6 Building Application using IDE

Objectives

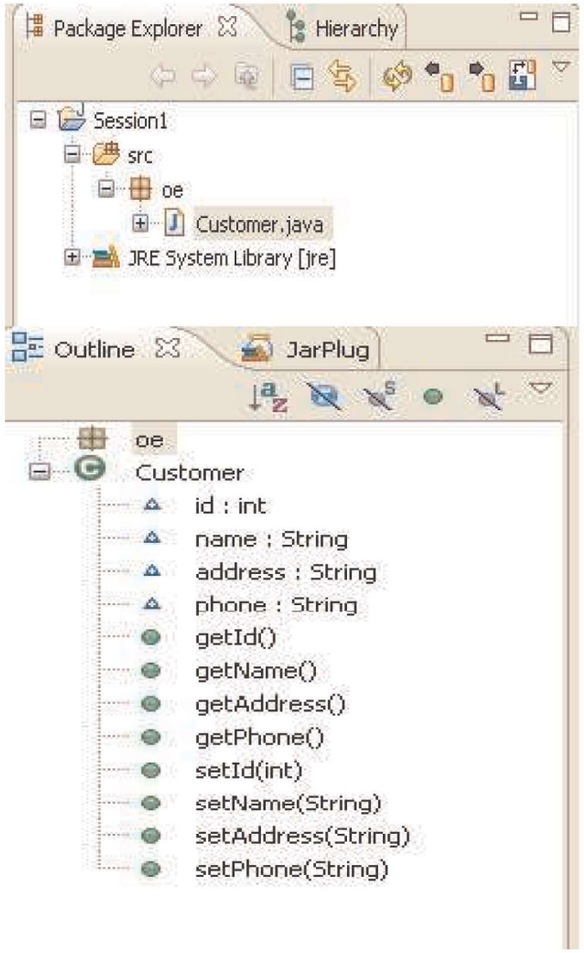
After completing this lesson, you should be able to do the following:

- Create applications and new projects
- Build Java applications in IBM Eclipse
- Enhance user interface frame design
- Debug applications with the Eclipse debugger
- Define classes with Eclipse
- Describe how Eclipse can be used to build enterprise applications

IBM Eclipse

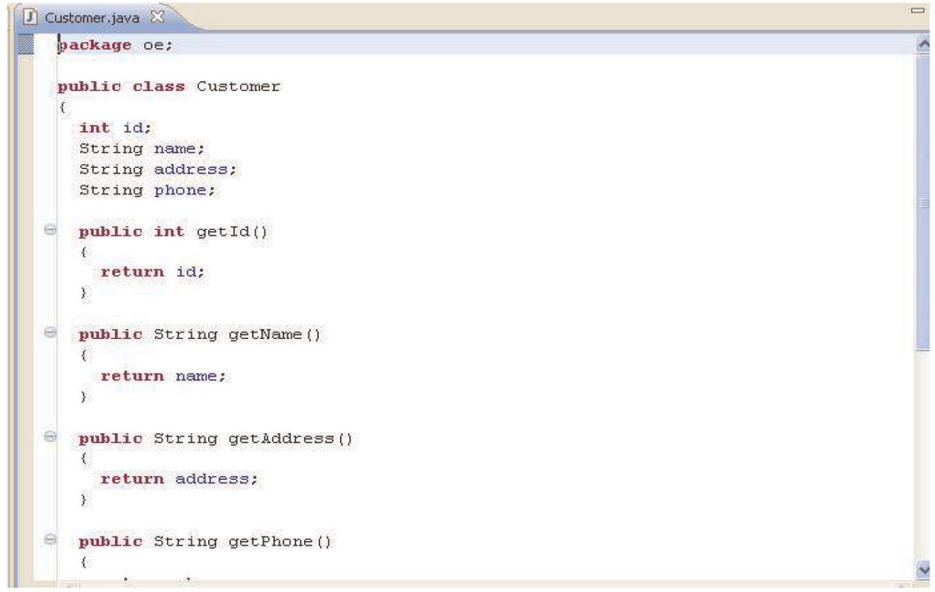
- IBM Eclipse 3.X provides an integrated development environment (IDE).
- It enables you to:
  - Build, compile, and run Java applications
  - Use wizards to help build source code
  - View objects from many perspectives: code, structure, layout, and so on

