EDI, Environnement de Développement Intégré

Anglais : IDE, Integrated Development Environment

Un EDI est un logiciel utilisé dans le cadre des développements informatiques, regroupant :

- Un éditeur de texte pour la saisie des programmes sources, disposant d'une coloration syntaxique et de la '*complétion*' de code contextuelle ;
- Un compilateur qui traduit le code source en code exécutable par la JVM (*JVM*, *Java Virtual Machine*)
- Un débogueur (*anglais : debugger*) : outil d'aide au diagnostic en cas d'erreur d'exécution d'un programme
- Un système d'aide
- Un certain nombre d'outils dédié à la productivité du développement informatique (assistants)

Les EDI gèrent les développements d'applications sous forme de **projets** : un projet comporte plusieurs fichiers, parmi lesquels les fichiers sources, des fichiers de configuration utiles à l'EDI (*paramètres des options de configuration de l'interface graphique de l'EDI, etc.*)

EDI Eclipse

C'est un outil Open Source et gratuit.

Il est développé en Java, ce qui le rend portable sur les plateformes disposant de l'environnement d'exécution Java (*la JVM, Java Virtual Machine*).

2- Téléchargement et installation – version actuelle 4.4 (sept 2014)

Site officiel : <u>https://www.eclipse.org</u>

Le lien de téléchargement est disponible sur le site : <u>https://www.eclipse.org/downloads/</u>



Choisir le site de téléchargement : **Download** eclipse-standard-luna-R-win32-x86_64.zip from:



Enregistrer le fichier ZIP (archive compressée) : taille. env. 200 Mo. puis extraire les fichiers de l'archive

3- Lancement et création d'un premièr projet

| Démarrage | readme |
|--|---|
| Dans le répertoire extrait, double-clic sur Eclipse : | artifacts.xml |
| Le chargement est relativement long : beaucoup de modules sont chargés au cepart afin d'améliorer l'utilisation du produit en développement. | eclipse.ini eclipsec.exe epl-v10.html |
| Au premier lancement, une fenêtre propose de définir le répertoire de travail d'Eclipse. Une case à cocher permet de ne plus afficher cette fenêtre par la suite. | notice.html |
| Select a workspace Eclipse stores your projects in a folder called a workspace. Choose a workspace folder to use for this session. | |
| Workspace: C:\Users\moi\workspace 		 Browse | |
| Use this as the default and do not ask again OK Cancel | |

La fenêtre d'accueil propose divers choix, parmi lesquels l'espace de travail (workbench) :



La fenêtre de l'application peut comporter des différences dans la présentation :

| Java - Eclipse File Edit Source Ref. | actor Navigate Searc | h Project Run Window Help | State of Lot of | |
|---|------------------------------------|---------------------------|---|------------------------------|
| | • ○ • ♀ • ⊕ <i>⊙</i> | ★ (2) ★ 2 ★ 3 ★ 5 ← ★ → ★ | Qui | ick Access |
| 📕 Package Explorer 🛛 | | | - 8 | 🗄 Outline 🖾 👘 |
| | | | | An outline is not available. |

Par défaut, la fenêtre est prête pour les développements Java : le terme « perspective » définit un ensemble d'outils (*ensemble de « vues », fenêtres présentant des informations particulières en fonction du contexte*):

- perspective « Java » pour saisir ou modifier du code
- perspective « Debug » pour aider à la découverte des anomalies de fonctionnement d'un programme

| Créer un nouveau | projet | | | | | | | |
|-------------------|----------------|---------------------|----------|--|--------|--------------|-------|------|
| >> menu File > Ne | w > Java Pı | <mark>roject</mark> | | | | | | |
| • | Java - Eclipse | | | | | | | |
| Fi | e Edit Source | Refactor | Navigate | Sea | Projec | t Run Wi | indow | Help |
| | New | | Alt+ | +Shit | | Java Project | t | |
| | Open File | | | and the second s | | Project | | |
| | Close | | | Ctrl+W | 1 1 | Package | | |
| | Close All | | Ctel | Ch:++ . \\ | 6 | Class | | |

Une nouvelle fenêtre permet de définir

1. le nom du projet

- 2. le répertoire de stockage des fichiers du projet (codes sources Java, entre autres)
- *3. la version de Java utilisée (Eclipse a dû le découvrir par défaut)*
- 4. le fait de créer un répertoire différent pour les codes sources et les codes compilés

Cliquer sur « Next » pour continuer

| Project name: MonProjet1 | | |
|--|--------------------|-----------------|
| Project name. MonProjett | | 1 |
| Use <u>default</u> location | 10.000 | |
| Location: C:\Users\moi\workspace\MonPr | rojet1 2 | Browse |
| JRE | | |
| Our Sean execution environment JRE: | JavaSE-1.7 | |
| lise a project specific IPE: | idk1 7.0.11 | |
| | [JUKI./.0_II | |
| Use detault JKE (currently Jak1.7.0_11 |) | Contigure JKE |
| Project layout | | |
| O Use project folder as root for sources | and class files | |
| <u>•</u> <u>Create separate folders for sources an</u> | id class files 🧹 🗸 | Configure defau |
| | | |
| | | |

