

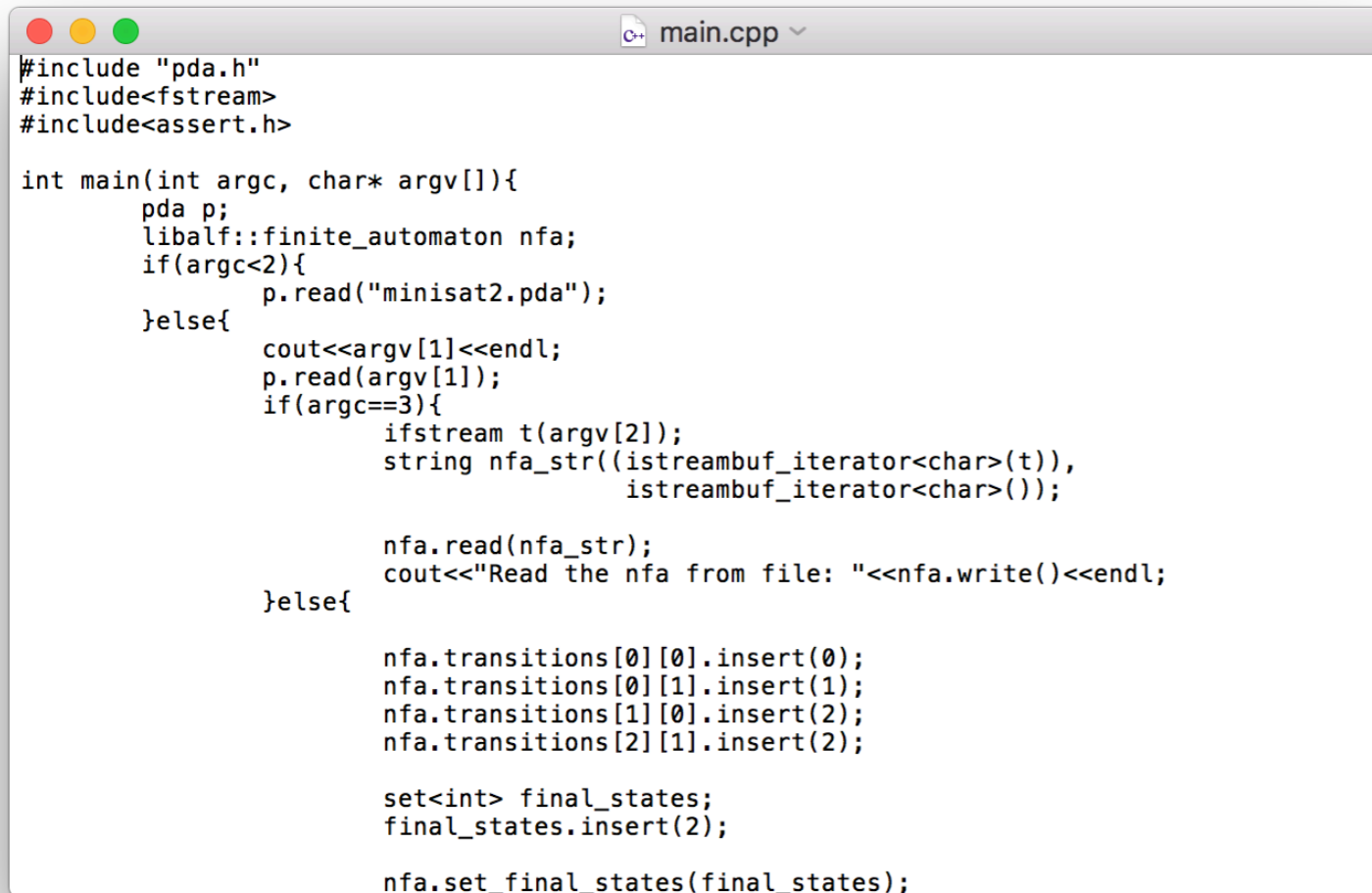
Eclipse (version Oxygen)

Ming-Hsien Tsai
Academia Sinica

SDM 2020

Writing Code With...

Text Editors



```
#include "pda.h"
#include<fstream>
#include<assert.h>

int main(int argc, char* argv[]){
    pda p;
    libalf::finite_automaton nfa;
    if(argc<2){
        p.read("minisat2.pda");
    }else{
        cout<<argv[1]<<endl;
        p.read(argv[1]);
        if(argc==3){
            ifstream t(argv[2]);
            string nfa_str((istreambuf_iterator<char>(t)),
                istreambuf_iterator<char>());

            nfa.read(nfa_str);
            cout<<"Read the nfa from file: "<<nfa.write()<<endl;
        }else{

            nfa.transitions[0][0].insert(0);
            nfa.transitions[0][1].insert(1);
            nfa.transitions[1][0].insert(2);
            nfa.transitions[2][1].insert(2);

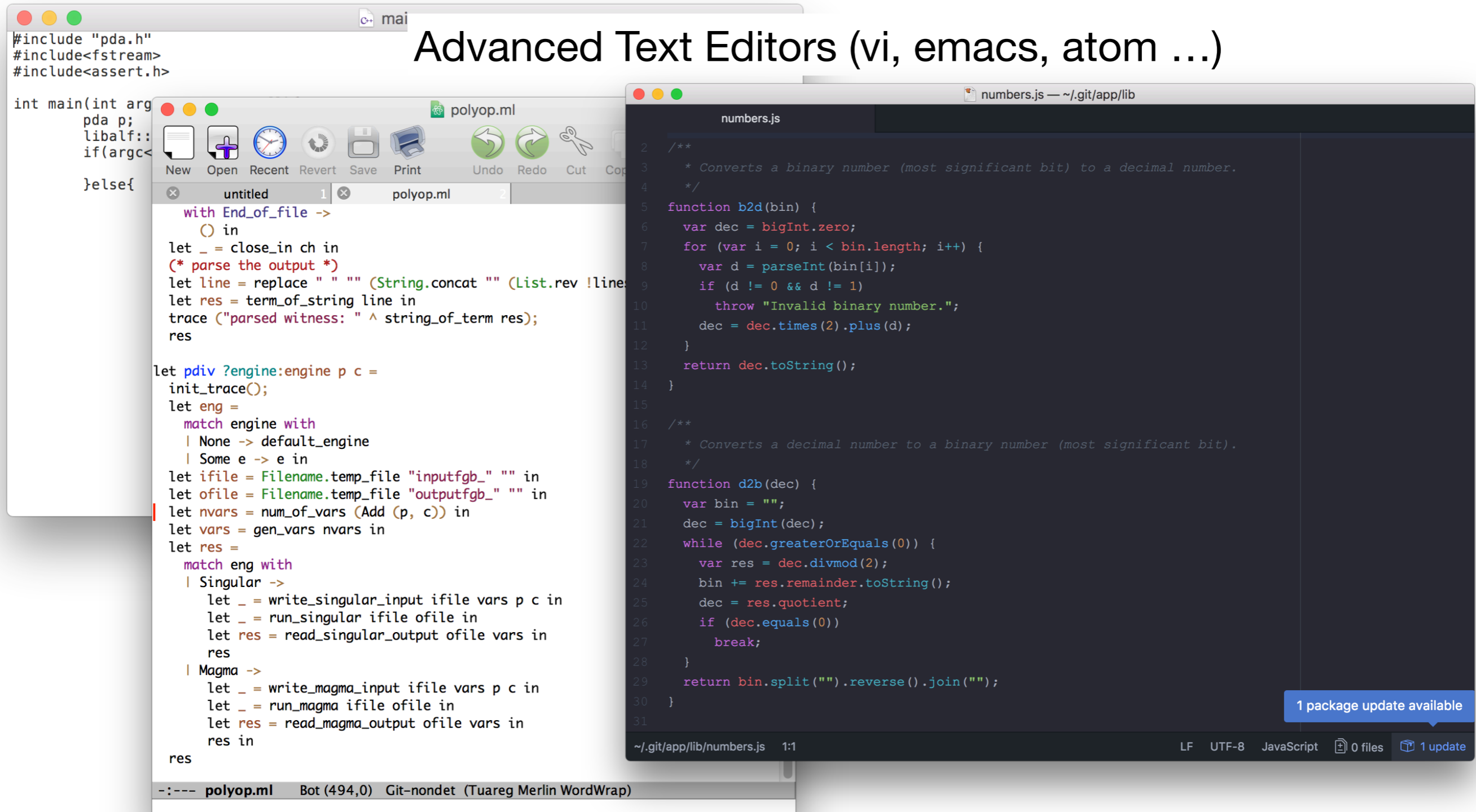
            set<int> final_states;
            final_states.insert(2);

            nfa.set_final_states(final_states);
```

Writing Code With...