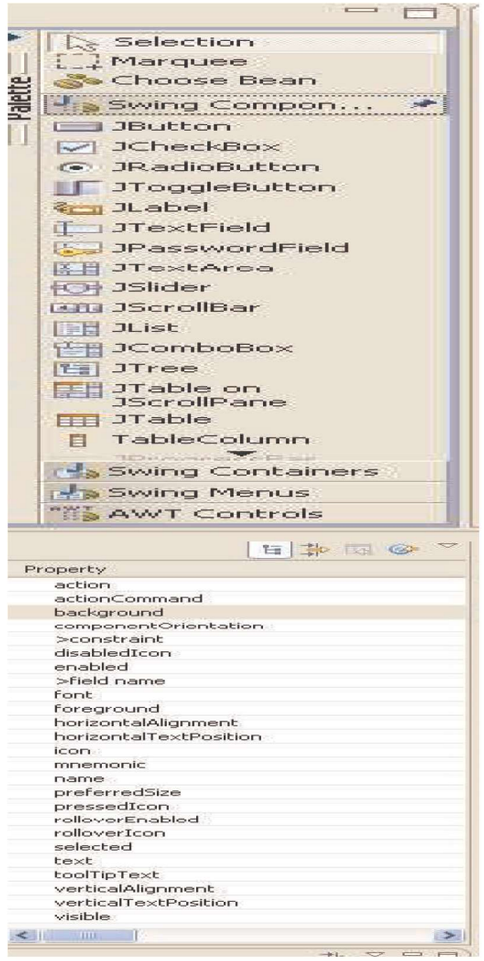
Eclipse Environment



Package Explorer

Component Palette





Outline View

Code Editor

Property Inspector

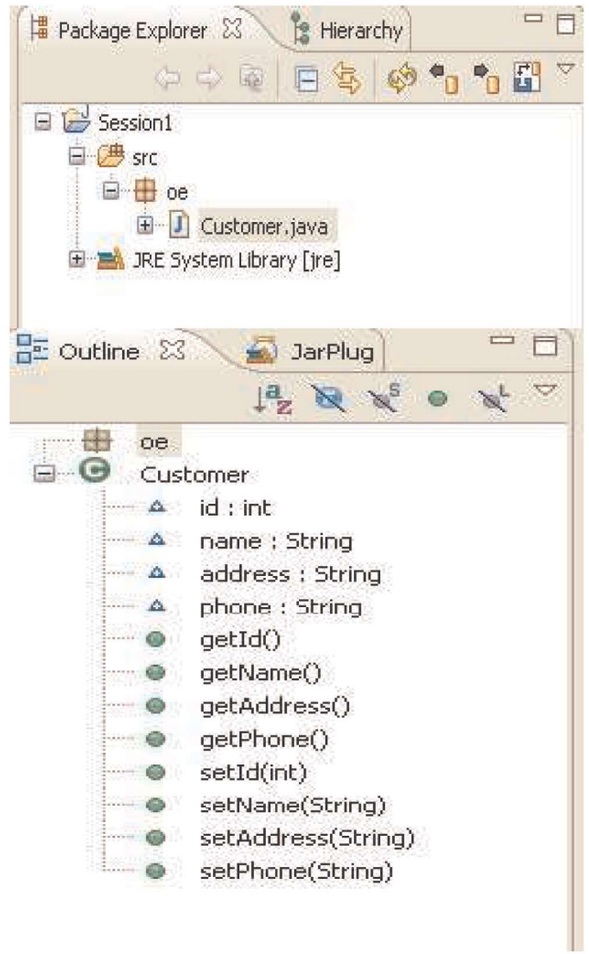
Projects

- May contain multiple projects
- Enable you to view currently used objects