La possibilité est offerte d'ajuster les paramètres du projet : (laisser par défaut) : Finish

| O New Java Project | | | X |
|--|-------------------|--------------|--------|
| Java Settings Define the Java build settings. | | V | |
| Image: Source Image: Source< | 20 | BR (7 | |
| ▼ Details | | | 5 HI 3 |
| Create new source folder: use this if you want to add a new source folder to project. Link additional source: use this if you have a folder in the file system that source as additional source folder. Add project 'MonProjet1' to build path: Add the project to the build path | o your hould b | pe | |
| ? < Back Einish | C | ancel | |

Le projet est créé avec un répertoire « src » pour le stockage des fichiers sources Java (.java) :



Ajouter une classe au projet

Un clic-droit sur « src » permet l'ajout d'une nouvelle classe :

| File Edit Sou | rce Refactor N | avigate Search | Project Run | Wind | dow Help |
|--|----------------|----------------|----------------|------|--------------|
| 🖻 🔻 🔪 🖄 | 10 4 + 0 - | 9 IP @ - | - 1 🦛 🛷 🗸 1 🖉 | - | |
| | New | | • | 凶 | Java Project |
| H Pa | Open in New W | indow | | | Project |
| ~ | Open Type Hier | archy | ۲4 | | Package |
| MonP | Show In | | Alt+ | G | Class |
| src 🖉 | | | | 0 | Interface |
| The second secon | - | | Ale a constant | | |

Une nouvelle énêtre permet de définir

- 1. le nom de la classe
- 2. le nom du package (un message « l'utilisation du package par défaut est à éviter » : c'est pourtant ce que nous ferons dans un premier temps...
- 3. des modificateurs de classe
- 4. *des squelettes de méthodes ou autre pour préparer la classe*
- 5. la génération des commentaires de documentation

| New Java Class | | | |
|--|---|-----------|-----------------|
| Java Class | efault package is discouraged. | | C |
| Source fol <u>d</u> er: | MonProjet1/src | | Br <u>o</u> wse |
| Pac <u>k</u> age: | 2 | (default) | Bro <u>w</u> se |
| Enclosing type: | | | Bro <u>w</u> se |
| Na <u>m</u> e: 1 Modifie Superclass: | Point | ected 3 | Brows <u>e</u> |
| Which method stub | os would you like to create? public static void main(String[] args) Constructors from superclass Inherited abstract methods d comments? (Configure temr's and default val | ue here) | <u>A</u> dd |
| ? | Generate comments 5 | nish | Cancel |

Le code source de la classe est prêt à saisir :



Modifier le code source - Complétion de code et anticipation des erreurs

Au fur et à mesure de la saisie, l'éditeur propose des options contextuelles valides pour simplifier la saisie : c'est ce qu'on appelle la « complétion de code » : ici, après « this. », on peut saisir un de ses membres (*ou un des membres hérités de la classe Object*).

La combinaison de touches CTRL+espace permet à tout moment d'afficher le liste des propositions)



En cas d'erreur supposée de syntaxe, un indicateur d'erreur dans la marge permet d'aider au diagnostic et à la correction (ici, on est passé à la ligne suivante sans saisie de « ; » ni d'une éventuelle affectation) :



A tout moment, des aides contextuelles permettent également la génération de portions de code :

Génération de commentaires Javadoc :

par exemple ici, un clic-droit dans le bloc du constructeur donne accès à la génération des commentaires documentaires de cet élément :



PROG_EDI_eclipse.doc

15/09/2014 06:57:00

ce qui génère les commentaires suivants, qu'il est possible d'enrichir afin d'obtenir une documentation claire (*utilisation de Javadoc, intégrée à Eclipse*) :

| 🔵 Java - MonProjet1/src/Point.java - | Eclipse | | | | | |
|--|---|--|---------------|------------|---|---|
| <u>File Edit Source Refactor Navig</u> | ate Se <u>a</u> rch <u>P</u> roject | <u>R</u> un <u>W</u> indow | <u>H</u> elp | | | |
| i → (a) ≥ (×, l ↔ + O + 9 | • # G • 9 % | • 🖗 🗾 🔳 🔳 | § • § • ⊕ ⊖ · | • 👳 • | Quick | Access 🖹 🖻 🐉 Java |
| Package Explorer 23 MonProjet1 Src (default package) (Point.java Point JRE System Library [JavaSE- | Point.java ⋈ 1 2 public class 3 private 4 private 5⊕ /** 6 * @pain 7 * @pain 8 */ 9⊕ Point of 10 th: 11 th: 12 } 13 | <pre>ss Point { e int x; e int y; ram x ram y (int x, int y) is.x = x; is.y = y;</pre> | | | | E Outline S □ □ ↓ ⁴ z × × ⁵ • × ⁴ ~ ■ Point d ^a x : int d ^a y : int ▲ ° Point(int, int) |
| | Problems ⊠ @. | Javadoc 📵 Decla | ation | | * | V |
| | 0 errors, 2 warnings, | 0 others | | The second | in a second s | |
| | Description | - | | Resource | Path | Location |
| • | 4 141 | ····· | Ш | | 1 | • |
| | Writable | Smart Insert | 9:27 | | | |