Text Editors

Advanced Text Editors (vi, emacs, atom ...)

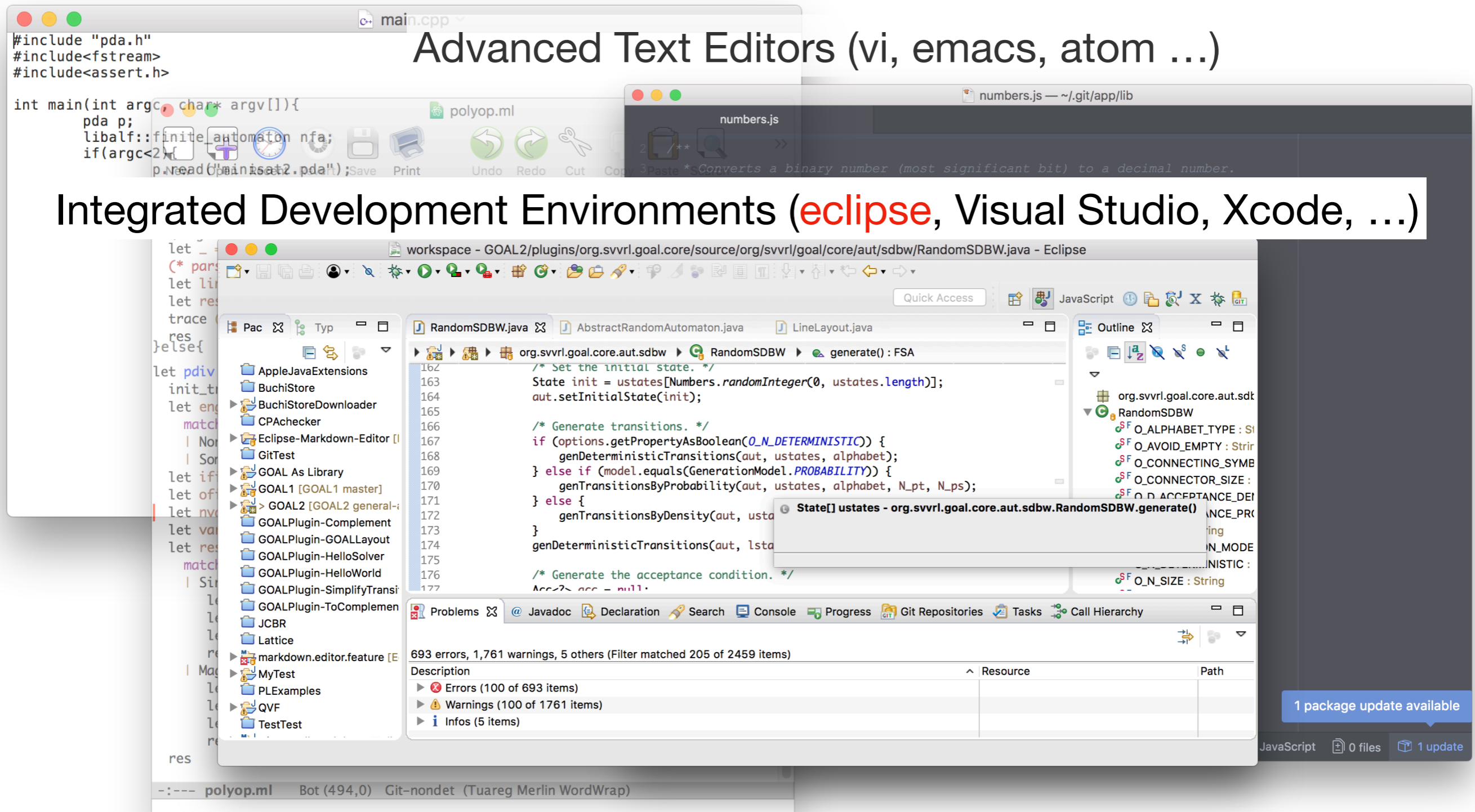


Writing Code With...

Text Editors

Advanced Text Editors (vi, emacs, atom ...)

Integrated Development Environments (eclipse, Visual Studio, Xcode, ...)

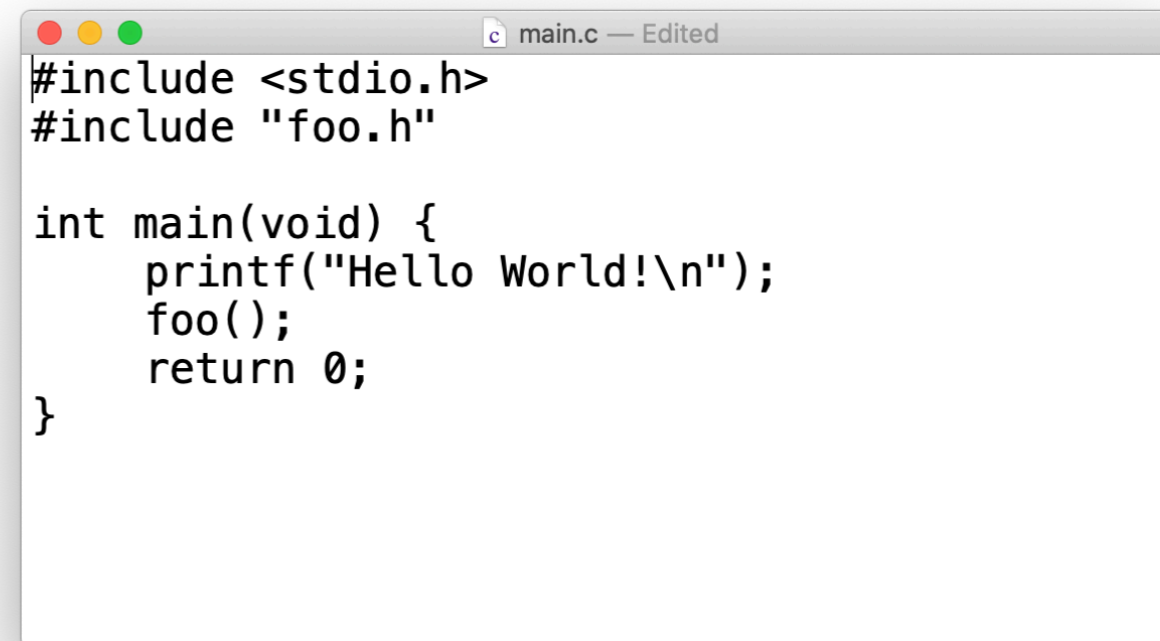


Writing Code With...