Project node

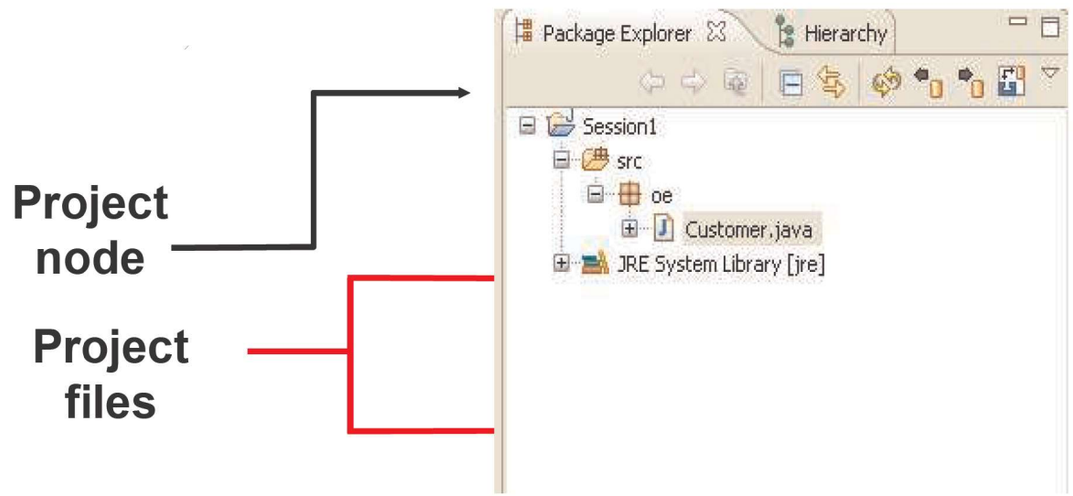
Projects Navigator pane

Outline pane

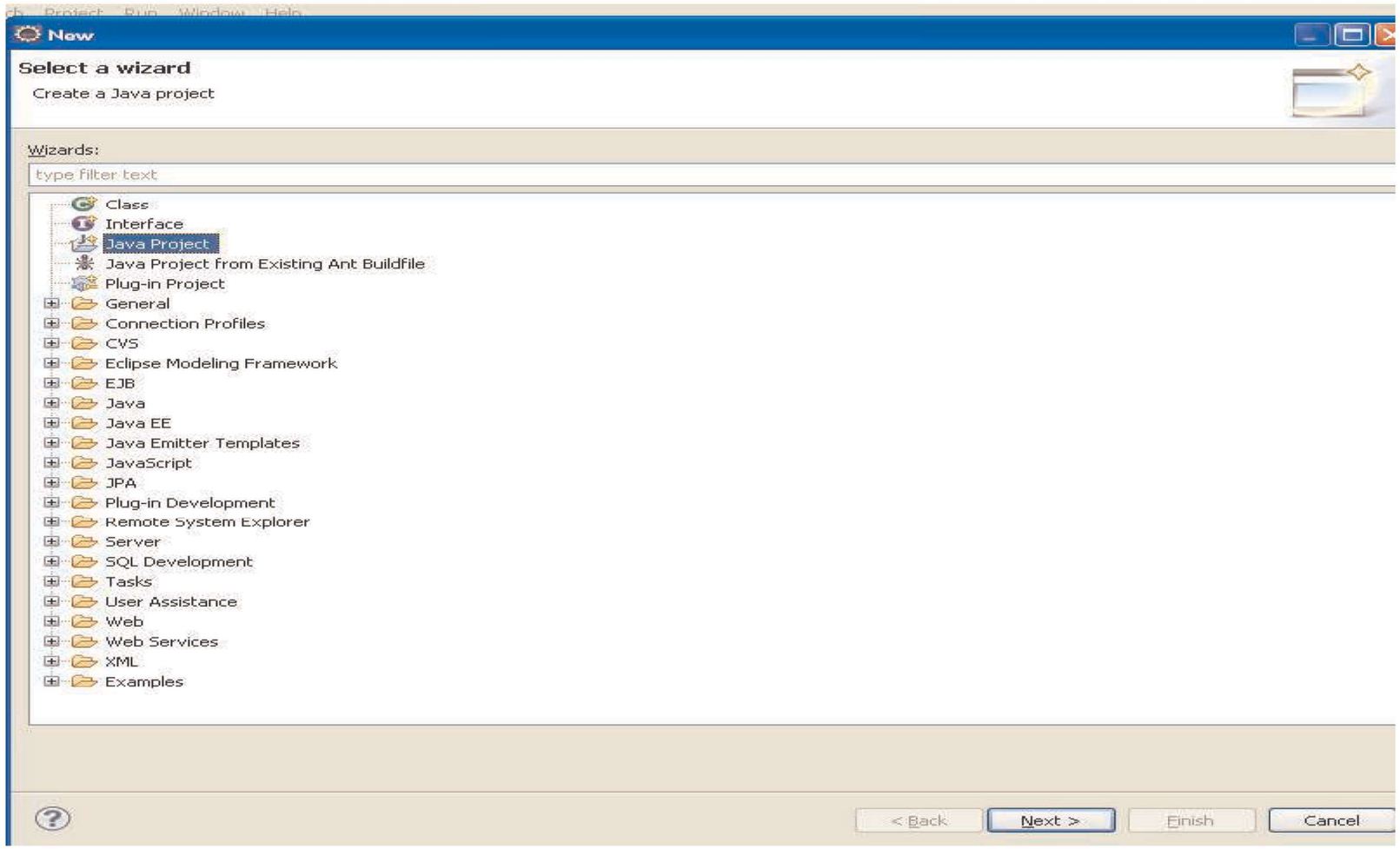


Projects

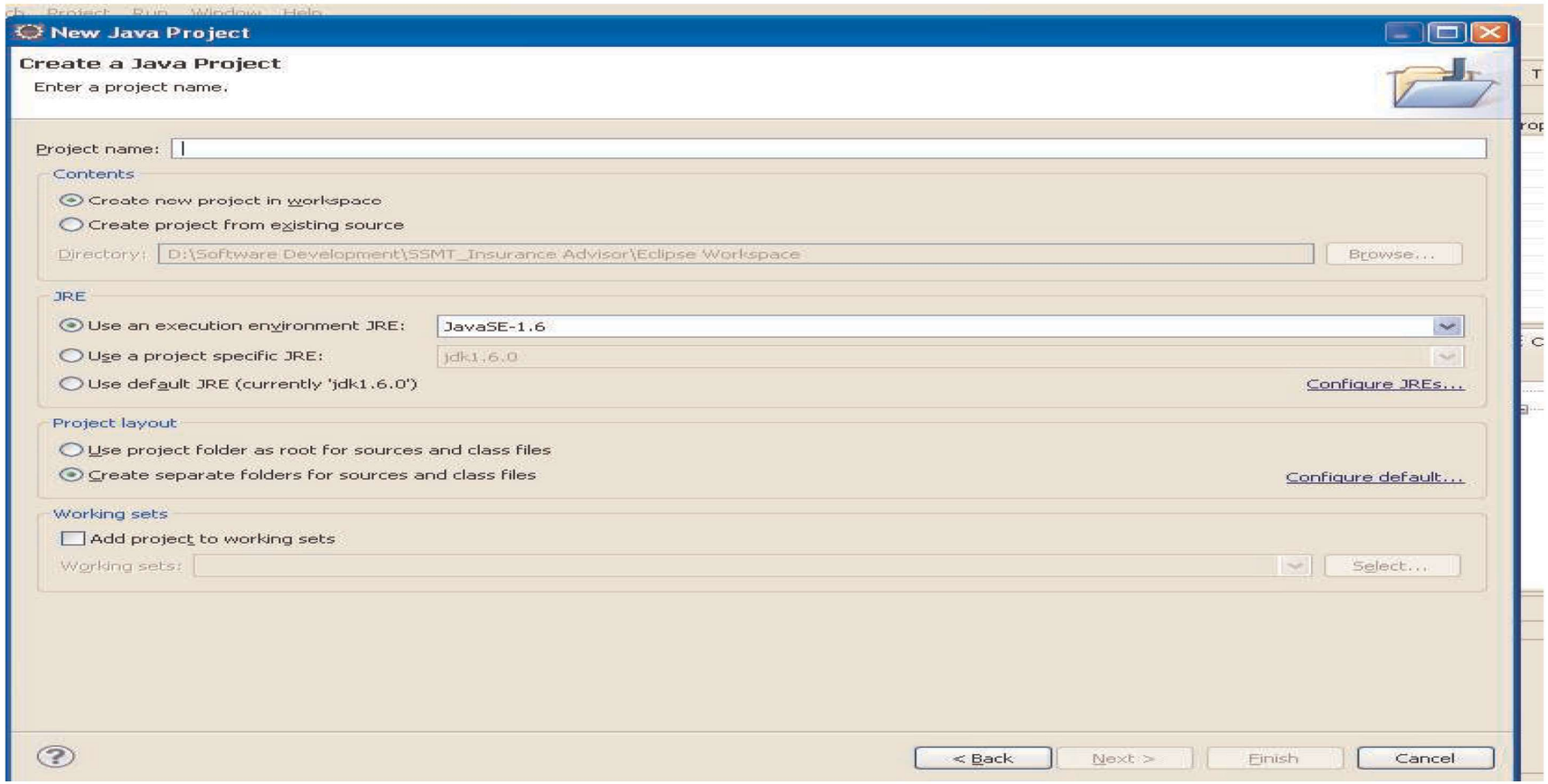
- Contain related files
- Manage project and environment settings
- Manage compiler and debug options



