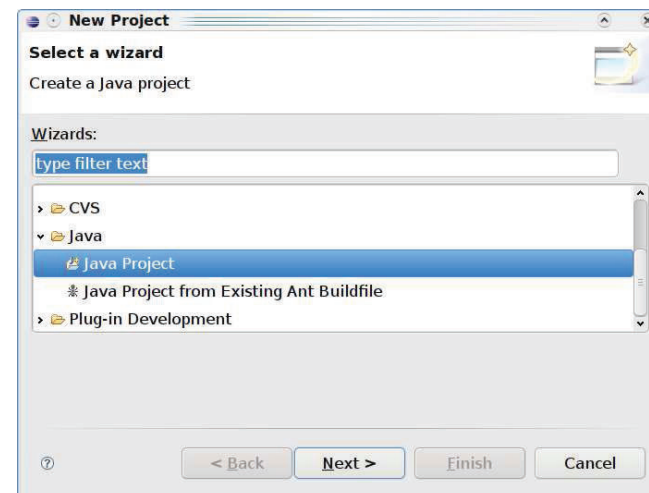
## Eclipse : une plateforme de développement JAVA et plus

- Il fournit un ensemble d'outils permettant de créer facilement des classes.
- Possibilité de créer des classes JAVA comportant :
  - une méthode principale "main".
  - des méthodes hérités.
  - des accesseurs "get" et "set".
- Existence de nombreux plugins permettant de développer dans d'autres langages que JAVA.

## Présentation de la plateforme Eclipse



## Création d'un projet JAVA



## Nommer le projet et finaliser sa création

**Create a Java project**  
Create a Java project in the workspace or in an external location.

Project name:

**Contents**

- Create new project in workspace
- Create project from existing source

Directory:

**JRE**

- Use default JRE (Currently 'jdk1.6.0\_19')
- Use a project specific JRE:

**Project layout**

- Use project folder as root for sources and class files
- Create separate source and output folders

## Création d'une nouvelle classe principale

**Java Class**  
The use of the default package is discouraged.

Source folder:

Package:

Enclosing type:

Name:

Modifiers:  public  default  private  protected  
 abstract  final  static

Superclass:

Interfaces:

Which method stubs would you like to create?

- public static void main(String[] args)
- Constructors from superclass
- Inherited abstract methods

Do you want to add comments as configured in the properties of the current project?

## Résultats de la génération

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

## Lancement de l'application

```
public class MaClasse {  
    /**  
     * @param args  
     */  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        System.out.println ("Bonjour à tous !");  
    }  
}
```

## Résultat de l'exécution

The screenshot shows the Eclipse IDE with the 'MaClasse.java' file open. The code in the editor is:

```
public class MaClasse {  
    /**  
     * @param args  
     */  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        System.out.println ("Bonjour à tous !");  
    }  
}
```

The Outline view on the right shows the class structure with the 'main(String[] args)' method selected. The Console view at the bottom displays the output:

```
<terminated> MaClasse [Java Application] /usr/local/jdk1.6.0_19/bin/java (9 avr. 10 13:2  
Bonjour à tous !
```

The status bar at the bottom indicates 'Claude Duvallet — 13/14 ECLIPSE'.

## Création de la JavaDOC

The screenshot shows the Eclipse IDE with the 'MaClasse.java' file open. The code in the editor is:

```
/**  
 * Description de la classe  
 * @author Claude Duvallet  
 */  
public class MaClasse {  
    /**  
     * Il s'agit de la méthode principale.  
     * @param args paramètre permettant de récupérer les  
     * arguments saisis en ligne de commande.  
     */  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        System.out.println ("Bonjour à tous !");  
    }  
}
```

The Outline view on the right shows the class structure with the 'main(String[] args)' method selected. The Javadoc view at the bottom displays the generated documentation:

```
void MaClasse.main(String[] args)  
  
Il s'agit de la méthode principale.  
Parameters:  
args paramètre permettant de récupérer les arguments saisis en ligne de commande.
```

The status bar at the bottom indicates 'Claude Duvallet — 14/14 ECLIPSE'.