Text Editors

- Notepad (Windows), TextEdit (Mac), ...
 - Basic text editing
 - Basic search and replacement
 - Display with various fonts

A screenshot of a text editor window titled "main.c — Edited". The window contains the following C code:

```
#include <stdio.h>
#include "foo.h"

int main(void) {
    printf("Hello World!\n");
    foo();
    return 0;
}
```

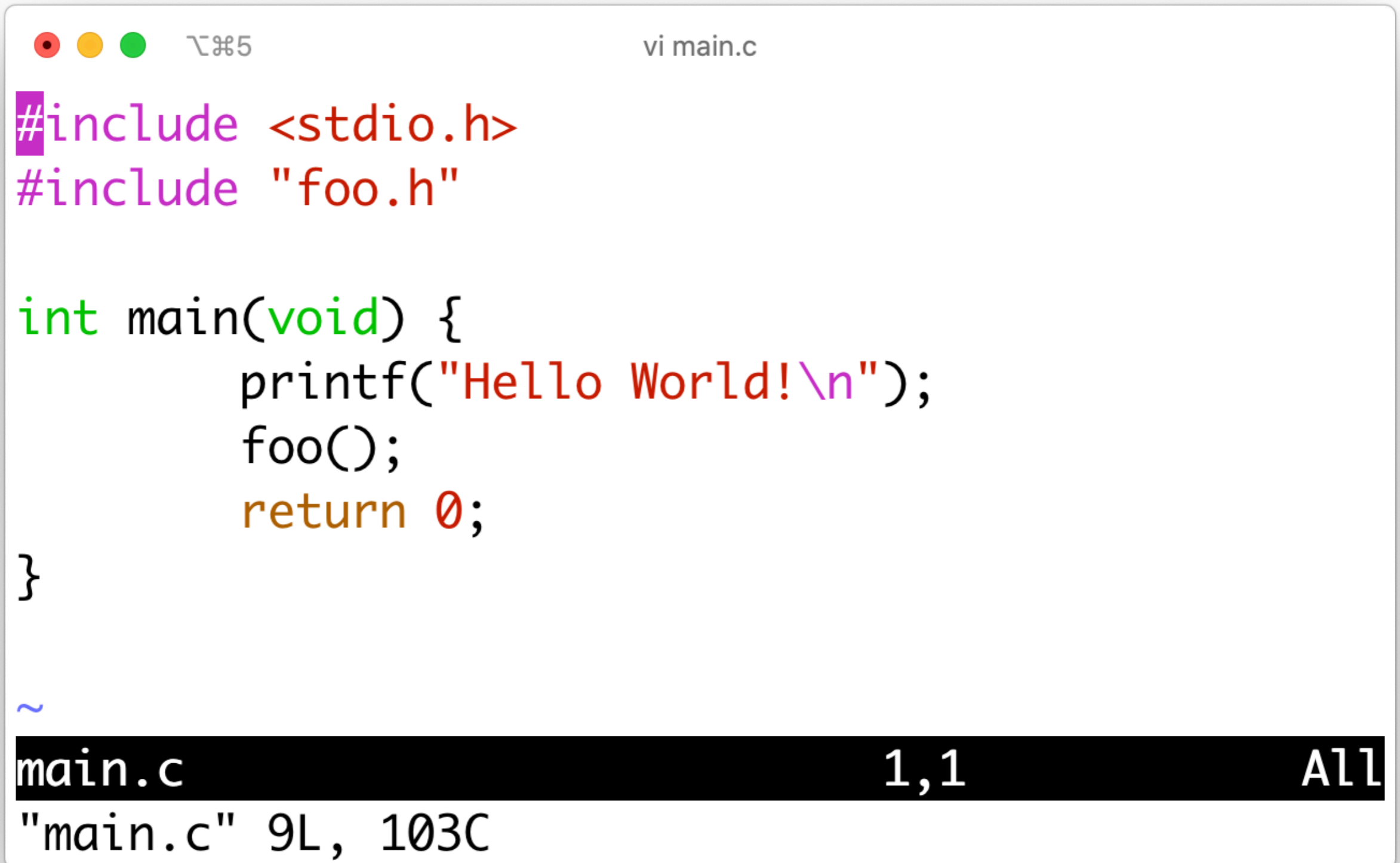
Advanced Text Editors

- Emacs, vi/vim, atom, notepad++, ...
 - Syntax highlight
 - Auto-indent
 - Search and replacement using regular expressions
 - Build source code

Vi / Vim

- A powerful editor on UNIX-like systems
- Two modes: command mode and insert mode
 - Press i (a, o, ...) to enter insert mode
 - Press ESC to enter command mode
- In command mode
 - Type :w and press ENTER to save a file
 - Type :q (or :qw, :q!) and press ENTER to quit vi/vim

Vi / Vim



The image shows a terminal window with a Vi/Vim editor. The window title is "vi main.c". The editor displays a C program with the following code:

```
#include <stdio.h>
#include "foo.h"

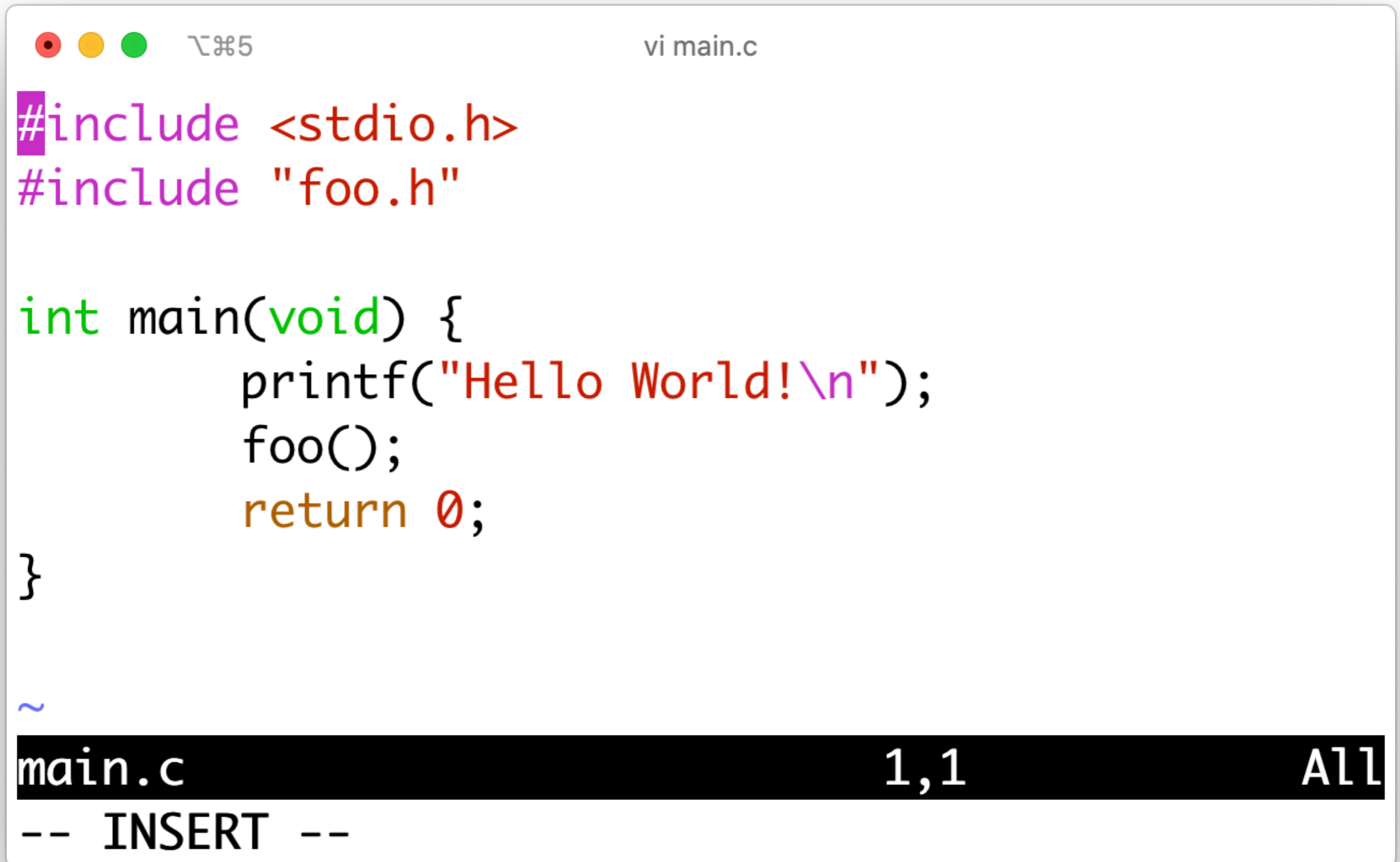
int main(void) {
    printf("Hello World!\n");
    foo();
    return 0;
}

~
```

