Java programming with Eclipse

D. Conan and S. Leriche

Internship course

February 2008

Outline

1 References

- Documentation
  - Web site: `http://www.eclipse.org`
  - A short guide [Burnette, 2005]
  - The welcome screen proposes to explore introductory materials, including examples and tutorials
    - After the first execution, available in Help/Welcome
- Installation: `debian package` or package from the website
  - Different flavors:
    - Java is the standard one
    - J2EE adds tools and wizards for enterprise development
    - RCP/Plugin allows to build your own Eclipse plugins
    - Classic contains developpment tools and source code for customizing Eclipse
- Plugins: `http://www.eclipseplugincentral.com`
- Tutorials: `http://eclipse-tutorial.dev.java.net`
- Eclipse help: `http://help.eclipse.org`

2 Motivations and principles

- Widely used IDE for Java (and made in Java), but not only !
- Eclipse needs a JRE (1.4+)
  - A specific Java compiler is embedded, no need for a JDK
  - Suggestion: Use latest Sun's JRE (some distrib use gcj, not recommended)
- Everything is a plugin
  - Very extensible (UML, Test, GUI...) and customizable
  - A plugin can add new wizards/functionalities/...
  - If you have boring/dirty work to do, search for a plugin
  - Plugins can be installed by a simple 'unzip' in the `Plugins` sub-directory

3 Terminology

- Workspace = location where settings will be stored
  - Suggestion: Do not manually install your source code in your workspace
- Workbench = main window, composed of views and editors
- Perspective = a set of views, editors and toolbars = a configuration
  - Resource: Arrange your files and projects
  - Java: Develop programs in the Java language
  - Debug: Diagnose and debug at runtime
  - Java type hierarchy: Explore your code based on class relationships
- Main views
  - Package explorer: All projects, Java packages and files
  - Hierarchy: Class and interface relationships for the selected object
  - Outline: Structure of the currently open file
  - Problems: Compiler errors and warnings
  - Console: Output of program executions

4 Workbench

- Open a view: Window/Show View
- Switch to a perspective: Window/Open Perspective or icon to the right of the main toolbar
  - You can save your favourite window arrangements as named perspectives
- Configure the Java perspective
  - Editor (emacs):
    Windows/Preferences/General/Editors/Keys/Modify/Scheme
  - Text encoding (ISO-8859-1):
    Windows/Preferences/Workspace/Text file encoding
  - Show line numbers and print margin (80 columns)
    Windows/Preferences/General/Editors/Text editors
  - Code folding/unfolding (unfolding)
    Windows/Preferences/Java/Editor/Folding
  - Code formating (eclipse 2.1 built-in)
    Windows/Preferences/Java/CodeStyle/Formatter

- Java JRE (do not choose `gcj` for Java 1.5)
  Windows/Preferences/Java/Installed JREs... Add... Browse...
- XML plug-in (for Fractal ADL files)
  - XML Buddy: `http://www.xmlbuddy.com`
  - `unzip` in the directory eclipse/plugins
  - Close and open eclipse, check for the new plug-in in Help/About Eclipse SDK... Plug-in Details
  - Parse `*.fractal` files as XML files
    Windows/Preferences/General/Editors/File association... Add *.fractal and associate XMLBuddy
  - Disable XML validation by XMLBuddy since the Fractal DTD is not a "true" DTD

5 Managing a project

- A project = a directory structure
  - Hidden config files (`.project` and `.classpath`)
- Create a project, a package, a class
- Configure a project
  - Compiler options
    Project/Properties/Java Compiler...
  - Code templates (Java files header)
    Project/Properties/Java Code Style/Code Templates... Enable project specific settings... Comments... Files
  - Indicate where is the source code
    Project/Properties... Java build path... Source
  - Default output folder (`eclipse-build` instead of `build`)
    Project/Properties... Java build path... Source
  - `jar` dependencies
    Project/Properties... Java build path... Libraries

6 Writing Java code

- Automatic insertion of "imports"
- "Content assist" for a list of suggested completions for partially entered strings
  - Ctrl+Space or invoke Edit/Content Assist
- "Java assist" for automatic completion and documentation
  - *E.g.*, `sysout` → `System.out.println();`
  - *E.g.*, `for` → proposes four forms
- Quick fix of errors and warnings
  - Right click in the Problems view on the warning/error and chose Quick fix
  - Click on the bubble of the warning/error, and select the correction

7 Tips and Tricks

- Ctrl+Shift+L: List of current key definitions for shortcuts
  - Window/Preferences/General/Keys
- Ctrl+Shift+F: Reformat (beautify) your code
- Shift+Ctrl+O: Automatically add "imports"
- Ctrl+click on class name: Go to that source code
- Shift+Alt+/: Call to Java assist
- Ctrl+Space: Call to Content assist
- Shift+Alt+J: Add Javadoc comments
- Shift+Ctrl+/: Comment/Uncomment a line or a region

8 More on Eclipse *

- Source control management with Subversion
- Execute Ant tasks: WINDOWS/SHOW VIEW/ANT and drag&drop the `build.xml` file
- Debugging
- Unit testing with JUnit
- Installing the plug-in `subclipse` to use Subversion
  `http://subclipse.tigris.org/install.html`

References

[Burnette, 2005] Burnette, E. (2005). *Eclipse IDE: Pocket Guide*. O'Reilly.