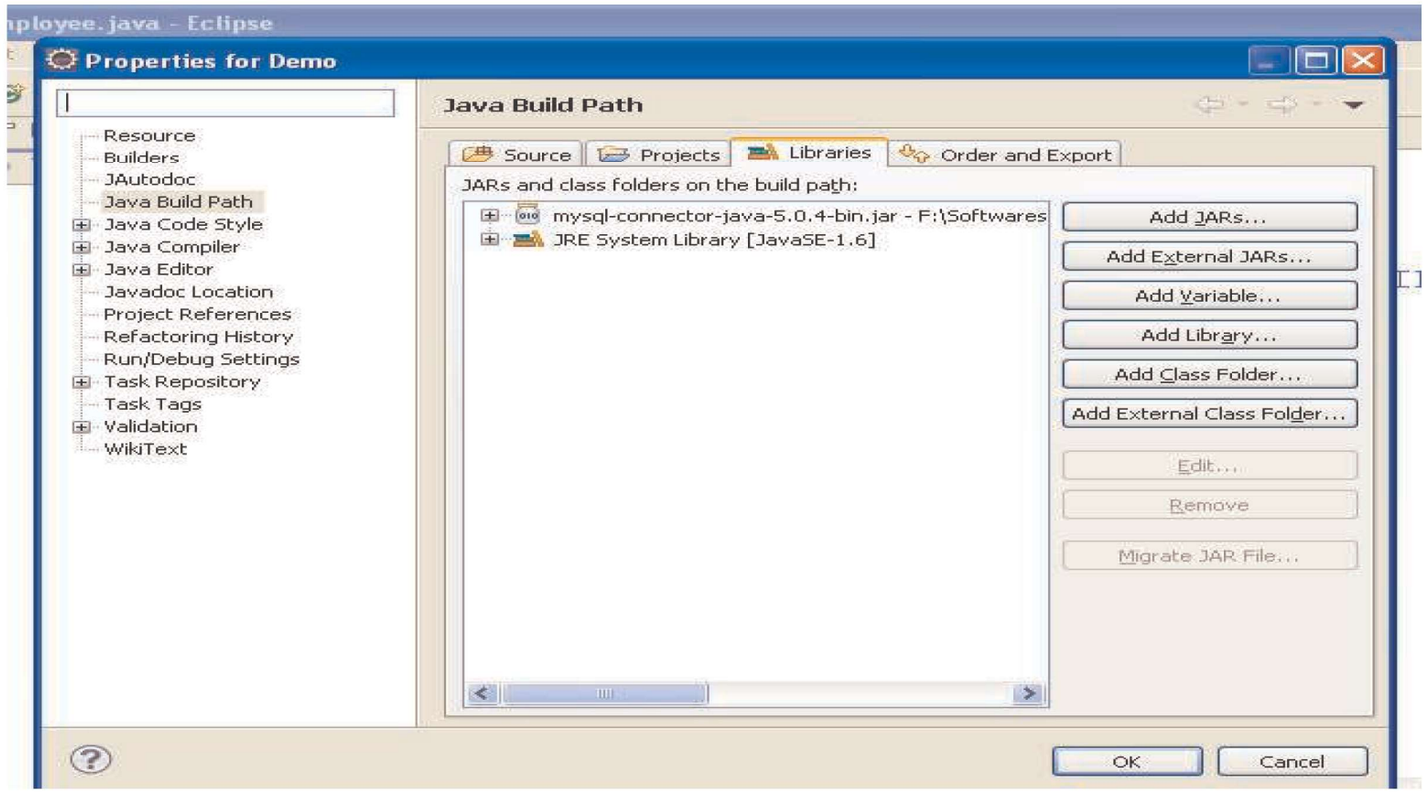
Creating Eclipse Items



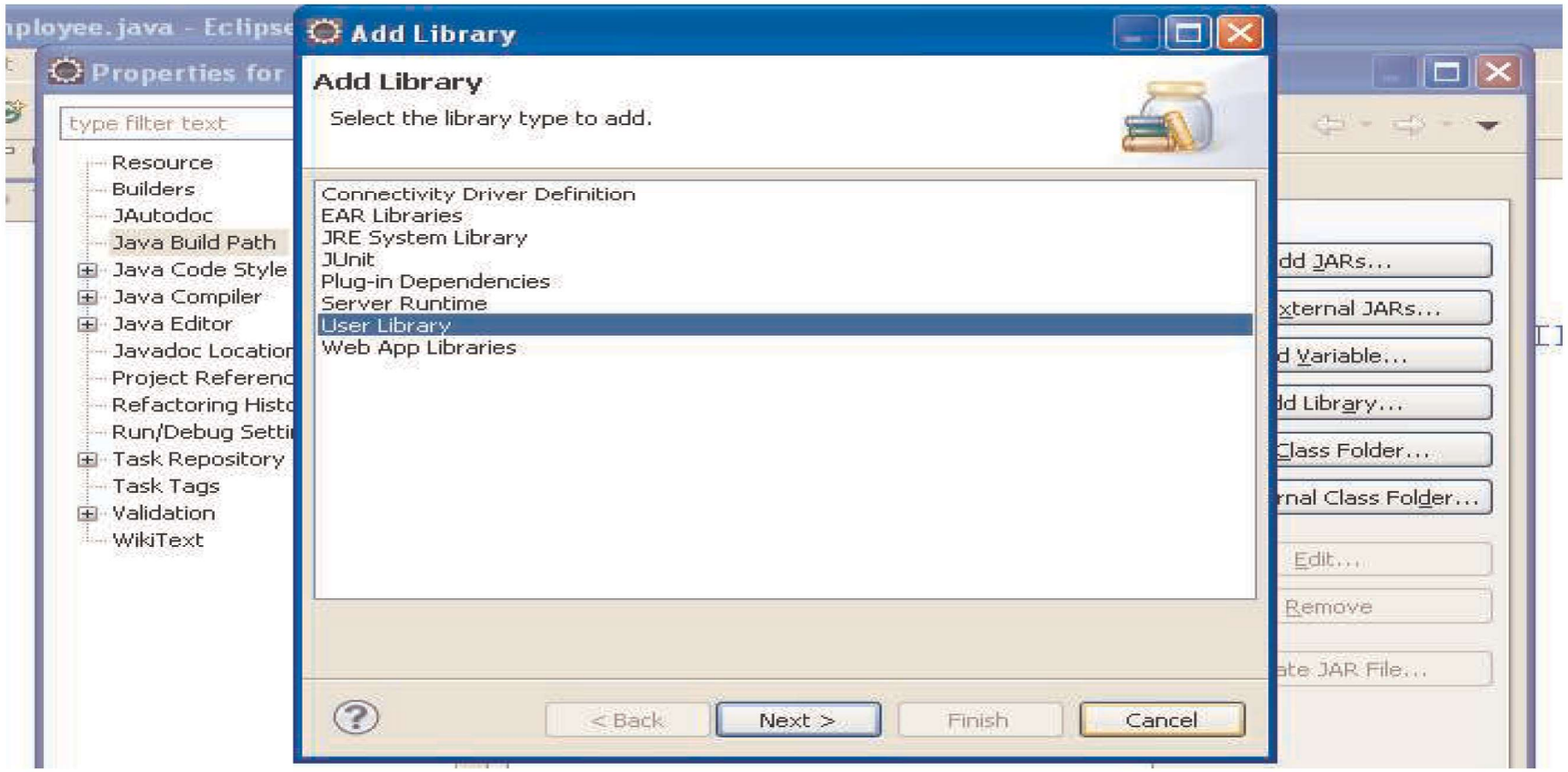
Creating an Project



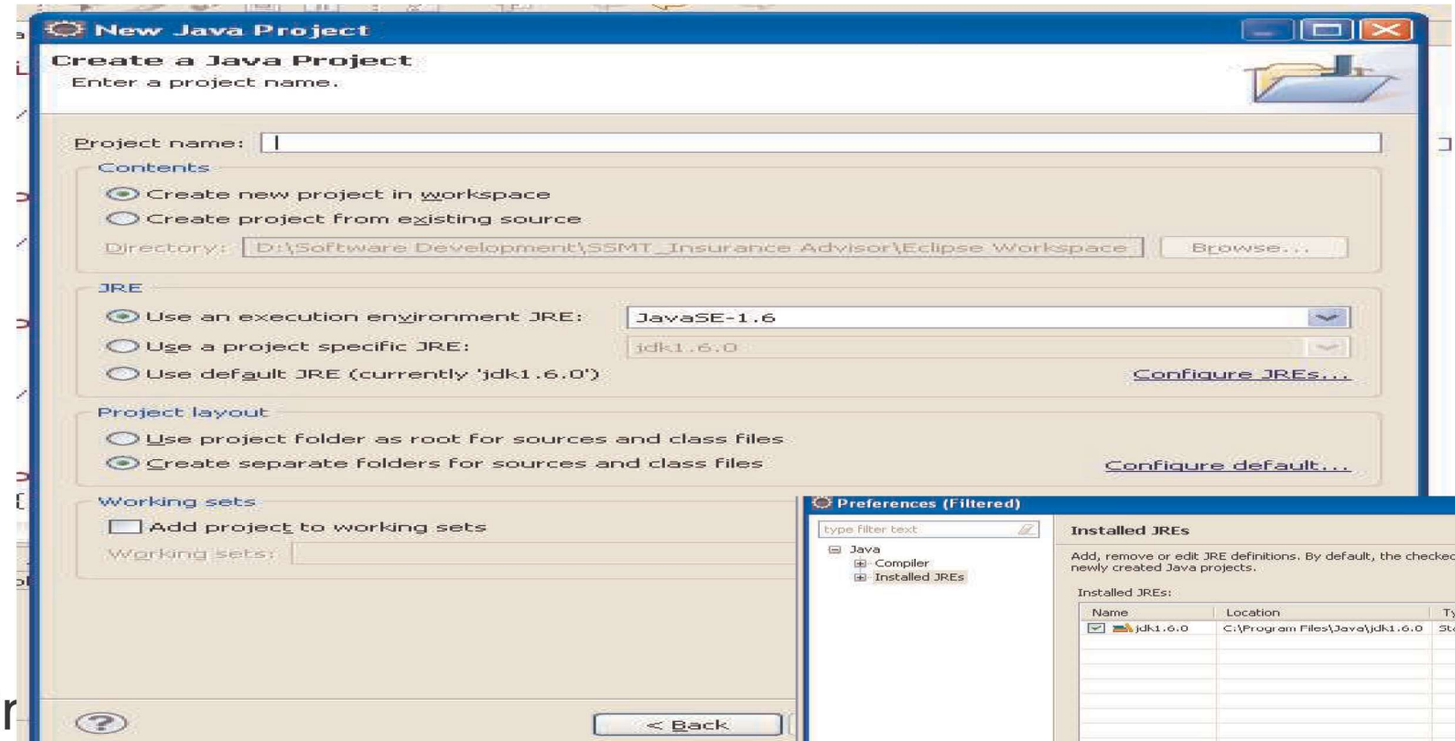
Project Properties:Specifying Project Details



