Integrated Development Environments & Eclipse

IDE

- IDE stands for Integrated Development Environment
- Features at least:
  - Resource editors
  - Compilers
  - Debuggers
  - An integration with external Version Control Systems
- What else can it provide?

Popular Java-oriented IDEs (1/4)

- Eclipse
  - Open Source:
    - Eclipse Public License
  - http://www.eclipse.org
  - Supported by Actuate, CA, IBM, Nokia, Oracle, SAP, etc.

Popular Java-oriented IDEs (2/4)

- IntelliJ IDEA
  - Community Edition is Open Source
  - Ultimate Edition has far more features but is commercial
  - http://www.jetbrains.com/idea/
  - Developed by JetBrains

Popular Java-oriented IDEs (3/4)

- NetBeans
  - Open Source:
    - CDDL: Common Development and Distribution License
    - GPL (General Public License) v2 with Classpath Exception
  - http://www.netbeans.org
  - Backed by Sun Microsystems (currently being acquired by Oracle)

Popular Java-oriented IDEs (4/4)

- Oracle JDeveloper
  - Not Open Source...
  - … But based on many Open Source technologies
  - … And free to use
  - http://www.oracle.com/technology/jdev
  - Developed by Oracle

Eclipse – Short History

- Started as a replacement for IBM's VisualAge in the late 90s
- 2001:
  - Eclipse 1.0
  - Released as Open Source
  - Still under IBM's control
- 2004:
  - Eclipse 3.0
  - Creation Eclipse Foundation
  - Wider adoption

Eclipse – Main Concepts (1/2)

- Resource:
  - A folder, a Java file (~Java class), a Groovy script, an HTML file, etc.
- Project:
  - Contains a set of related resources
  - Is itself a resource
  - E.g.: A Java project contains generally a set of Java files

Eclipse – Main Concepts (2/2)

- Workspace:
  - Contains a set of related projects
  - Contains a set of preferences
- Preference:
  - Allows setting up a workspace
  - Used to enforce standards
  - Examples:
    - Java compiler compliance level (1.3, 1.4, 5.0, 6.0)
    - Web links validation (HTML, JSP, etc.)
    - Runtime environments definitions

Eclipse – UI Concepts (1/3)

- Perspective
  - Defines the visual representation of resources
  - Defines the set of actions availables for these resources
  - Is organized through the use of *views* and *editors*
  - Examples:
    - Java perspective
    - Plug-in Development perspective
    - Debug perspective

Eclipse – UI Concepts (2/3)

- View:
  - Contributed to perspectives
  - Displays resources based on a specific point of view
  - Examples:
    - Package Explorer (in the Java perspective)
    - Plug-ins (in the Plug-in Development perspective)
    - Breakpoints (in the Debug perspective)

Eclipse – UI Concepts (3/3)

- Editor:
  - Shared by perspectives
  - Allows editing resources
  - One editor is available for one resource type
  - Examples:
    - Java Editor
    - Plug-in Manifest Editor

Eclipse – Plugins (1/4)

- A plugin is an add-on which extends Eclipse mainly to add new functionalities:
  - Perspectives
  - Views
  - Editors
  - Actions
  - Etc.

Eclipse – Plugins (2/4)

- Technically speaking, a plugin is:
  - A JAR file...
  - … Which contains Java classes
  - … And some libraries (JAR files)
  - … And a plugin manifest (MANIFEST.MF + plugin.xml)
- It can have dependencies on other plugins
- It extends *extension points*
- It can create new extension points

Eclipse – Plugins (3/4)

- A plugin is bundled through the use of *features* and *update sites*
- Features:
  - Are a mean to package coherent sets of plugins (and features) by allowing:
    - To define branding information (company, license, etc.)
    - To specify requirements (operating system, architecture, etc.)
  - Are JAR files containing mainly a feature manifest (MANIFEST.MF + feature.xml)

Eclipse – Plugins (4/4)

- Update sites:
  - Are a mean to easily distribute features...
  - … By bundling as an HTTP Web site, a local site or an archive
  - Integrates with Eclipse's “Find and Install” feature to easily manage the installation, update and removal of plugins
  - Contain a site manifest (MANIFEST.MF + site.xml), features (JAR files) and plugins (JAR files)

What is the mechanism behind Eclipse plugins?