Programmation Orientée Objet
Programmation JAVA

FISE 3A ICy / FISA 3A Informatique

TP Initiation à Java et Eclipse

Mohamed Amine BOUDIA
UPHF, CNRS, UMR 8201 - LAMIH, F-59313 Valenciennes, France
Email : mohamedamine.boudia@uphf.fr

LAMIH UMR CNRS 8201 | Université Polytechnique HAUTS-DE-FRANCE | cnrs

Objectif Zéro Papier

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES HAUTS-DE-FRANCE

JDK et JVM

- Nous aurons besoin d'installer le JDK (Java Development Kit), c'est l'environnement dans lequel un code Java est compilé afin JVM (Java Virtual Machine) l'exécute.
- Le JDK est constituer de plusieurs outils, principalement :
  - le compilateur Java (javac),
  - L'exécution (java)
  - l'archiveur (jar),
  - le générateur de documentation (javadoc)
  - et le débogueur (jdb).
- Il contient également l'environnement d'exécution Java (JRE pour Java Runtime Environment).

IDE (Integrated Development Environment)

- Nous pouvons utiliser ses outils directement sur console
  - pour compiler javac.exe
  - pour exécuter java.exe
- Nous pouvons utiliser un IDE (Integrated Development Environment) qui est un programme regroupant un ensemble d'outils pour le développement de logiciels.
- La collaboration du JDK et du l'IDE facilite le développement (automatise les étapes de compilation et exécution par exemple), grâce à une interface graphique GUI dédiée.
- Il existe plusieurs IDE : Netbeans, Eclipse, JDeveloper, JBuilder….

Eclipse

- Eclipse est un environnement de développement intégré libre extensible, universel et polyvalent, permettant de créer des projets de développement mettant en œuvre n'importe quel langage de programmation.
- Eclipse IDE est principalement écrit en Java (à l'aide de la bibliothèque graphique SWT, d'IBM), et ce langage, grâce à des bibliothèques spécifiques, est également utilisé pour écrire des extensions.
- La spécificité d'Eclipse IDE (Integrated Development Environement) vient du fait de son architecture totalement développée autour de la notion de plugin (en conformité avec la norme OSGi)

Création d’un projet sur Eclipse

- Ouvrez l'IDE Eclipse, on vous demandera de choisir l'espace de travail Workspace dans lequel vous allez trouver vos projets. (Si vous n'avez pas encore créé d'espace de travail, Eclipse le créera pour vous).
- Créez un nouveau projet Java. Pour cela, appuyez sur File → New → Java Project. La fenêtre suivante apparaîtra:
  - Tapez comme indiqué le nom de votre projet. Conservez les réglages par défaut, et appuyez sur Finish.
- Dans l'espace de travail, sous l'onglet Packages, vous verrez que le projet est créé, et qu'il contient déjà un répertoire src (qui doit contenir tous les fichiers source que vous créez), et JRE System Library, qui est utilisée pour compiler votre code.

Création de package et classe

- Pour créer un package sous le répertoire src, cliquer sur celui-ci, puis sur l'icône , ou clic-droit sur src, et choisir New -> Package. Choisissez un nom pour le package.
- Pour créer une classe :
  - Cliquer sur le package qui doit contenir la classe puis sur l'icône , ou clic-droit sur le package, et choisir New -> Class. Dans la fenêtre qui apparaît, choisissez le nom de la classe.
  - Si la classe n'est pas définie dans un package (ce qui est déconseillé), refaites l'opération ci-dessus à partir du répertoire src.
  - Vous pouvez générer automatiquement la méthode main en cochant la case public static void main(String[] args).
- La classe générée apparaîtra sous le package que vous avez choisi. Double-cliquez dessus pour modifier son code dans la partie édition. Vous verrez qu'un squelette de la classe vous est proposé. Vous n'aurez qu'à terminer le reste du code.
- Avec Eclipse, vous n'avez pas besoin de compiler explicitement votre code : la compilation se fait en temps réel. De plus, les erreurs de syntaxe seront affichées pendant l'écriture du code, avec des propositions de corrections.

Nouveau projet : Helloworld

- Helloworld – Version simplifiée
  - Créer un nouveau projet Helloworld comme indiqué dans la partie précédente.
  - Créer un package nommé helloPack, contenant une classe Helloworld qui contient une méthode main.
  - Dans la méthode main, écrire "System.out.println("Hello World!");"
  - Exécuter votre programme en cliquant directement sur l'icône 
  - L'affichage apparaîtra dans la partie inférieure, sous l'onglet Console.

Nouveau projet : Helloworld

- Helloworld – Ajout d'arguments
  - Créer dans le même projet Helloworld, un nouveau package nommé argsPack, contenant une classe Helloworld avec une méthode main.
  - Dans le code de la méthode main, écrire : "System.out.println("Hello "+args[0]+"!"); "
  - Pour définir des arguments à la classe, cliquer sur la flèche à côté de l'icône d'exécution et sélectionner Run Configurations, ou cliquer sur la classe que vous voulez exécuter, et aller à Run -> Run Configurations
  - Dans la partie de gauche, sélectionner Java Application, puis cliquer sur l'icône (en haut, à gauche), pour ajouter une nouvelle configuration. Vous verrez que votre classe Helloworld a été ajoutée sous Java Application.
  - Sélectionner l'onglet Arguments (ci-dessus encadré en rouge) et, dans le cadre Program Arguments, tapez simplement votre nom.
  - Cliquer ensuite sur Run. Vous verrez dans la console l'affichage "Hello votre_nom!".