Project Properties:Selecting Additional Libraries



Adding a New Java SE



- Java SE defir
  - Java executable
  - Classpath
  - Source path
  - Doc path

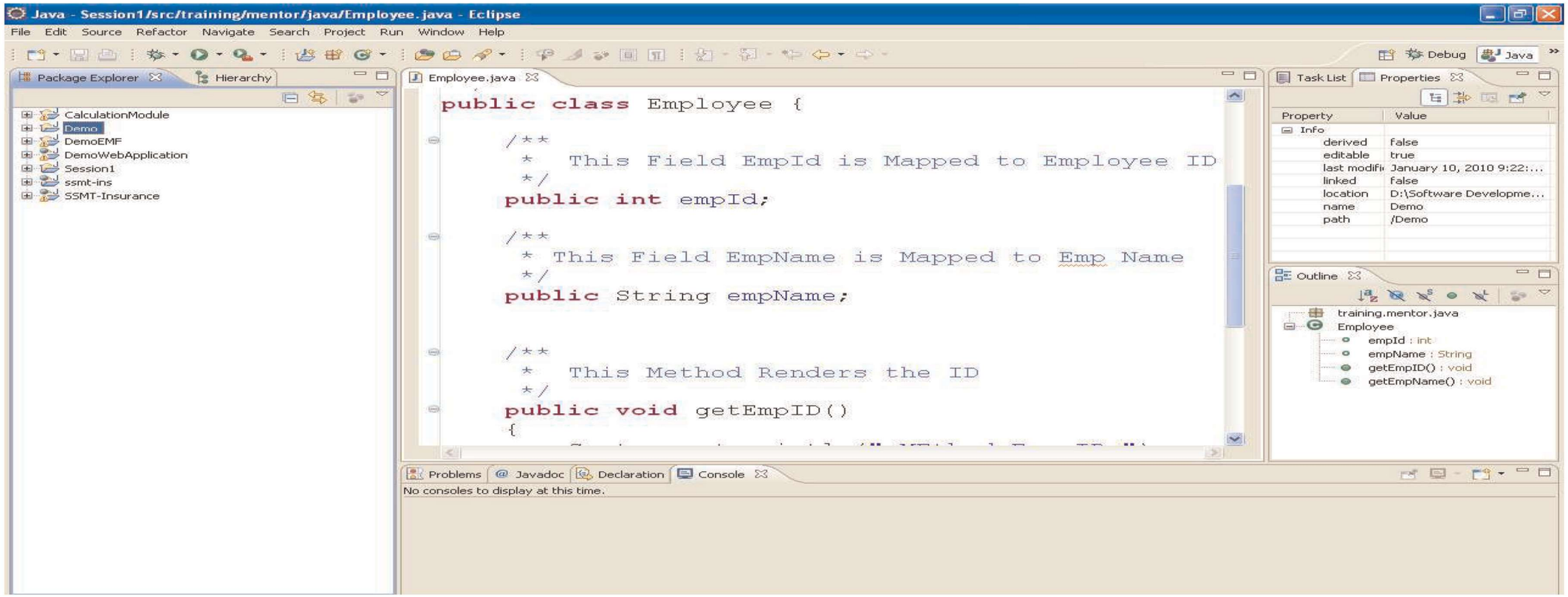
Directory Structure

Eclipse creates and stores `.java` and `.class` files by using the following conventions:

- `<Workspace>\<Name>`
- Followed by the application name
- Followed by the project name
  - `\bin\<package name>\`
  - `\src\<package_name>\`
- Followed by `bin` and `src` files

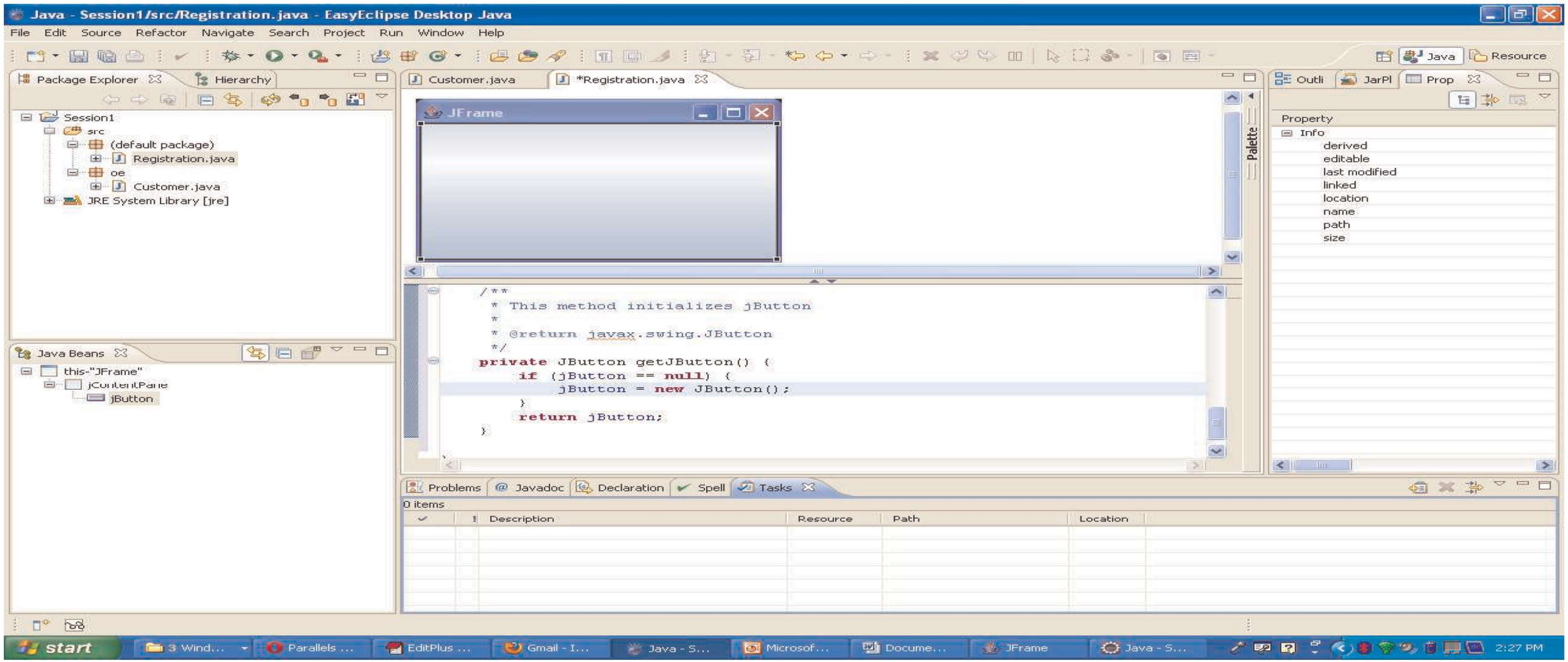
Exploring the Skeleton Java Application

Contains application classes:



Finding Methods and Fields

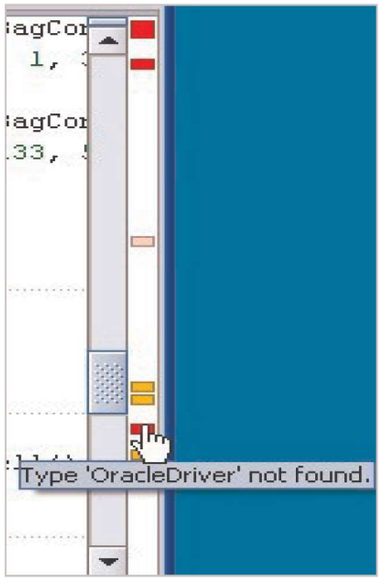
Find methods and fields using the Structure pane:



Supporting Code Development

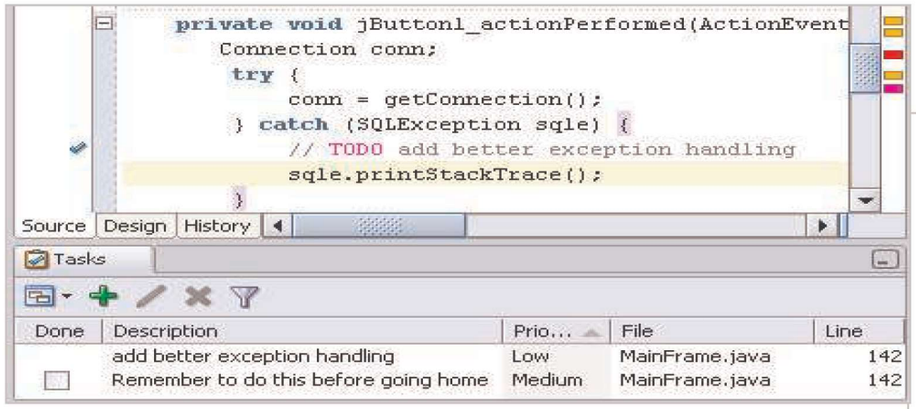
- Improve code quality with Code Coach.
- Evaluate execution stack with the Execution Sample profiler.
- Examine heap memory usage with the Memory profiler.
- Analyze event occurrence and duration with the Event profiler for:
  - JVM events
  - Business components for Java events
  - Custom events

New Code Editor Features

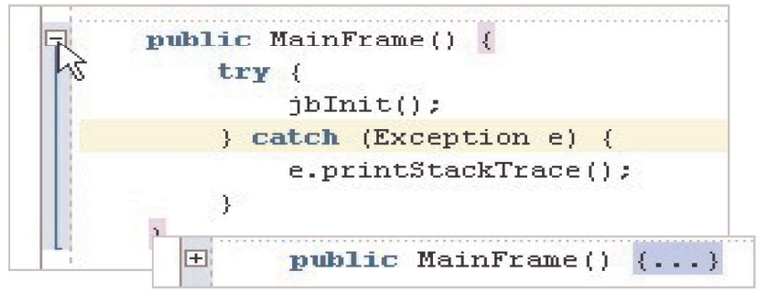


Overview margin

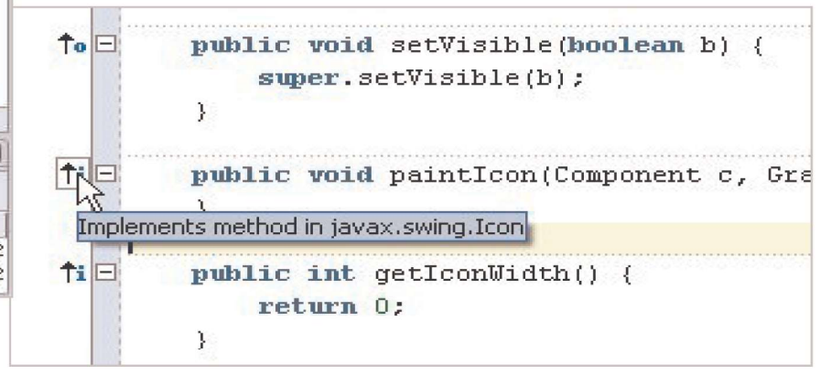
Code Assist



Tasks list



Scope and code folding



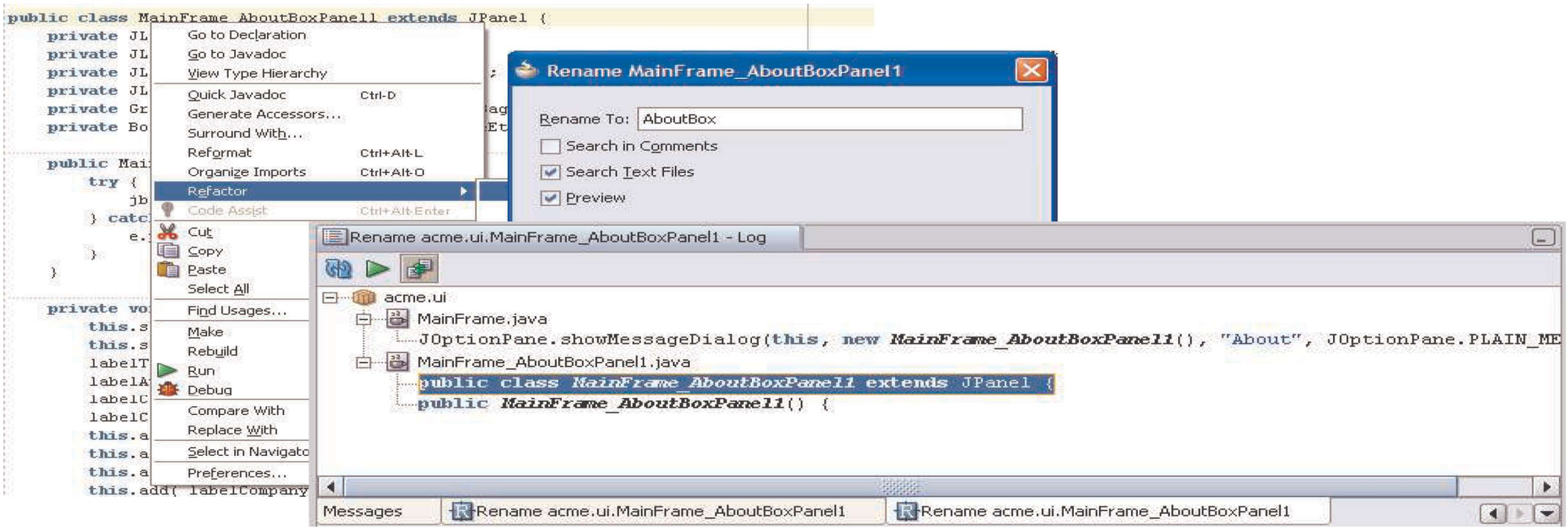
Implements and overrides navigation

Customizing Eclipse

Customize the IDE:

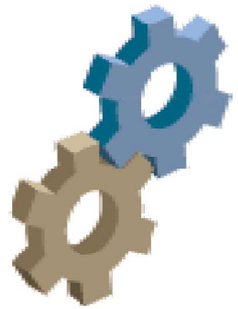
- Look and feel
- General environment
- Dockable windows
- Component Palette
- Preset keymaps

Modify the structure of code without changing its behavior (or breaking it).



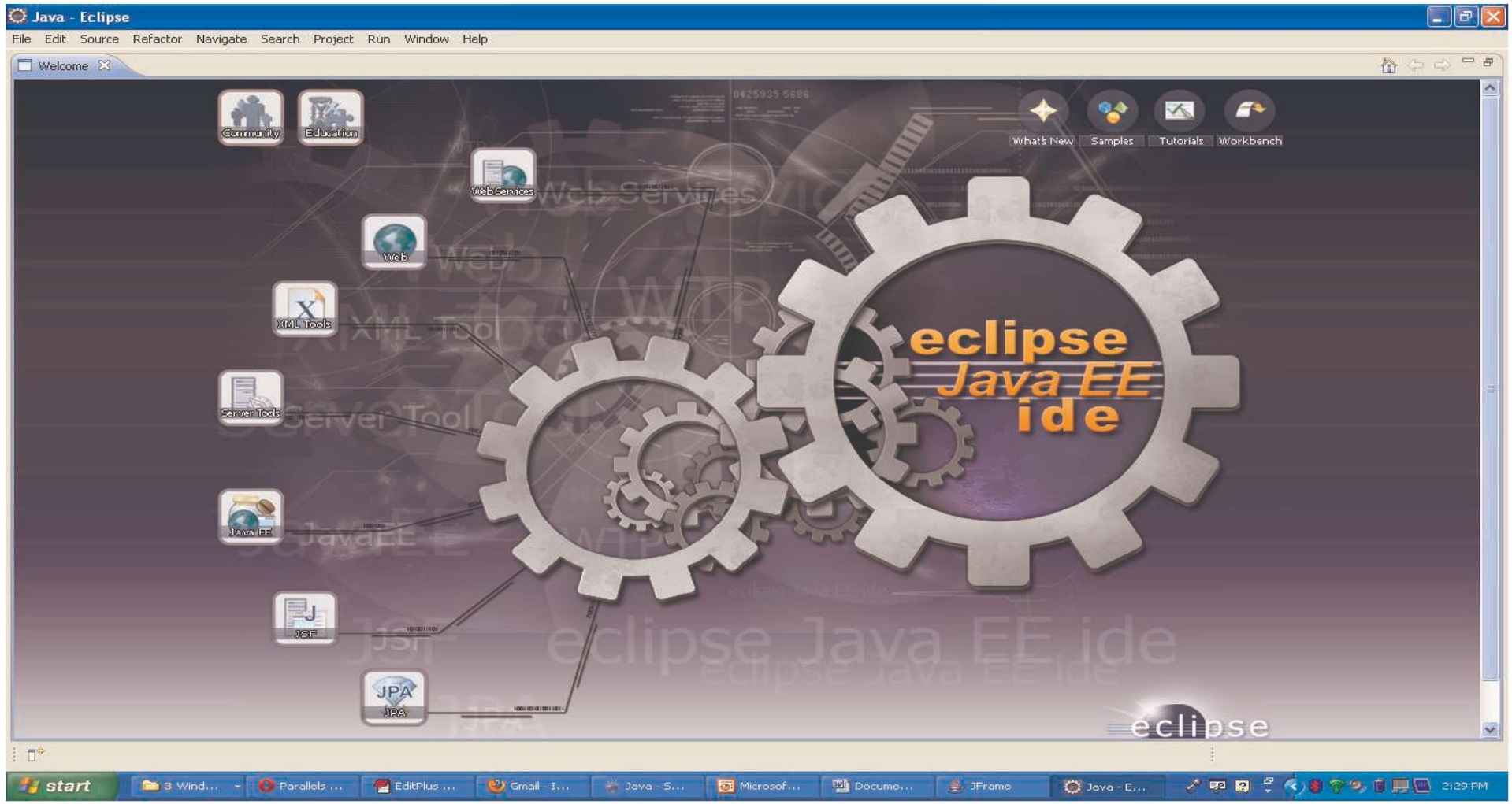
Refactoring

- Drag-and-drop refactoring
- Refactor across entire application
- Refactor across source control
- More than 35 new refactoring operations, including:



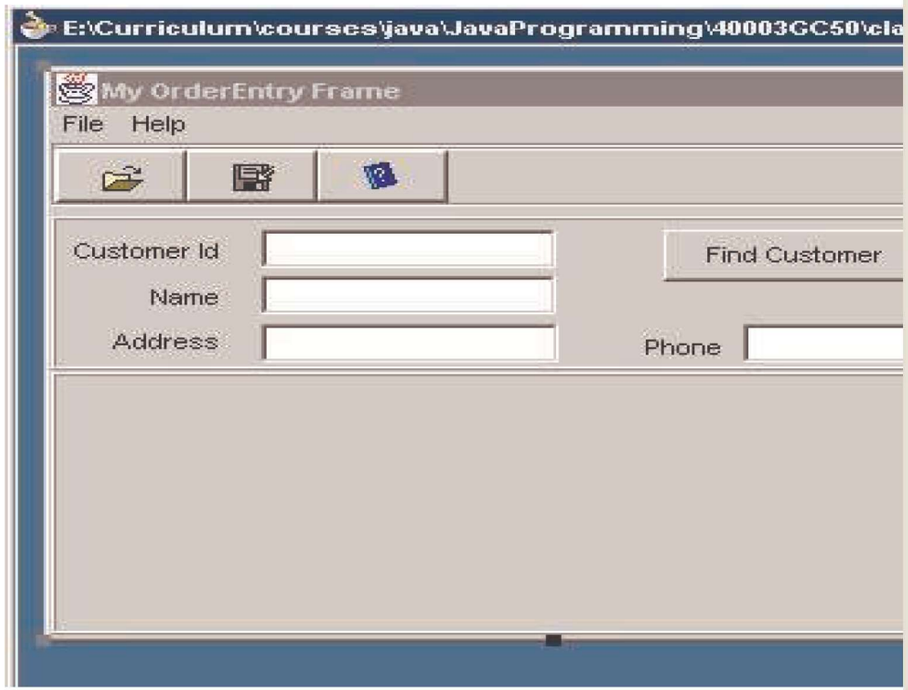
- **Rename Class**
- **Rename Field**
- **Rename Method**
- **Rename Package**
- **Rename Parameter**
- **Change Method Signature**
- **Introduce Variable**
- **Introduce Field**
- **Extract Interface**
- **Use Supertype Where Possible**
- **Move Class**
- **Duplicate Class**
- **Pull Members Up**
- **Safe Delete**

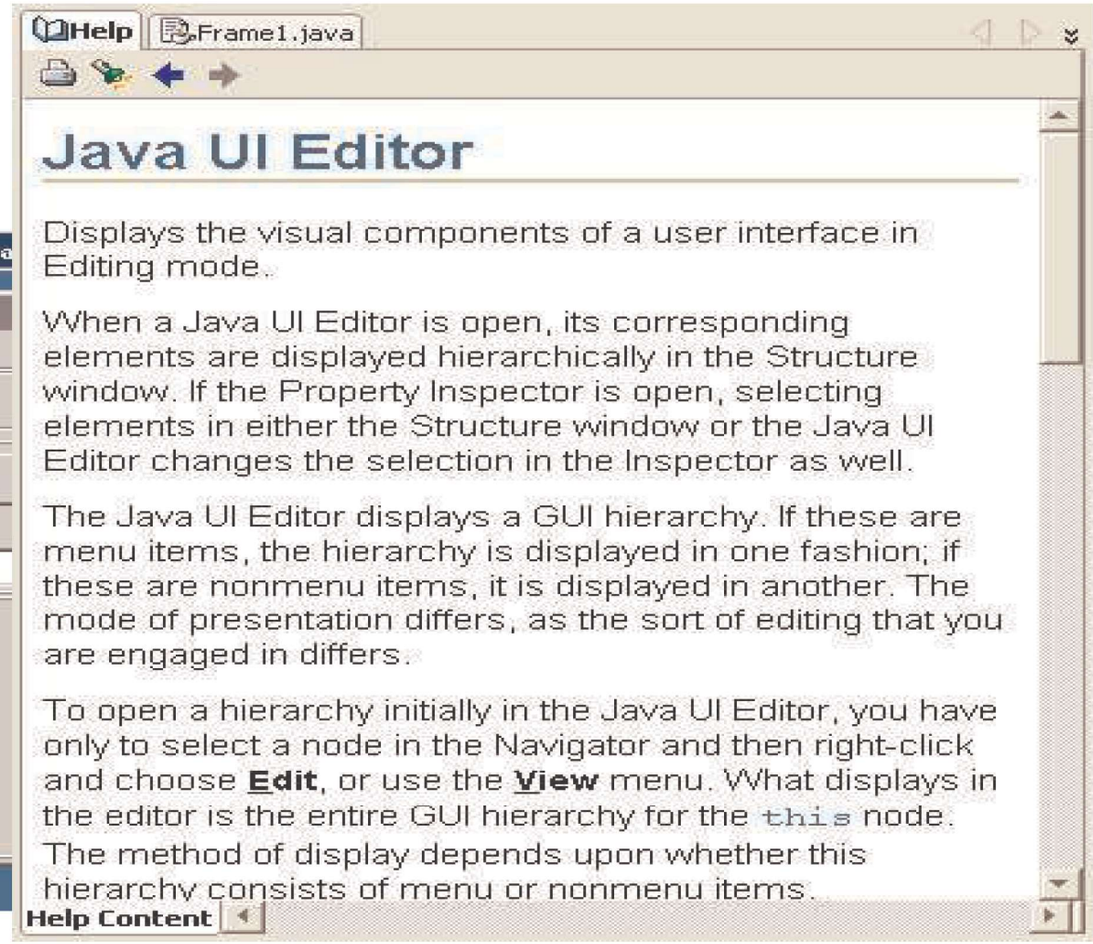
Eclipse Help System



Obtaining Help on a Topic

Use [F1] to invoke context-specific help.





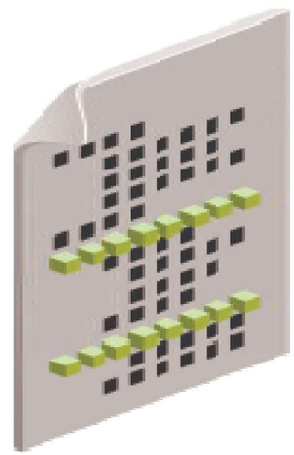
Eclipse Debugger

- Helps find and fix program errors:
  - Run-time errors
  - Logic errors
- Allows control of execution
- Allows examination of variables

Breakpoints

Setting breakpoints:

- Manage multiple breakpoints
- Manage conditional breakpoints
- Define columns displayed in window
  - Description
  - Type
  - Status
- Control scope of action
  - Global > Application > Project



Debugger Windows

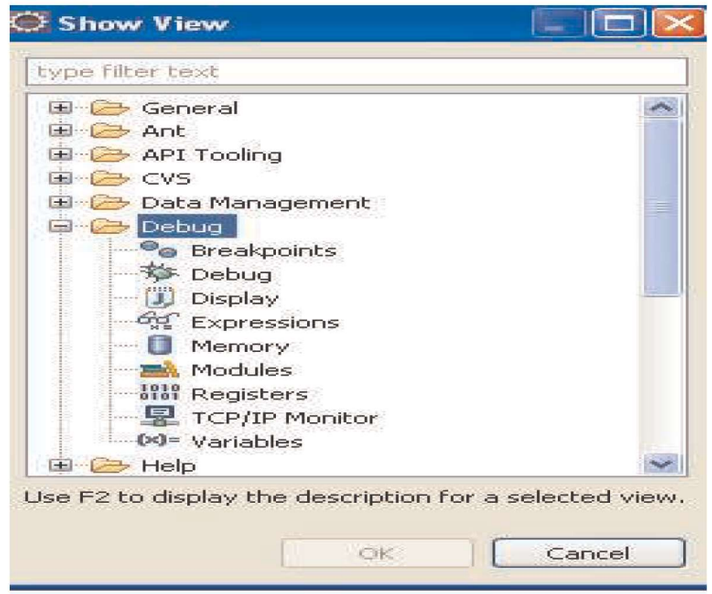
View debugging information:

- Classes: Displays list of loaded classes and status
- Watch: Evaluates and displays expressions
- Monitors: Displays information about active monitors
- Threads: Displays the names and statuses of all threads
- Smart Data: Analyzes source code near execution point
- … and more

Stepping Through a Program

Use the buttons on the debugger toolbar:

- Start the debugger.
- Resume the program.
- Step over a method call.
- Step into a method call.
- Step out of a method call.
- Step to the end of the method.
- Pause execution.
- Stop the debugger.



Summary

In this lesson, you should have learned that:

- Eclipe builds, debugs, and runs all types of Java applications
- Eclipse can be used to develop:
  - Java applications
  - Java servlets
  - JSPs
  - EJBs
- Eclipse can be used to build enterprise applications

Practice : Overview

This practice covers the following topics:

- Exploring the Eclipse 3.x IDE
- Creating an application and a project
- Populating the project with existing files