

# Eclipse Basics

(with Eclipse Juno for Java)

Ming-Hsien Tsai

# Eclipse

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

# History

<b>Version Name</b>	<b>Date</b>	<b>Platform Version</b>
N/A	21 June 2004	3.0
N/A	28 June 2005	3.1
Callisto	30 June 2006	3.2
Europa	29 June 2007	3.3
Ganymede	25 June 2008	3.4
Galileo	24 June 2009	3.5
Helios	23 June 2010	3.6
Indigo	22 June 2011	3.7
Juno	27 June 2012	3.8 and 4.2
Kepler	26 June 2013	4.3
Luna	25 June 2014	4.4
Mars	24 June 2015	4.5
Neon	22 June 2016	4.6
Oxygen	June 2017 (planned)	4.7

Source: [https://en.wikipedia.org/wiki/Eclipse\\_\(software\)](https://en.wikipedia.org/wiki/Eclipse_(software))

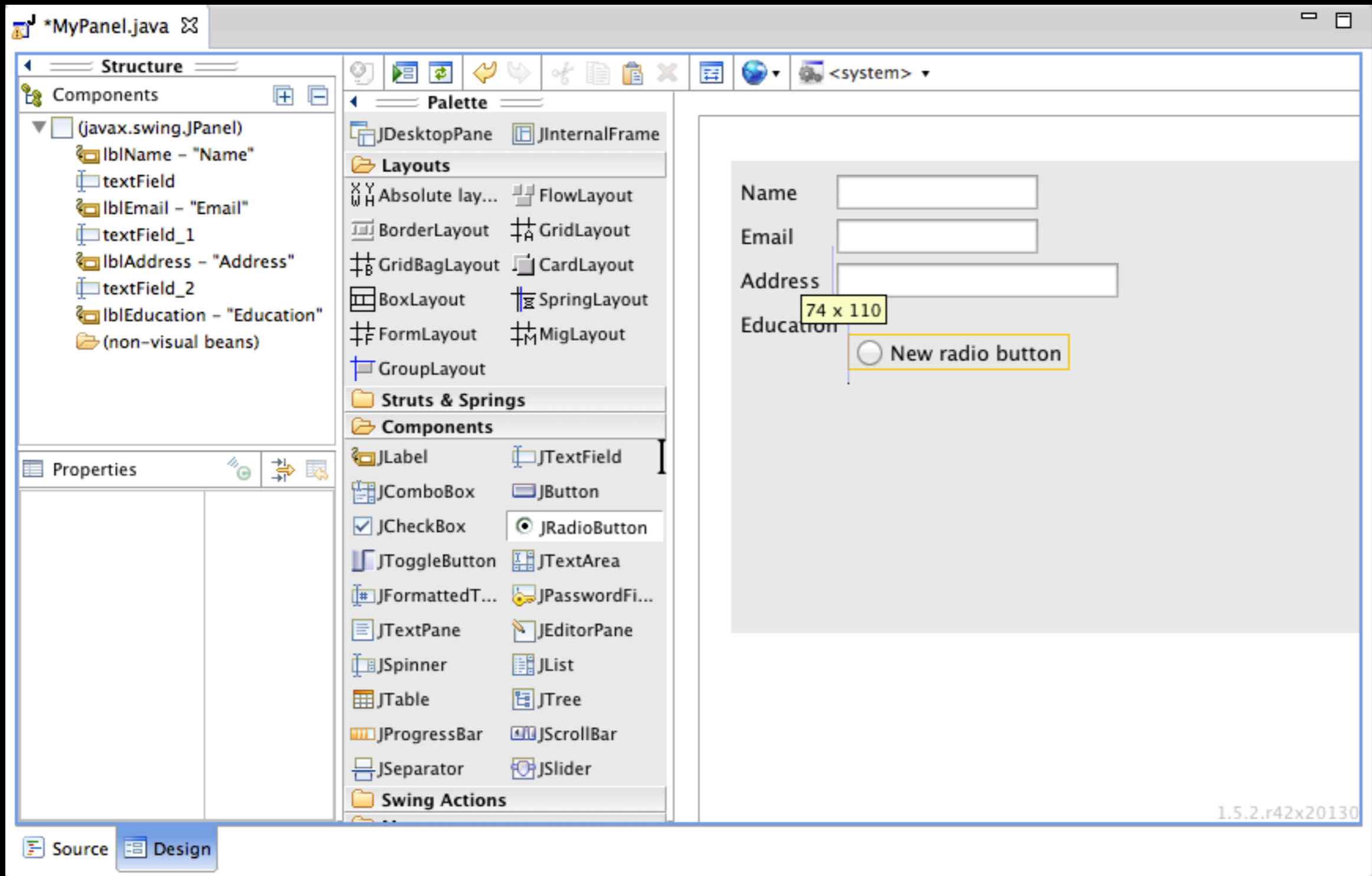
# Without IDE

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

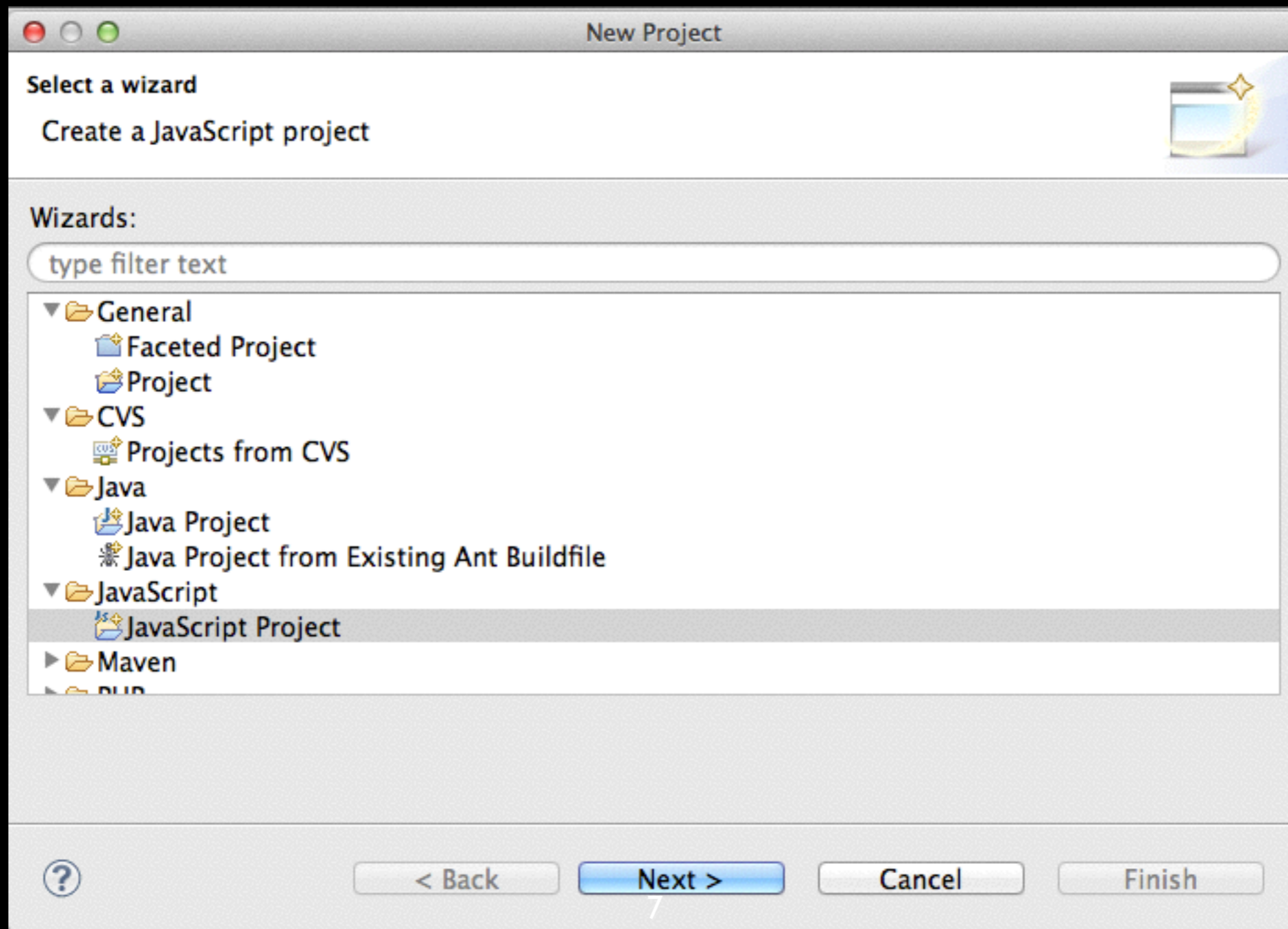


# Workspace

- Where your projects are stored
- Multiple workspaces are allowed

# Create Project

*File / New / Project...*



# Perspective - Java

The screenshot displays the Eclipse IDE interface in Perspective mode. The top menu bar includes File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, and Help. The title bar shows the current file: Java - GOAL2/plugins/org.svrl.goal.core/source/org.svrl.goal.core/aut/Automaton.java. The main workspace is divided into three panes:

- Package Explorer (Left):** Shows the project hierarchy. The current project is GOAL2 [GOAL2 develop], with a sub-project plugins/org.svrl.goal.core. The file Automaton.java is selected under the org.svrl.goal.core.aut package.
- Code Editor (Center):** Displays the source code for Automaton.java. The method `renamePropositions` is visible, which takes a `Map<String, String>` parameter and returns `void`. The code includes Javadoc comments and a `try-catch` block for handling `IllegalArgumentException`.
- Outline (Right):** Lists the methods and classes in the current file, including `renamePropositions`, `removeAutomatonListener`, `removeComponent`, and others.

The bottom status bar shows various toolbars and indicators, including Problems, Javadoc, Declaration, Search, Console, Progress, History, Git Repositories, Tasks, Call Hierarchy, and LogCat. The current cursor position is at line 2462, column 18.



# Perspective - Browsing

The screenshot displays the Eclipse IDE in the Java Browsing perspective. The title bar indicates the current file is `CPALib.java` within the `CPAchecker/src/org/sosy_lab/cpachecker/cmdline` package. The interface is divided into several panes:

- Projects:** Shows the project structure with folders like `src`, `bin`, and various JAR files.
- Packages:** Lists the package hierarchy, including `org.sosy_lab.cpachecker.cmdline` and its sub-packages.
- Types:** Lists the classes and interfaces available in the current package, such as `CmdLineArguments`, `CPALib`, `CPAMain`, `CPASelfCheck`, `CPATop`, `ForceTerminationOnShutdown`, and `ShutdownHook`.
- Members:** Shows the members of the selected class, including `ERROR_OUTPUT : PrintStream`, `ERROR_EXIT_CODE : int`, `cpachecker : CPAchecker`, `shutdownNotifier : ShutdownNotifier`, `shutdownHook : ShutdownHook`, `forcedExitOnShutdown : ShutdownRequestList`, and `initialize(String[]) : void`.

The main editor shows the source code of `CPALib.java`, with the `initialize` method highlighted. The code includes comments and error handling for command line arguments and configuration files.

```
62 static ShutdownRequestListener forcedExitOnShutdown = null;
63
64 @SuppressWarnings("resource")
65 // We don't close LogManager
66 public static void initialize(String[] args) {
67     // initialize various components
68     Configuration cpaConfig = null;
69     LogManager logManager = null;
70     try {
71         try {
72             Pair<Configuration, String> p = createConfiguration(args);
73             cpaConfig = p.getFirst();
74         } catch (InvalidCmdlineArgumentException e) {
75             ERROR_OUTPUT.println("Could not process command line arguments: " + e.getMessage());
76             System.exit(ERROR_EXIT_CODE);
77         } catch (IOException e) {
78             ERROR_OUTPUT.println("Could not read config file " + e.getMessage());
79             System.exit(ERROR_EXIT_CODE);
80         }
81
82         logManager = new BasicLogManager(cpaConfig);
83
84     } catch (InvalidConfigurationException e) {
85         ERROR_OUTPUT.println("Invalid configuration: " + e.getMessage());
86         System.exit(ERROR_EXIT_CODE);
87         return;
88     }
89     cpaConfig.enableLogging(logManager);
```

At the bottom of the IDE, the status bar shows "Writable", "Smart Insert", and "69 : 13".

# Perspective - Debug

The screenshot displays the Eclipse IDE in the Debug perspective. The main editor shows the source code for `CPALib.java`, with the `initialize` method highlighted. The code includes a `try-catch` block for handling `InvalidCmdlineArgumentException`.

```
62 static ShutdownRequestListener forcedExitOnShutdown = null;
63
64 @SuppressWarnings("resource")
65 // We don't close LogManager
66 public static void initialize(String[] args) {
67     // initialize various components
68     Configuration cpaConfig = null;
69     LogManager logManager = null;
70     try {
71         try {
72             Pair<Configuration, String> p = createConfiguration(args);
73             cpaConfig = p.getFirst();
74         } catch (InvalidCmdlineArgumentException e) {
75             ERROR_OUTPUT.println("Could not process command line arguments: " + e.getMessage());
76         }
77     }
78 }
```

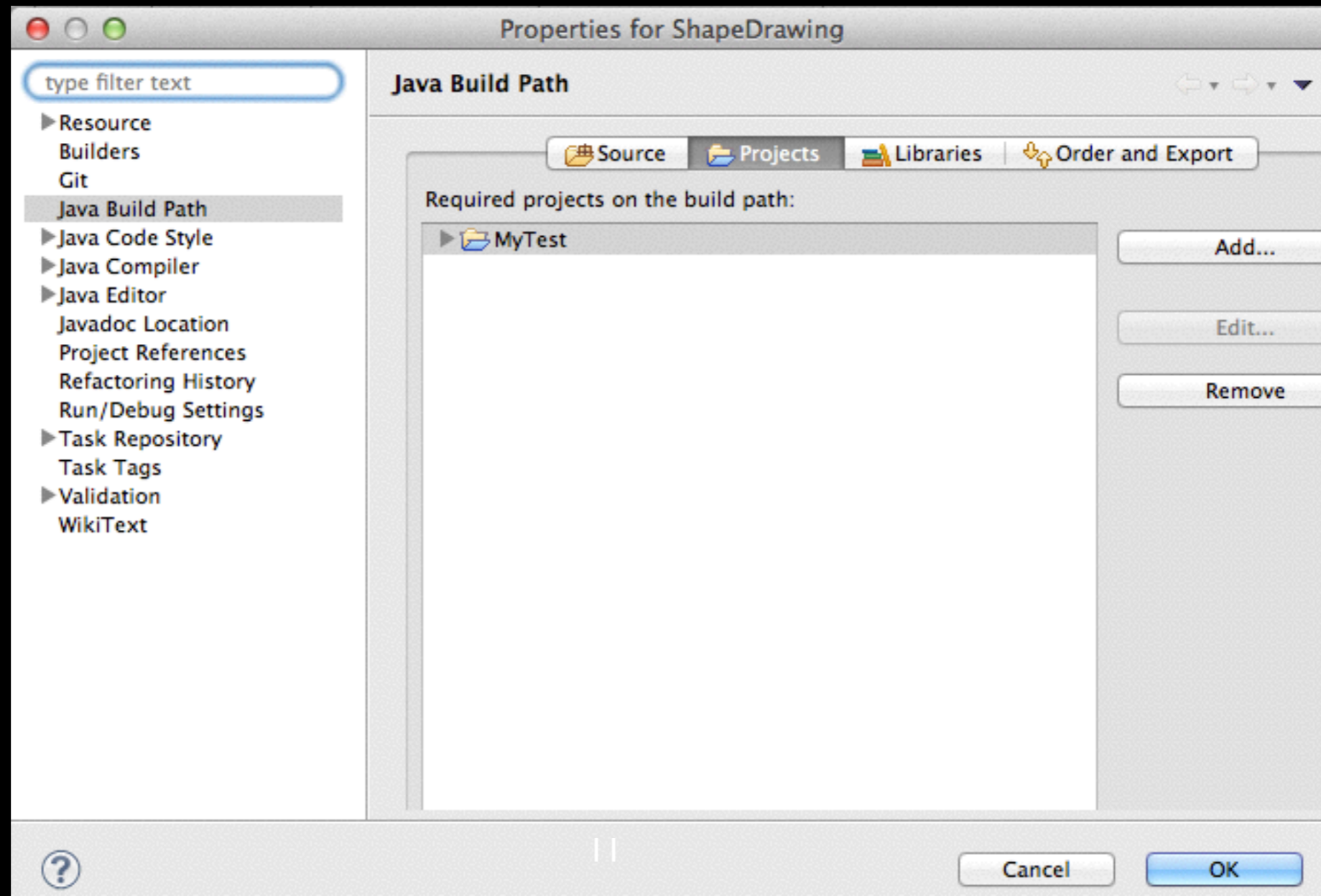
The Outline view on the right shows the class hierarchy for `org.sosy_lab.cpagechecker.cmdline`, including `CPALib`, `BootstrapOptions`, and `MainOptions`. The `CPALib` class contains several static fields and methods, such as `cpachecker`, `ERROR_EXIT_CODE`, `ERROR_OUTPUT`, `forcedExitOnShutdown`, `shutdownHook`, `shutdownNotifier`, and `createConfiguration`.

The Console view at the bottom shows the message: "No consoles to display at this time."

The status bar at the bottom indicates the current line is 69 of 13, and the editor is in Writable mode with Smart Insert enabled.

# Configure Project

*Project popup menu / Properties*

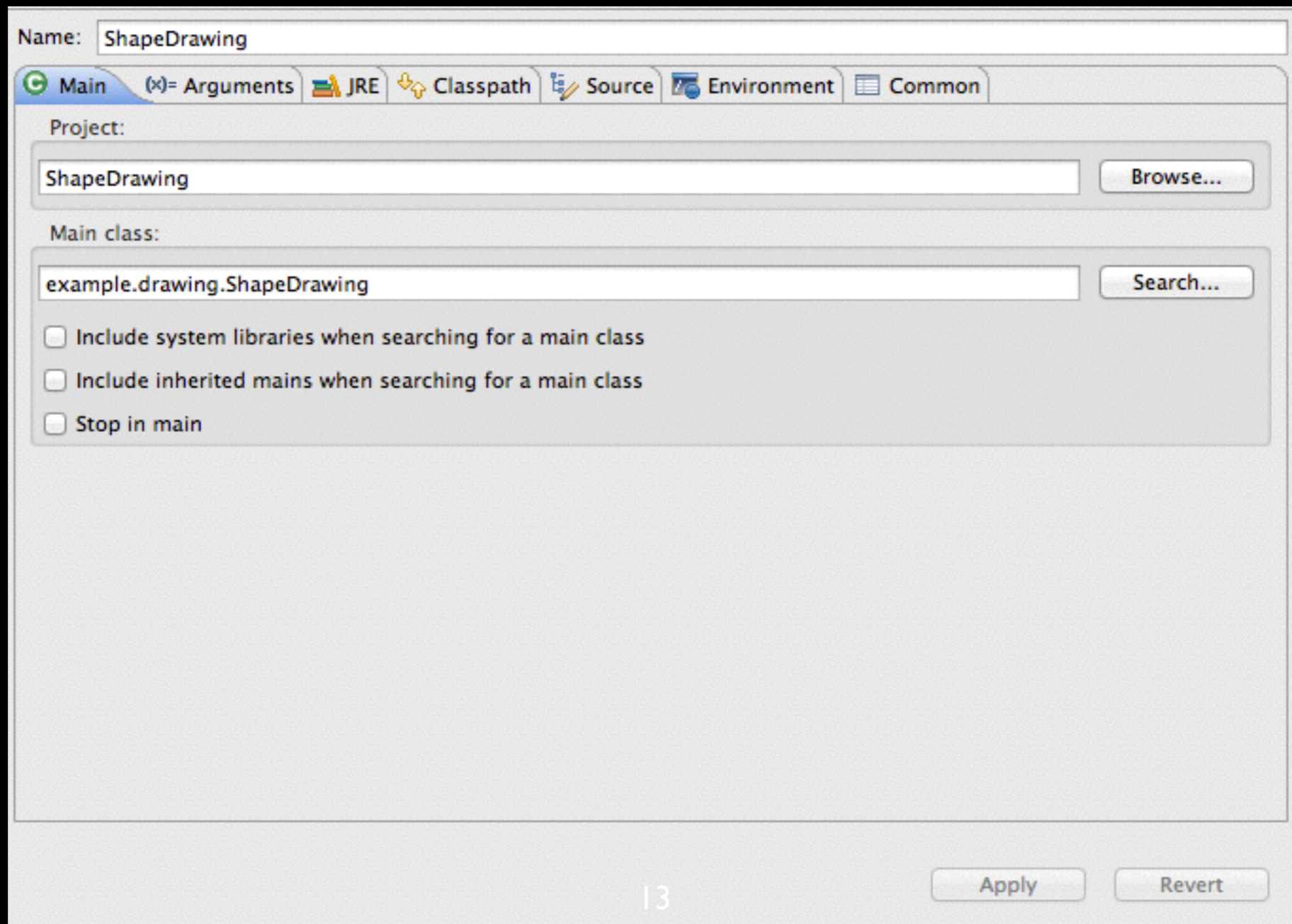


# Build Project

- Java projects can be built automatically
- Build tools:
  - GNU Make
  - Apache Ant (with Ivy)
  - Apache Maven
  - Gradle
  - ...

# Run Project

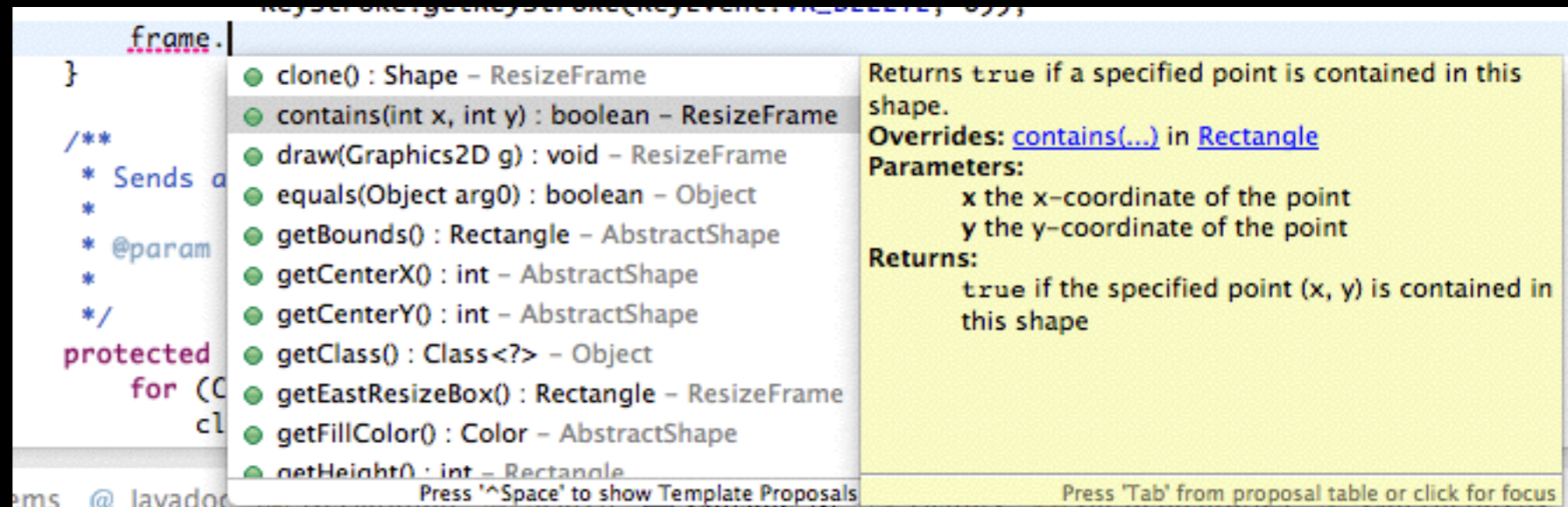
*Run / Run Configurations...*



# 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

<a href="#">Shape</a>	<a href="#">clone()</a> Makes a clone of this shape.
boolean	<a href="#">contains(int x, int y)</a> Returns true if a specified point is contained in this shape.



# Other Documentation Generators

- Oxygen
  - C, Objective-C, C#, PHP, Java, Python, IDL (Corba, Microsoft, and UNO/OpenOffice flavors), Fortran, VHDL, Tcl
- Sphinx
  - Python, C/C++
- ScalaDoc
- ocamlDoc

More generators can be found in [https://en.wikipedia.org/wiki/Comparison\\_of\\_documentation\\_generators](https://en.wikipedia.org/wiki/Comparison_of_documentation_generators)

# 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

  **Ctrl-Shift-F**

## Source / Organize Imports

  **Ctrl-Shift-O**

## Source / Toggle Comment

 **Ctrl-/**

```
public static <T> Set<T> inte(List<T> l1, List<T> l2) {
    Set<T> res = new HashSet<T>();

    for(int i=0;i<l1.size();i++){
        T o1 = l1.get(i);
        for(int j=0;j<l2.size();j++){
            T o2=l2.get(j);
            if(o1.equals(o2)){
                res.add(o1);
                break;
            }
        }
    }

    return res;
}
```

# Refactor

*Refactor / Rename...*

**⌘⌥R**    **Alt-Shift-R**

*Refactor / Move...*

**⌘⌥V**    **Alt-Shift-V**

# 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>
- Scala IDE for Scala
  - <http://scala-ide.org>



# Other Features

(may need third-party plugins)

- Debugging
- UML diagrams and code generation
  - UML Designer, UML to Java code generator
- Task management
  - Mylyn
- Issue tracking
  - Bugzilla, JIRA, Redmine, ...

# Other Features

(may need third-party plugins)

- Continuous integration
  - Eclipse Hudson
- Program verification
  - Java PathFinder, Leon, EpiSpin