The status bar at the bottom of the editor shows "main.c" on the left, "1,1" in the center, and "All" on the right. Below the status bar, the text "main.c" 9L, 103C is displayed.

Command Mode

Vi / Vim



The image shows a terminal window with a Vi/Vim editor. The window title is "vi main.c". The editor content is a C program with the following code:

```
#include <stdio.h>
#include "foo.h"

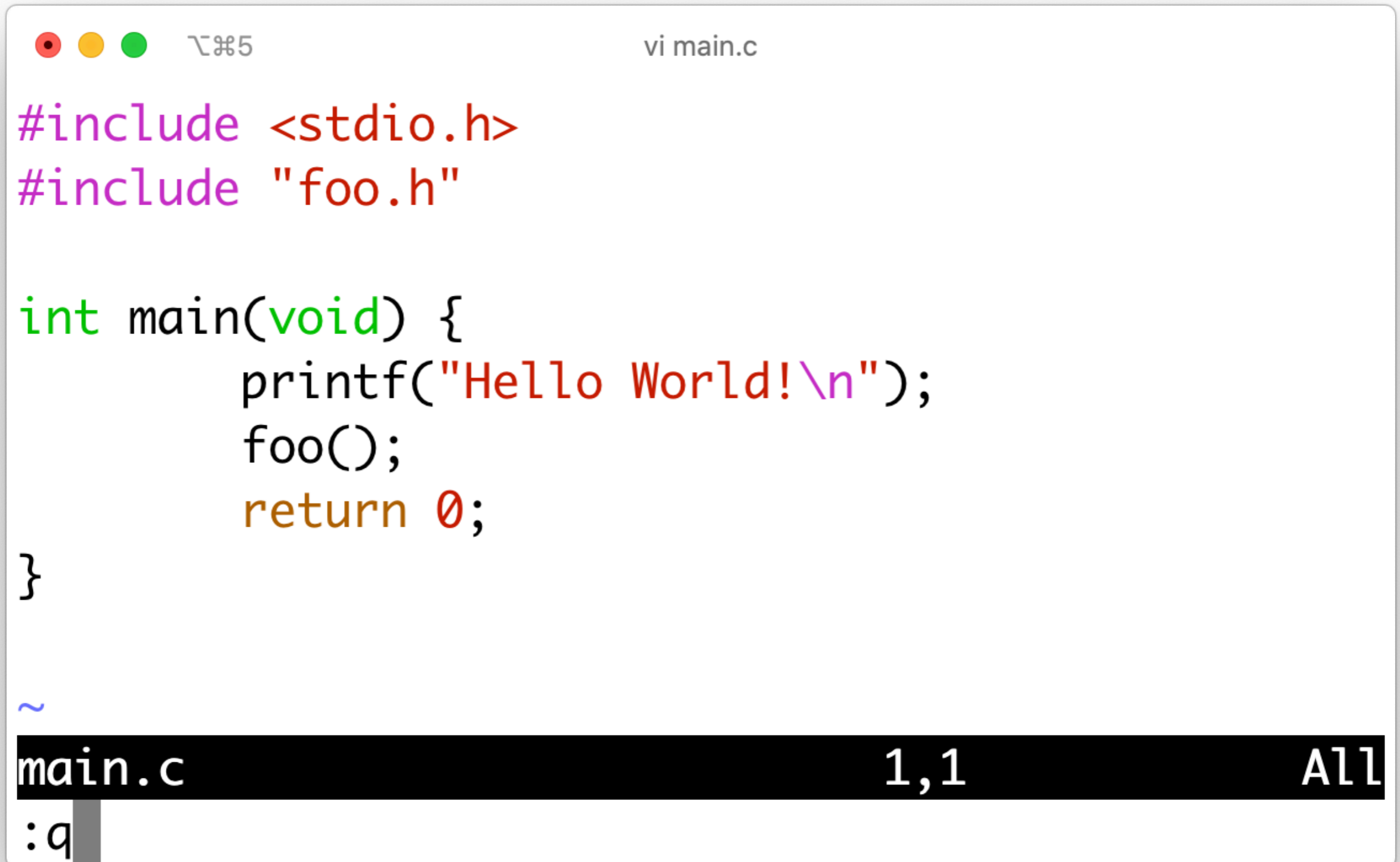
int main(void) {
    printf("Hello World!\n");
    foo();
    return 0;
}

~
```

The status bar at the bottom of the editor shows "main.c", "1,1", and "All". Below the status bar, the text "-- INSERT --" is displayed, indicating that the editor is in Insert Mode.

Insert Mode

Vi / Vim



The image shows a terminal window with a Vi/Vim editor. The window title is "vi main.c". The code being edited is a simple C program that prints "Hello World!". The editor is in normal mode, and the command prompt ":" is visible at the bottom left. The status bar at the bottom right shows "main.c", "1,1", and "All".