Génération de code Java

Pour les membres attributs, il est ainsi possible de générer automatiquement les accesseurs à partir d'un clicdroit dans le code source :

| P 🌮 Point 🛛 💆 | Show In | Alt+Shift+W ► | t = |
|---------------------------------------|----------------------------|---------------|------------------------------|
| JRE System Library [JavaSE- 8 | Cut | Ctrl+X | Toggle Comment |
| 90 | Сору | Ctri+C | Remove Block Comment |
| 10 | Copy Qualified N | ame | Generate Element Comment |
| 12 | Paste | Ctrl+V | Correct Indentation |
| • | Quick Fix | Ctrl+1 | Format |
| Summer and A | Source | Alt+Shift+S ► | Format Element |
| Prot | Refactor Local History | Alt+Shift+T > | Add Import |
| 0 errors Descri | References Declarations |).). | Sort Members Clean Up |
| · · · · · · · · · · · · · · · · · · · | Debug As | | Override/Implement Methods |
| Writa | k Run As | | Generate Getters and Setters |

Il est possible de préciser :

- 1. les accesseurs à insérer
- 2. la position d'insertion dans le code et l'ordre d'apparition des accesseurs
- 3. le modificateur d'accès des accesseurs
- 4. l'ajout des commentaires associés

| Generate Getters and Setters | |
|--|---|
| Select getters and setters to create: | |
| 1 | Select <u>All</u> <u>D</u> eselect All Select <u>G</u> etters Select Setters |
| Allow setters for final fields (remove 'final' modifier from fields if ne Insertion point: After 'Point(int, int)' Sort by: | ecessary) |
| Fields in getter/setter pairs | • |
| Access modifier public protected package private final synchronized | |
| Generate method <u>comments</u> The format of the getters/setters may be configured on the <u>Code Temp</u> i 4 of 4 selected. | <u>plates</u> preference page. |
| ? ОК | Cancel |

et le résultat !



Compléter le code avec les méthodes « métier » (spécifiques)

La classe sera complétée



Une représentation hiérarchique donne des indications sur le type de chaque membre (static, constructeur, attribut ou méthode) et donne un accès immédiat (double-clic) au code source de chacun d'eux :



Compiler la classe – construire le projet

L'option de construction, « Build », (compilation) est activée par défaut :

| Point.java - Eclipse | | |
|---|---|--------------|
| tor Navigate Search | Project Run Window Help | |
| • 0 • • • • • | Open Project Close Project | ⇔⇔⊸⇒⊸ |
| ige) | Build All Ctrl+B Build Project Build Working Set Clean | I main(Strin |
| and the second se | Build Automatically | ion |

Lancer l'exécution

L'exécution peut être lancée : >> menu Run > Run (ou CTRL-F11)



Le résultat de l'exécution est affiché dans la console intégrée à l'éditeur, au bas de l'écran :



Fermer le projet

Clic-droit sur le projet > Close project.

Le projet est fermé, on peut alors passer à un autre projet !



4 - Débogage d'applications

Le débogage est l'activité consistant à découvrir les anomalies d'exécution d'un programme qui ne donne pas le résultat attendu lors des tests avec un jeu d'essais.

Les EDI intègrent généralement des outils facilitant ces opérations :

- points d'arrêt : le point d'arrêt permet d'interrompre l'exécution d'un programme, de vérifier certaines valeurs, puis de relancer l'exécution
- consultation des valeurs intermédiaires des variables :
- etc.

Poser des points d'arrêt

Un clic-droit sur un numéro de ligne permet le positionnement d'un point d'arrêt de l'exécution (ici ligne 42) :



L'exécution en mode « Debug »

| ose | * | | 100 |
|-------------------------------|---------------------|----------|-------|
| Search Project Run | Resume | | |
| | Suspend | | mm |
| er and and the start of a 🖉 📕 | Terminate | | ar ve |
| 🕒 😫 🗸 🗖 🕻 🕅 | Disconnect | | |
| 23. | Step Into | | |
| | Step Into Selection | | |
| 5 | Step Over | | y=" - |
| _R | Step Return | | |
| -5] | Run to Line | | |
| 30 | Use Step Filters | Shift+F5 |] arg |
| 9 | Run | Ctrl+F11 | |
| J. | Debug | F11 | |

Avertissement de passage dans la perspective « Debug » :

| ? | This kind of launch is configured to open the Debug perspective when it suspends. |
|-------------|---|
| | This Debug perspective is designed to support application debugging. It incorporates views for displaying the debug stack, variables and breakpoint management. |
| | Do you want to open this perspective now? |
| <u>R</u> en | nember my decision |

La prespective « Debug » est activée, l'exécution a lieu jusqu'au prochain point d'arrêt. On peut alors observer le contenu des variables à ce stade de l'exécution :



Après arrêt de l'exécution, on peut revenir à l'édition du code Java afin d'effectuer des modifications puis de recompiler (ici automatique).

Pour revenir à la perspective « Java »: