

Eclipse Basics

(with Eclipse Juno for Java)

Ming-Hsien Tsai

Eclipse

- <http://www.eclipse.org>
- Integrated development environment (IDE)
- Extensible with plugins

Without IDE

obj.???

obj.func(???)

```
add(comp1, BorderLayout.NORTH);  
add(comp2, BorderLayout.CENTER);  
cs.weightx = 1;  
comp2.add(comp3, cs);  
cs.weightx = 2;  
comp2.add(comp4, cs);
```

With IDE

obj.???

```
JFrame frame = new JFrame();  
frame.
```

- pack() : void - Window - 100 %
- action(Event evt, Object what) : boolean - Component
- add(Component comp) : Component - Container
- add(PopupMenu popup) : void - Component
- add(Component comp, int index) : Component
- add(Component comp, Object constraints) : void
- add(String name, Component comp) : Component
- add(Component comp, Object constraints, int index) : Component
- addComponentListener(ComponentListener l) : void
- addContainerListener(ContainerListener l) : void
- addFocusListener(FocusListener l) : void - Component

Press '^Space' to show Template Proposals

pack

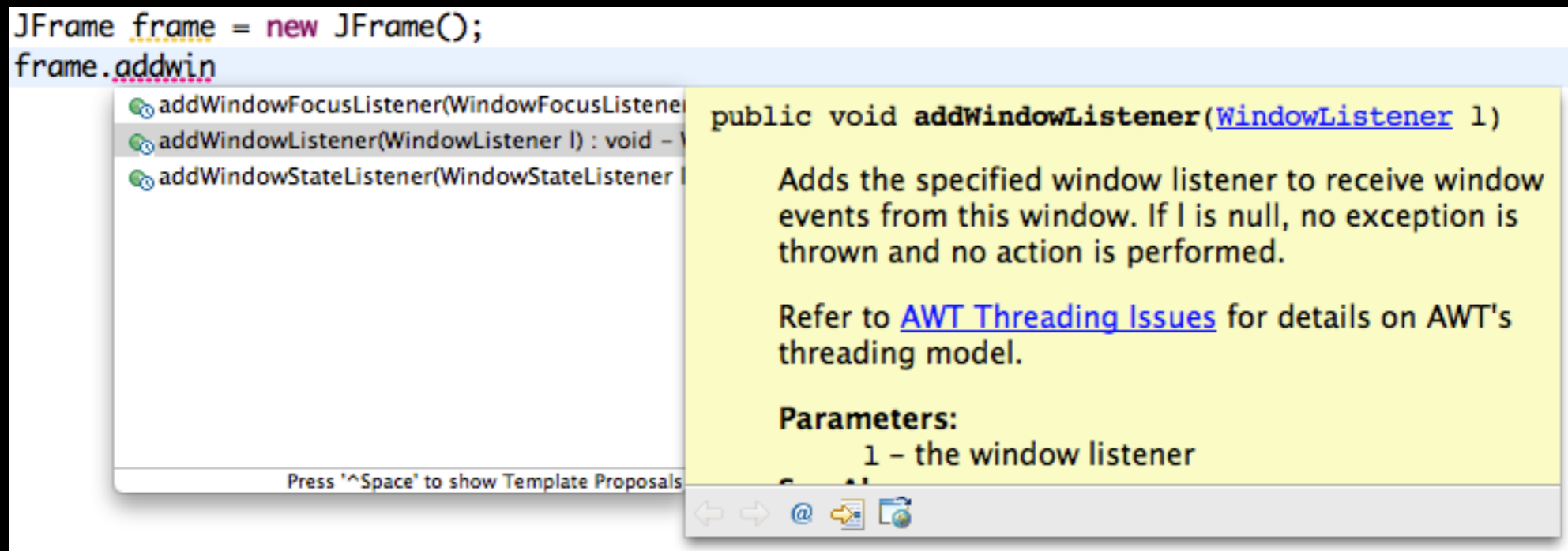
```
public void pack()
```

Causes this Window to be sized to fit the preferred size and layouts of its subcomponents. If the window and/or its owner are not yet displayable, both are made displayable before calculating the preferred size. The Window will be validated after

Press 'Tab' from proposal table or click for focus

With IDE

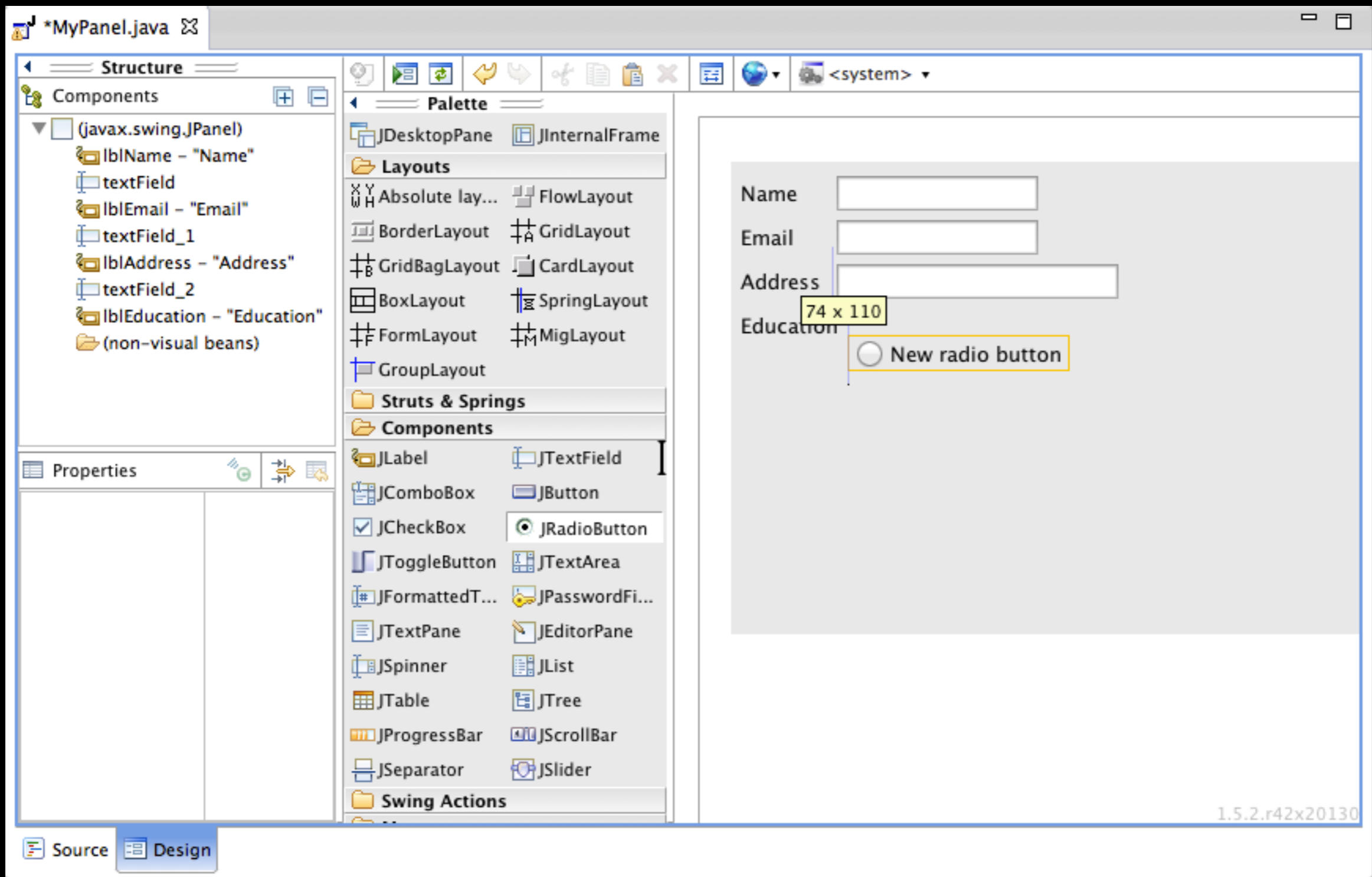
obj.func(???)



With IDE

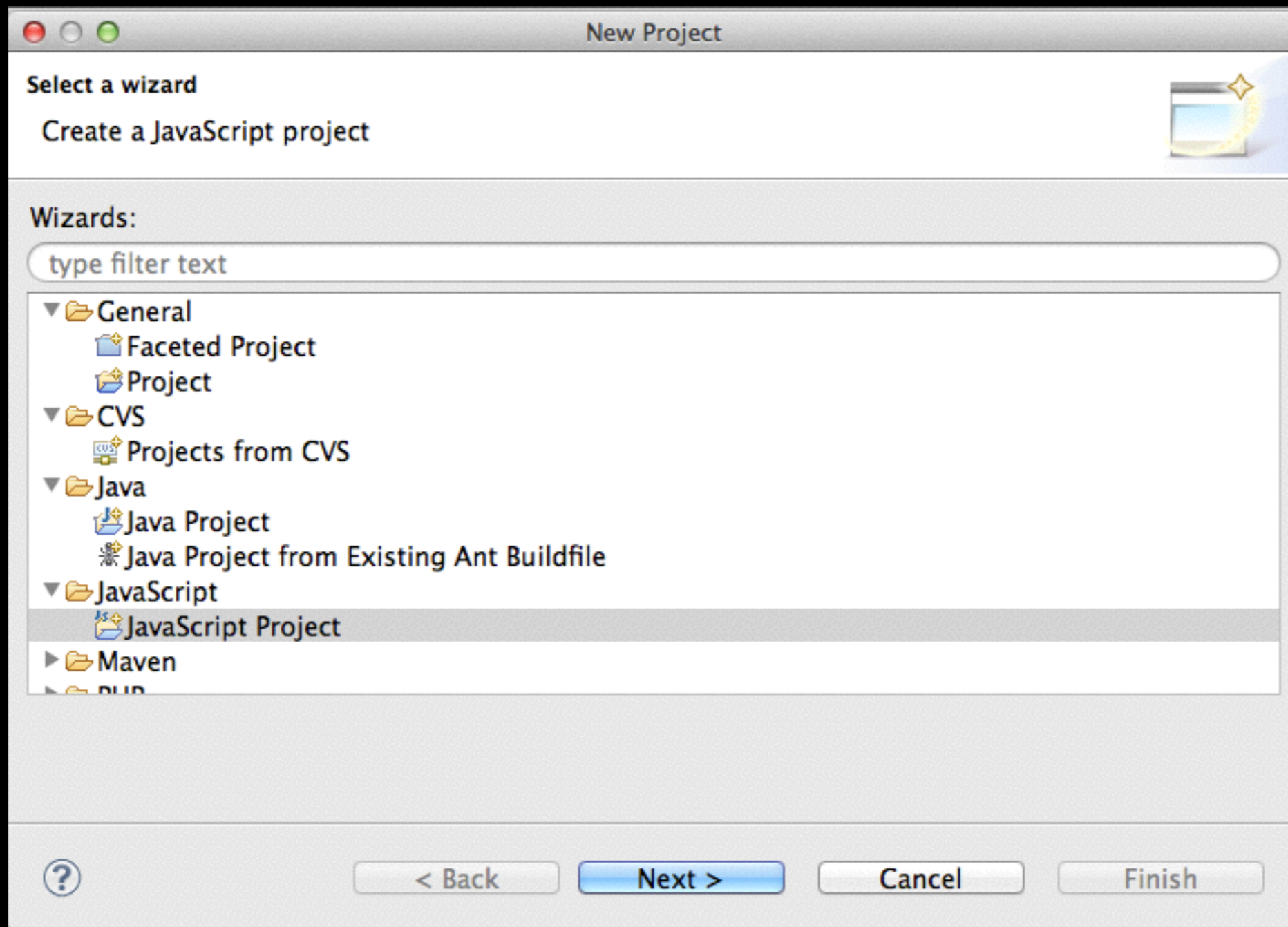
```
add(comp1, BorderLayout.NORTH);  
add(comp2, BorderLayout.CENTER);  
cs.weightx = 1;  
comp2.add(comp3, cs);  
cs.weightx = 2;  
comp2.add(comp4, cs);
```

With IDE



Create New Project

File / New / Project...



Java Perspective

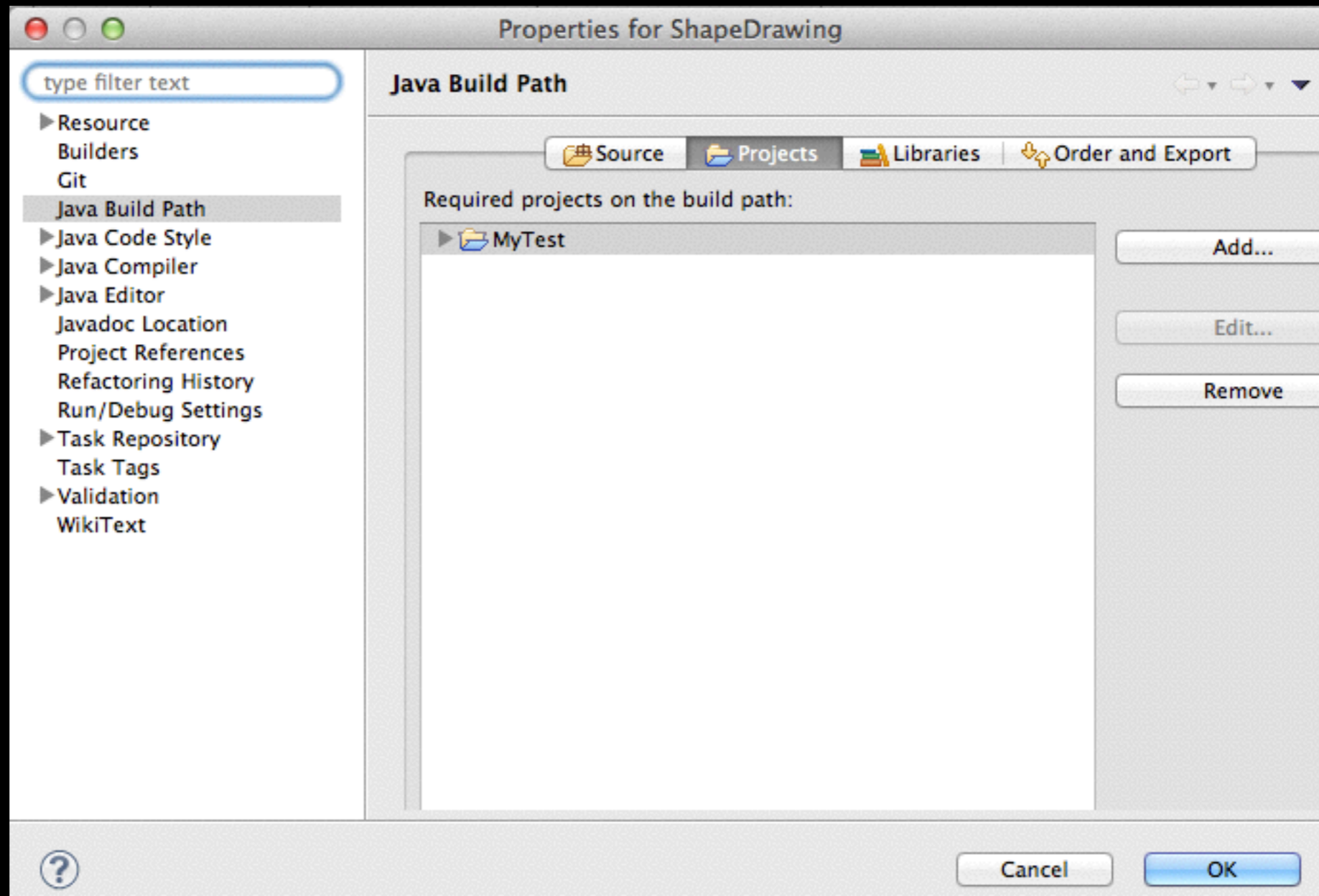
The screenshot shows the Eclipse IDE interface with the following components:

- Top Bar:** Eclipse File Edit Source Refactor Navigate Search Project Run Window Help. System tray shows 100% zoom, Wed 10:49 AM, and user profile.
- Address Bar:** Java - GOAL2/plugins/org.svrl.goal.core/source/org.svrl.goal.core/aut/Automaton.java - Eclipse - /Users/mht208/Documents/workspace
- Left Panel (Package Explorer):** Shows the project hierarchy under GOAL2 [GOAL2 develop], including plugins/org.svrl.goal.core/source/org.svrl.goal.core/aut/Automaton.java.
- Center Panel (Automaton.java):** Displays the source code for the `renamePropositions` method. The code is as follows:

```
2455     }
2456   }
2457
2458   /**
2459    * Renames simultaneously the propositions/symbols according to a specified
2460    * substitution mapping.
2461    *
2462    * @param sub
2463    *       a mapping from old proposition names to new literals
2464    * @throws IllegalArgumentException
2465    *       if this automaton does not have the old propositions, or a
2466    *       proposition is renamed to an existing proposition
2467    */
2468   public void renamePropositions(Map<String, String> sub) {
2469     /* Format the literals first. */
2470     Map<String, String> m = new HashMap<String, String>();
2471     for (String key : sub.keySet()) {
2472       try {
2473         key = atype.formatLabel(key);
2474         String value = atype.formatLabel(sub.get(key));
2475         /* Skip the substitution for literals that map to themselves. */
2476         if (!key.equals(value))
2477           m.put(key, value);
2478       } catch (IllegalArgumentException e) {
2479       }
2480     }
2481   }
```
- Right Panel (Outline):** Lists various methods of the Automaton class, including `renamePropositions`.
- Bottom Panel:** Shows toolbars for Problems, Javadoc, Declaration, Search, Console, Progress, History, Git Repositories, Tasks, Call Hierarchy, and LogCat. The status bar at the bottom indicates 'Writable', 'Smart Insert', and '2462 - 18'.

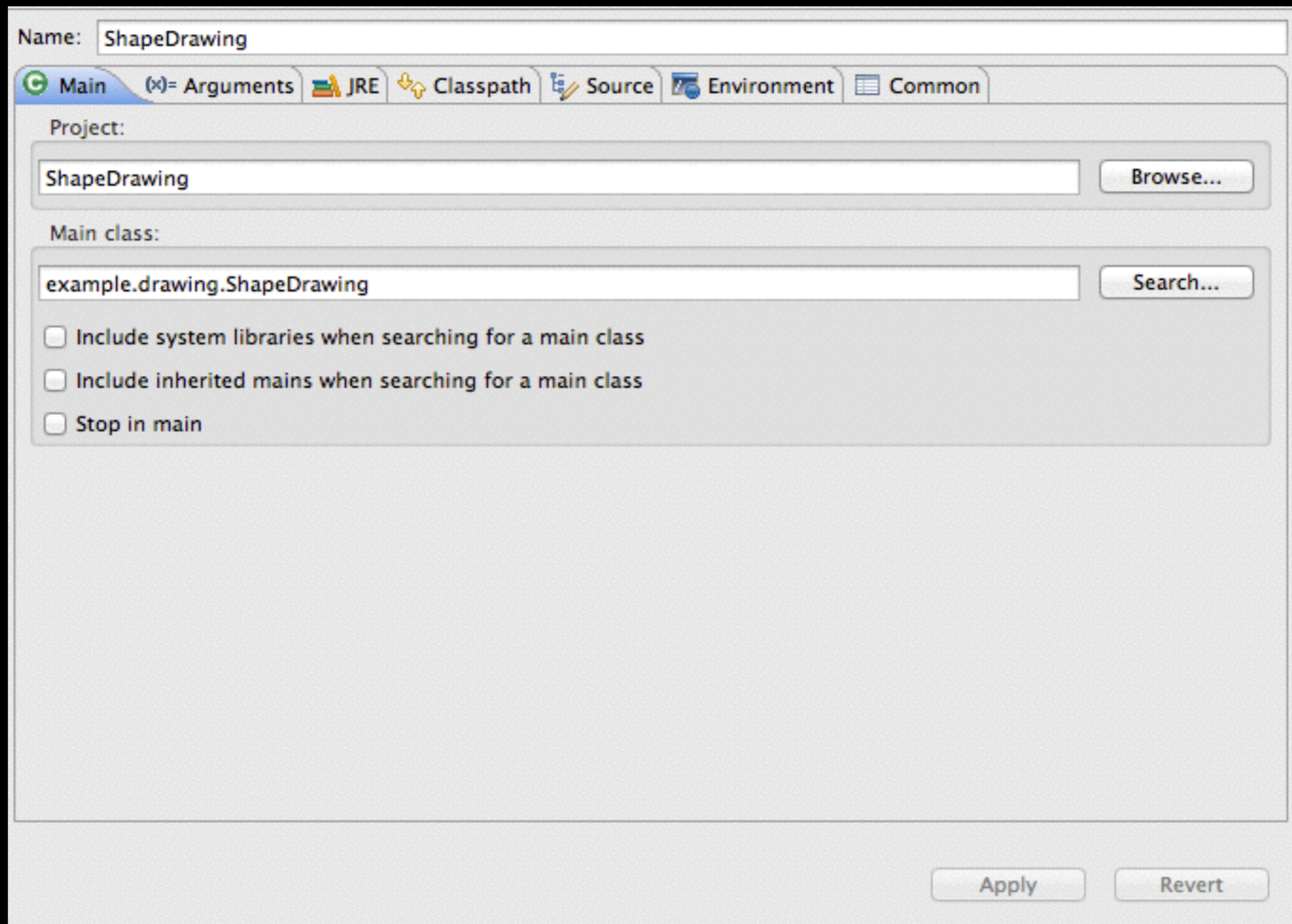
Project Reference

*Project popup menu / Properties / Java Build Path /
Projects / Add...*



Run Your Application

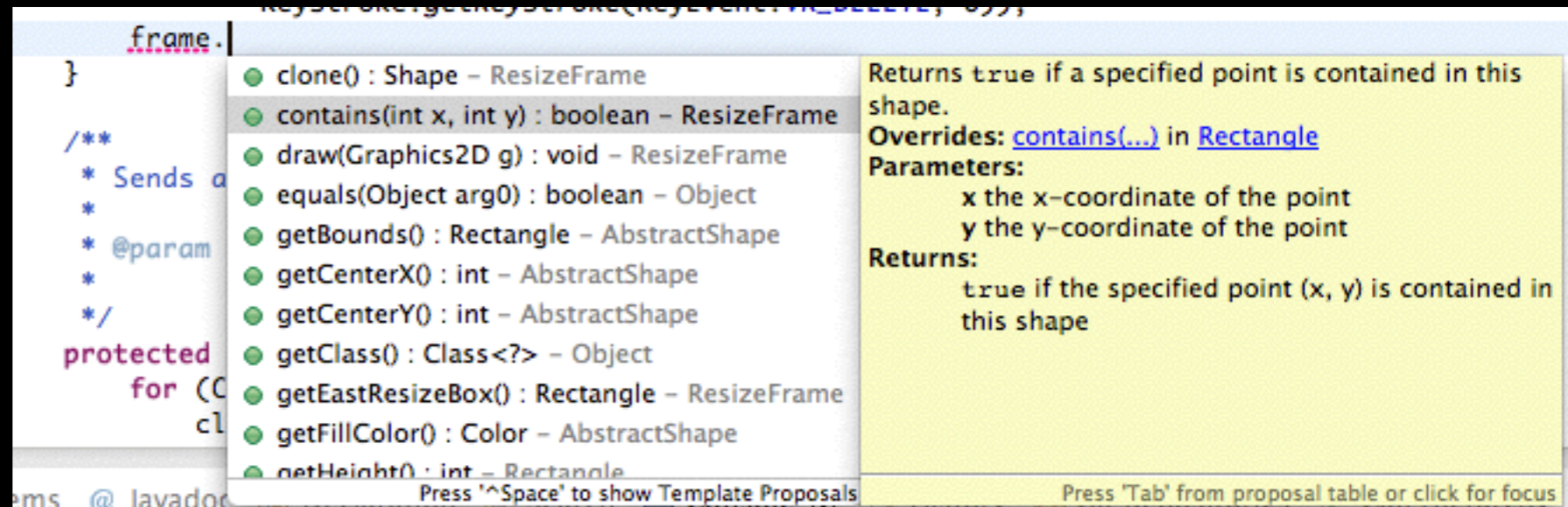
Run / Run Configurations...



Java Doc

⌘⇧J Alt-Shift-J

/**



/**

* Returns {@code true} if a specified point is contained in this shape.

*

* @param x

* the x-coordinate of the point

* @param y

* the y-coordinate of the point

* @return {@code true} if the specified point (x, y) is contained in this

* shape

*/

```
public boolean contains(int x, int y);
```

Java Doc - Tags

@author <NAME>

@version <VERSION>

@param <VARIABLE> <DESCRIPTION>

@return <DESCRIPTION>

@deprecated <DESCRIPTION>

@since <VERSION>

@throws <EXCEPTION> <DESCRIPTION>

@exception <EXCEPTION> <DESCRIPTION>

@see <CLASSPATH>

...

Java Doc - Export

File / Export / Java / Javadoc

[All Classes](#)

Packages

- [example.drawing](#)
- [example.drawing.action](#)
- [example.drawing.io](#)
- [example.drawing.menu](#)
- [example.drawing.preference](#)
- [example.drawing.shape](#)
- [undo](#)

[PreferenceDialog](#)

[PreviousWindowAction](#)

[Properties](#)

[Rectangle](#)

[RectangleBrush](#)

[RectangleCodec](#)

[RedoAction](#)

[ResizeFrame](#)

[ResizeShapeEdit](#)

[SaveAction](#)

[SelectTool](#)

[Shape](#)

[ShapeDrawing](#)

[Star](#)

[StarBrush](#)

[StarCodec](#)

[StarPolygon](#)

[StarPolygonBrush](#)

[StarPolygonCodec](#)

[StarPolygonOptionsPanel](#)

[ToolBar](#)

[UIDialog](#)

[UndoAction](#)

[Util](#)

[Window](#)

[WindowMenu](#)

[XMLUtil](#)

[Overview](#) [Package](#) **[Class](#)** [Use Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#) [FRAMES](#) [NO FRAMES](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#) [DETAIL: FIELD](#) | [CONSTR](#) | [METHOD](#)

example.drawing.shape

Interface Shape

All Superinterfaces:

- [java.lang.Cloneable](#)

All Known Implementing Classes:

- [AbstractShape](#), [Ellipse](#), [Rectangle](#), [ResizeFrame](#), [Star](#), [StarPolygon](#)

```
public interface Shape
extends java.lang.Cloneable
```

A shape is an object that can be drawn on a canvas. Every shape must be enclosed by a minimal rectangle, called frame. The location and the size of a shape may be adjusted by changing its frame. The following additional properties are defined for all shapes: line width, line color, and fill color. Note that not all the predefined properties are used by all shapes.

Author:

"Ming-Hsien Tsai"

Method Summary

Shape	clone() Makes a clone of this shape.
boolean	contains(int x, int y) Returns true if a specified point is contained in this shape.

Code Generation

Getters/Setters:

Source / Generate Getters and Setters...

Override/Implement:

Source / Overwrite/Implement Methods...

...

Navigation

Navigate / Open Declaration

F3

Navigate / Open Type Hierarchy

F4

Navigate / Open Call Hierarchy

^⌘H **Ctrl-Alt-H**

Search

Search / References / Workspace

⬆️ ⌘G **Ctrl-Shift-G**

Source

Source / Format

⌘⇧F **Ctrl-Shift-F**

Source / Organize Imports

⌘⇧O **Ctrl-Shift-O**

Source / Toggle Comment

⌘/ **Ctrl-/**

Refactor

Refactor / Rename...

⌘⇧R **Alt-Shift-R**

Refactor / Move...

⌘⇧V **Alt-Shift-V**

Source / Toggle Comment

⌘/ **Ctrl-/**

Others

Quick Fix:

⌘1 **Ctrl-1**

Shortcuts reference:

⇧⌘L **Shift-Ctrl-L**

Other Languages

- Eclipse CDT for C/C++
 - <http://www.eclipse.org/cdt/>
- Eclipse PDT for PHP
 - <http://projects.eclipse.org/projects/tools.pdt>
- Eclipse JSDT for Javascript
 - <http://www.eclipse.org/webtools/jsdt/>
- PyDev for Python
 - <http://marketplace.eclipse.org/content/pydev-python-ide-eclipse/metrics#.UkJQuxY5SfQ>

Other Features

(may need third-party plugins)

- Debugging
- Drawing UML diagrams
- Integration with project management tools
- Verifying Java programs