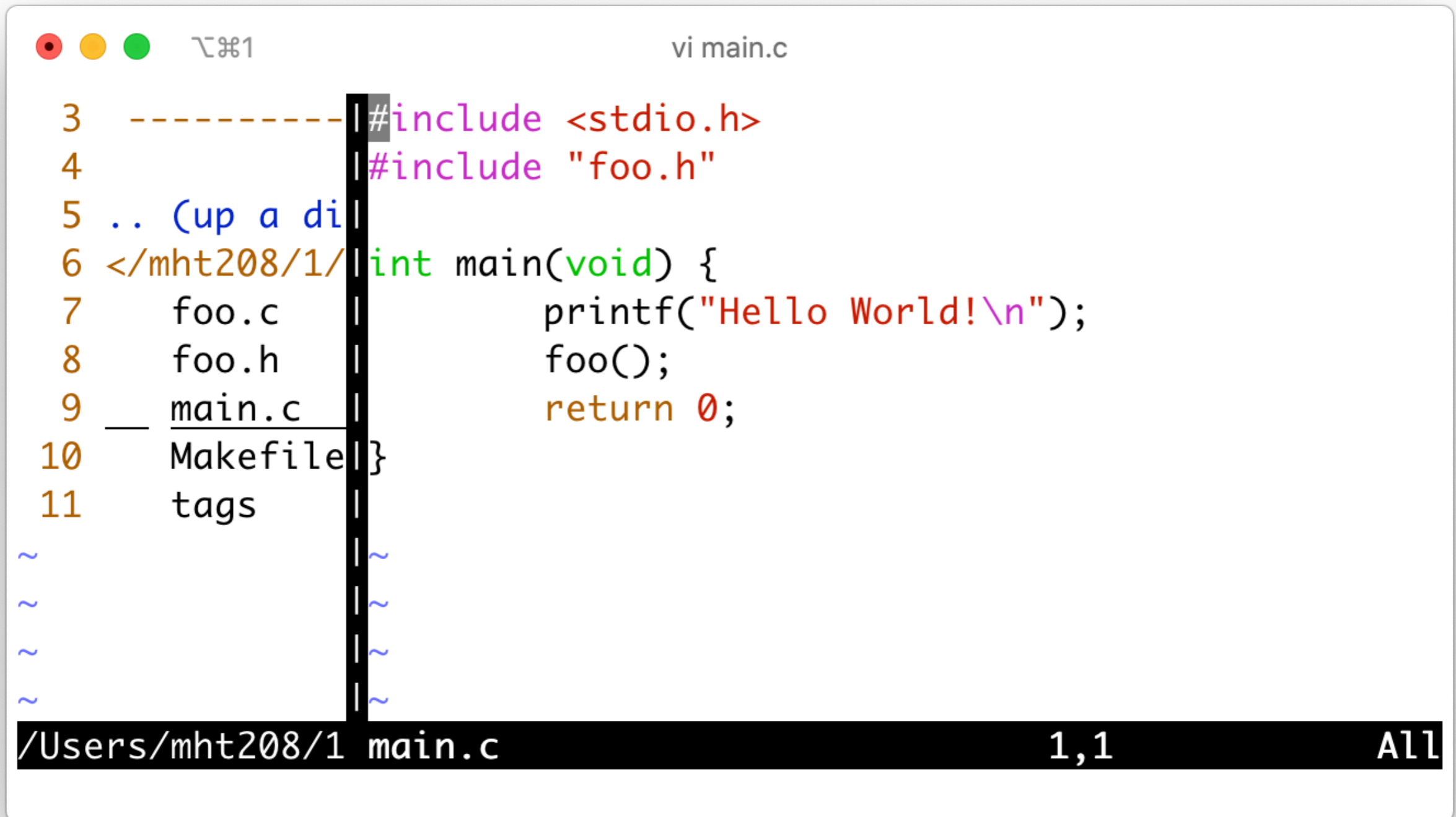
```
~
#include <stdio.h>
#include "foo.h"

int main(void) {
    printf("Hello World!\n");
    foo();
    return 0;
}

~
main.c 1,1 All
:q
```

Quit vi/vim

Vim with NERDTree



The image shows a Vim editor window with a split view. The left pane displays the NERDTree file explorer, listing files in a directory: `foo.c`, `foo.h`, `main.c`, `Makefile`, and `tags`. The right pane shows the contents of `main.c`, which includes `<stdio.h>` and `foo.h`, and contains a `main` function that prints "Hello World!\n". The status bar at the bottom indicates the current file is `/Users/mht208/1 main.c`, the cursor is at line 1, column 1, and the search scope is `All`.

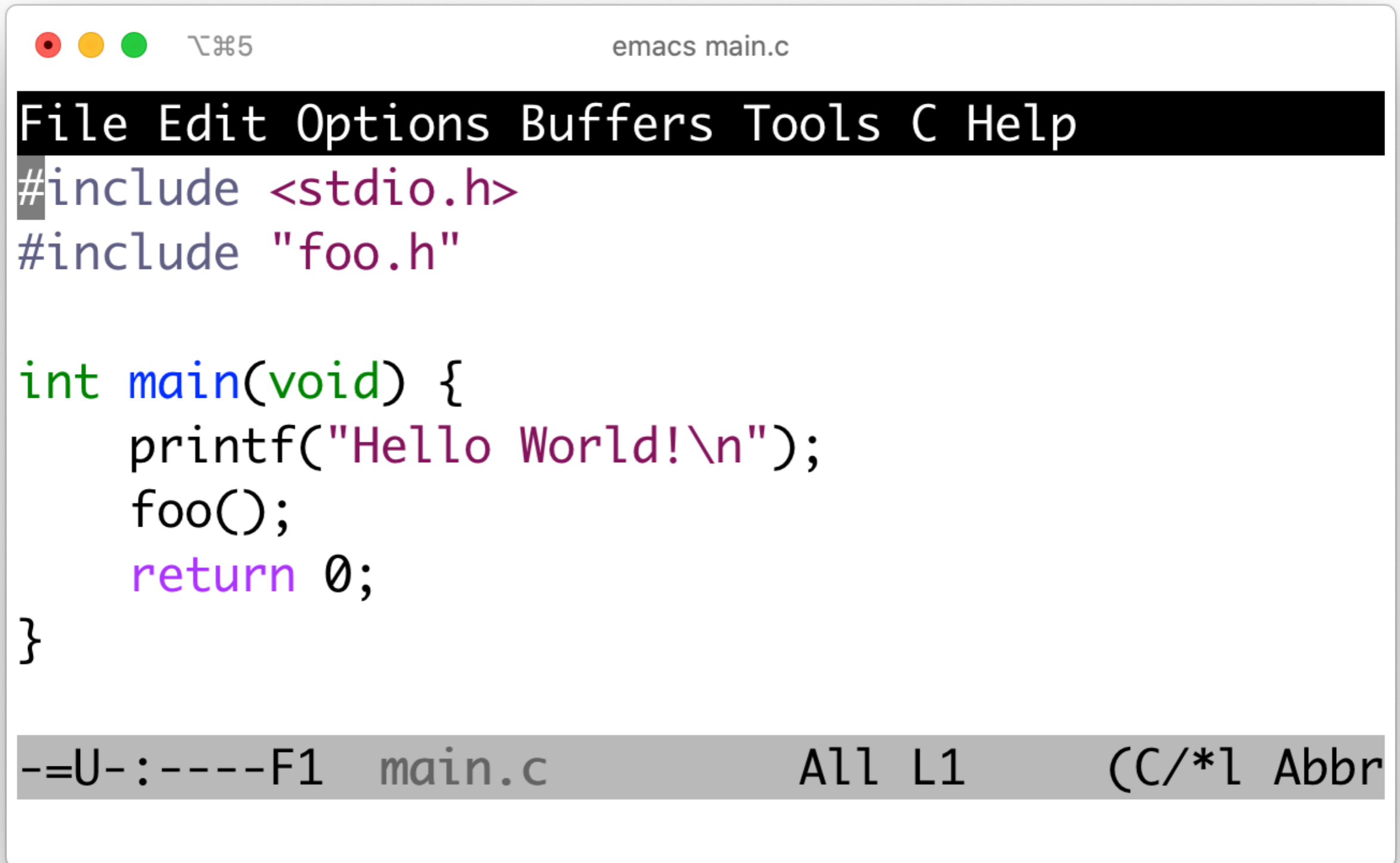
```
3  -----| #include <stdio.h>
4         | #include "foo.h"
5  .. (up a di|
6  </mht208/1/| int main(void) {
7  foo.c      |     printf("Hello World!\n");
8  foo.h      |     foo();
9  main.c     |     return 0;
10  Makefile  | }
11  tags      |
~          | ~
~          | ~
~          | ~
~          | ~
```

/Users/mht208/1 main.c 1,1 All

Emacs

- Another powerful editor on UNIX-like systems
- Use hotkeys
- Extensible with various packages
- Press C-C C-F to open a file
- Press C-X C-S to save a file
- Press C-X C-C to quit emacs
- Press M-x to enter commands

Emacs



The image shows a screenshot of the Emacs editor window. The title bar at the top contains three colored window control buttons (red, yellow, green) on the left, followed by the text "emacs main.c". Below the title bar is a menu bar with the items "File", "Edit", "Options", "Buffers", "Tools", "C", and "Help". The main editing area contains the following C code:

```
#include <stdio.h>
#include "foo.h"

int main(void) {
    printf("Hello World!\n");
    foo();
    return 0;
}
```

At the bottom of the window is a status bar with the text "--=U- : ----F1 main.c All L1 (C/*l Abbr".

Emacs in ecb-minor-mode

The screenshot shows the Emacs editor in ecb-minor-mode. The window title is "emacs ~/1/main.c". The menu bar includes "File Edit Options Buffers Tools C ECB Help". The main editing area is split into three windows:

- W-0 /Users/mht208/1**: A file explorer showing a tree view of the directory. The file `foo.c` is selected and highlighted in blue.
- W-3 foo.c**: A C source file with the following code:

```
#include <stdio.h>

void foo() {
    printf("foo\n");
}
```
- W-4 History**: A history window showing the current directory `/Users/mht208/1` and its contents: `foo.c`, `main`, and `main.c`. The file `foo.c` is selected and highlighted in blue.

The status bar at the bottom shows "W-1 /Users/mht208/1 | --U-:----F1 foo.c Top L3 | W-4 History".

Integrated Development Environment (IDE)

- A software application that provides comprehensive facilities to computer programmers for software development (Wikipedia)
 - source code editor
 - build automation tools
 - debugger
 - code completion
 - code refactoring
 - simulator
 - task / bug tracking
 - drag-and-drop graphic user interface creation

Using an IDE

- Advantages
 - Coding efficiency
 - Project management
- Disadvantages
 - Learning curve
 - Lag

Without/With IDE

obj.???

(what methods are available?)

Without/With IDE

obj.???

(what methods are available?)

```
public class Test {
```

```
    public static final void main(String[] args) {
```

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

```
        frame.
```

```
    }
```

- ★ setVisible(boolean b) : void - Window - 47%
- ★ getContentPane() : Container - JFrame - 28%
- ★ setTitle(String title) : void - Frame - 26%
- ★ dispose() : void - Window - 18%
- ★ setSize(int width, int height) : void - Window - 1%
- ★ setDefaultCloseOperation(int operation) : void - Window - 1%
- ★ setContentPane(Container contentPane) : void - JFrame - 1%
- action(Event evt, Object what) : boolean - Component
- add(Component comp) : Component - Container
- add(PopupMenu popup) : void - Component
- add(Component comp, int index) : Component - Container

Shows or hides this window depending on the value of parameter b.

If the method shows the window then the window is also made focused under the following conditions:

- The window meets the requirements outlined in the [isFocusableWindow](#) method.
- The window's `autoRequestFocus` property is of the true value.
- Native windowing system allows the window to get focused.

There is an exception for the second condition (the

Press '^Space' to show Template Proposals

Press 'Tab' from proposal table or click for focus

Problems ✖ @ Ja

Errors, 1,761 warnings

Intention

RESOURCE

Without/With IDE

obj.func(???)

(what arguments are needed?)

Without/With IDE

obj.func(???)

(what arguments are needed?)

frame.

- setLocale(Locale l) : void - Component
- setLocation(Point p) : void - Window
- **setLocation(int x, int y) : void - Window**
- setLocationByPlatform(boolean locationByPlatform) : void - Window
- setLocationRelativeTo(Component c) : void - Window
- setMaximizedBounds(Rectangle bounds) : void - Window
- setMaximumSize(Dimension maximumSize) : void - Window
- setMenuBar(MenuBar mb) : void - Frame
- setMinimumSize(Dimension minimumSize) : void - Window
- setModalExclusionType(ModalExclusionType exclusionType) : void - Window

Press '^Space' to show Template Proposals

therefore, invalidates the component hierarchy.

The method changes the geometry-related data. Therefore, the native windowing system may ignore such requests, or it may modify the requested data, so that the window object is placed and sized in a way that corresponds closely to the desktop settings.

Overrides: [setLocation\(...\)](#) in [Component](#)

Parameters:

x the x-coordinate of the new location's top-left corner in the parent's coordinate space
y the y-coordinate of the new location's top-left corner in the parent's coordinate space

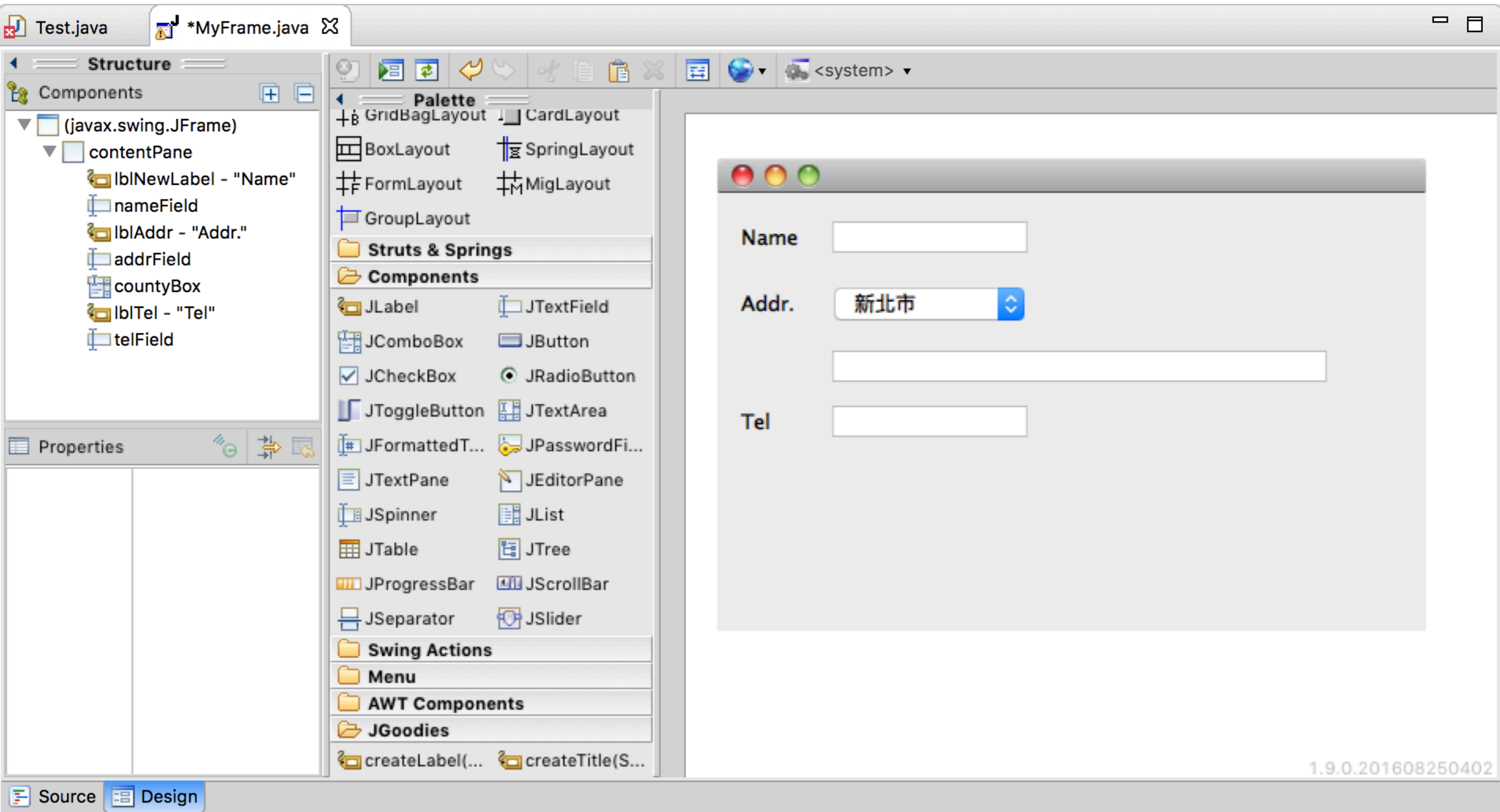


Without/With IDE

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

(build graphical user interface)

Without/With IDE



Eclipse

- <http://www.eclipse.org>
- Eclipse provides the rich client platform (RCP) for developing general-purpose applications
 - Core platform
 - Equinox (an implementation of OSGi core framework specification)
 - Standard Widget Toolkit (SWT)
 - JFace
 - Workbench

Eclipse-based Software

- Adobe ColdFusion Builder
- Coverity
- IBM Rational Software Architect
- IBM Notes
- Red Hat JBoss Developer Studio

Eclipse IDE

- Eclipse can serve as integrated development environment (IDE)
 - Java, C/C++, PHP, ...
- Various plugins are available (<http://marketplace.eclipse.org>)
 - WindowBuilder, EGit, Eclipse UML Generators, ...
- Free

Eclipse Features

About Eclipse IDE Features

About Eclipse IDE Features

Provider	Feature Name	Version	Feature Id
Eclipse.org	Eclipse Java Development Tools	3.18.200.v201912...	org.eclipse.jdt
Eclipse.org	Eclipse Platform	4.14.0.v20191210...	org.eclipse.platform
Eclipse.org	Eclipse RCP	4.14.0.v20191210...	org.eclipse.rcp
Eclipse.org	Eclipse User Storage	1.2.0.v20191120-...	org.eclipse.userstorage
Eclipse.org	Help System Base	2.3.0.v20191210-...	org.eclipse.help



Eclipse RCP

Version: 4.14.0.v20191210-0610
Build id: I20191210-0610

(c) Copyright Eclipse contributors and others 2000, 2019. All rights reserved.
Visit <http://www.eclipse.org/eclipse>

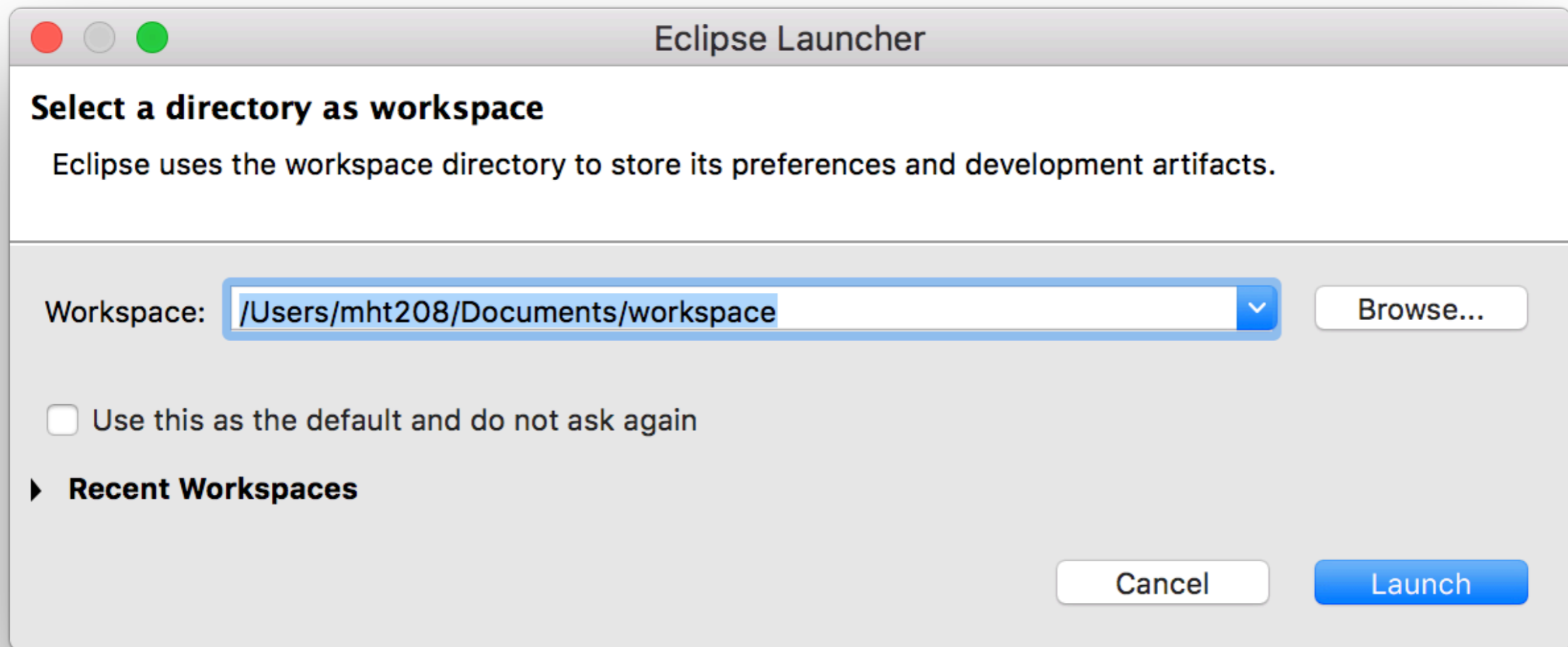
Eclipse History

Version Name	Date	Platform Version
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	28 June 2017	4.7
Photon	27 June 2018	4.8

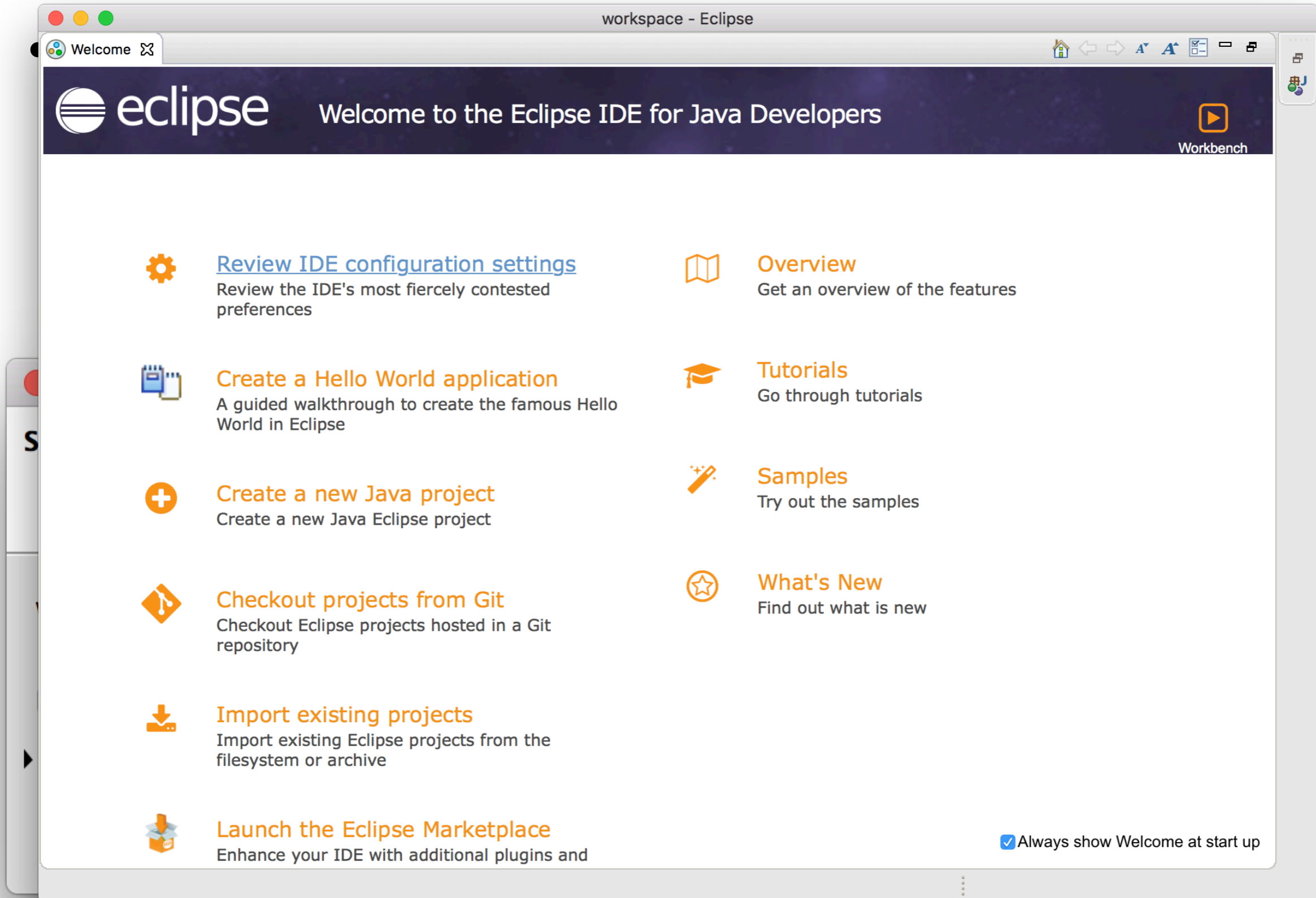
The latest version is Eclipse IDE 2019-12

First Start

- Workspace
 - Where your projects are stored
 - Multiple workspaces are allowed





First Start













The screenshot shows the Eclipse IDE interface. At the top, the title bar reads "workspace - Eclipse". Below it, a browser-like address bar contains "Welcome" and navigation icons. The main header features the Eclipse logo and the text "Welcome to the Eclipse IDE for Java Developers". A "Workbench" button with a play icon is in the top right. The main content area lists several tasks with icons: "Review IDE configuration settings" (gear icon), "Create a Hello World application" (notepad icon), "Create a new Java project" (plus icon), "Checkout projects from Git" (git icon), "Import existing projects" (download icon), and "Launch the Eclipse Marketplace" (package icon). On the right side, there are links for "Overview" (book icon), "Tutorials" (graduation cap icon), "Samples" (pencil icon), and "What's New" (star icon). At the bottom right, there is a checkbox labeled "Always show Welcome at start up" which is checked.

workspace - Eclipse

Welcome

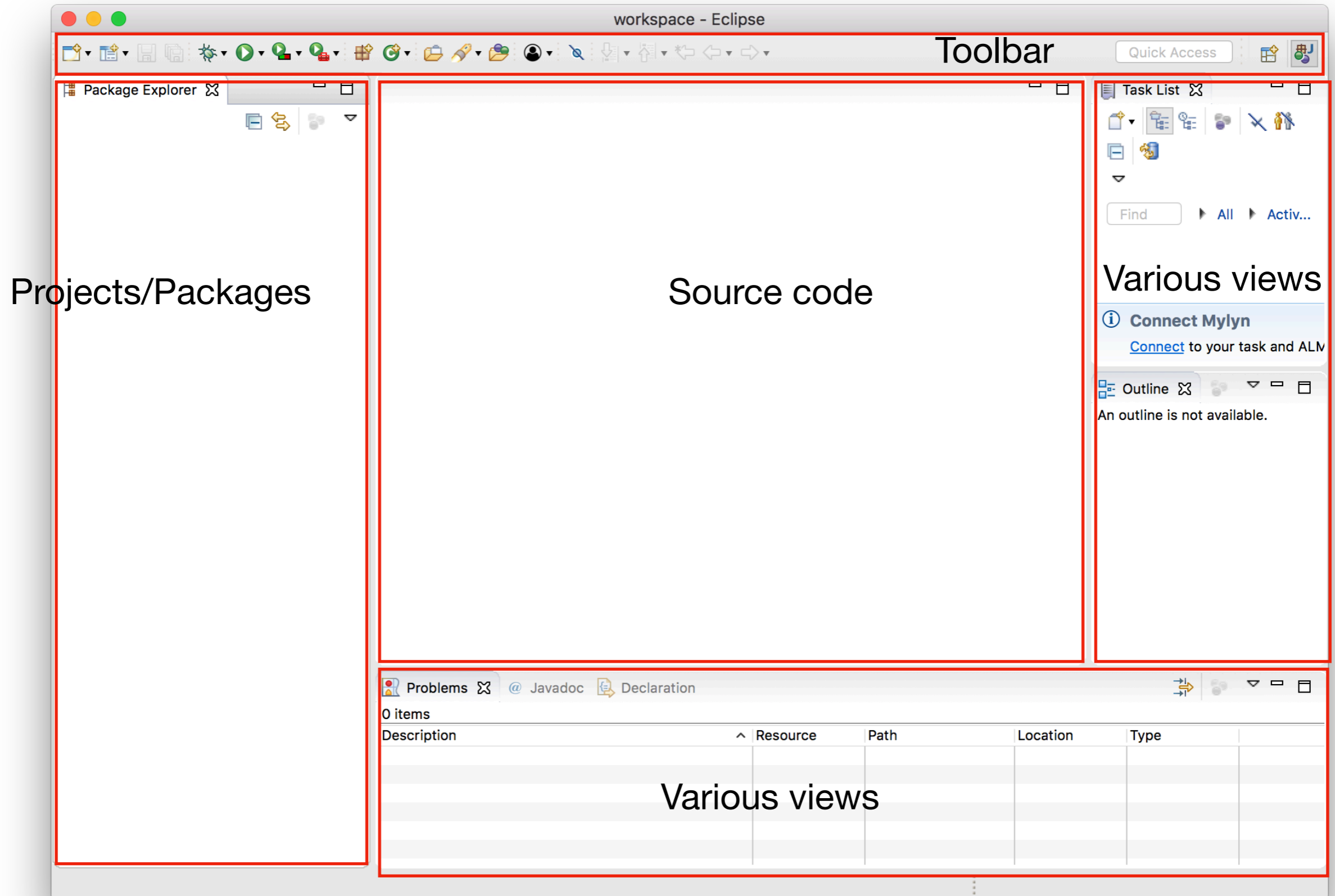
 Welcome to the Eclipse IDE for Java Developers  Workbench

-  **Review IDE configuration settings**
Review the IDE's most fiercely contested preferences
-  **Create a Hello World application**
A guided walkthrough to create the famous Hello World in Eclipse
-  **Create a new Java project**
Create a new Java Eclipse project
-  **Checkout projects from Git**
Checkout Eclipse projects hosted in a Git repository
-  **Import existing projects**
Import existing Eclipse projects from the filesystem or archive
-  **Launch the Eclipse Marketplace**
Enhance your IDE with additional plugins and

-  **Overview**
Get an overview of the features
-  **Tutorials**
Go through tutorials
-  **Samples**
Try out the samples
-  **What's New**
Find out what is new

Always show Welcome at start up

Perspective



Perspective Java

The screenshot displays the Eclipse IDE interface in the Perspective Java view. The main editor shows the source code for `AltAutomaton.java`. The code includes a package declaration, a `serialVersionUID`, a constructor, and two methods: `newInstance()` and `clone()`.

```
28 /**
29  *
30  */
31 private static final long serialVersionUID = 1716841552879597689L;
32
33 /**
34  * Constructor.
35  *
36  * @param atype
37  *     the alphabet type of this alternating automaton
38  * @param lpos
39  *     the label position of this alternating automaton
40  * @param style
41  *     the alternating style, either CNF or DNF, of this alternating
42  *     automaton
43  */
44 public AltAutomaton(AlphabetType atype, Position lpos, AltStyle style) {
45     super(atype, lpos, style);
46 }
47
48 @Override
49 protected Automaton newInstance() {
50     return new AltAutomaton(getAlphabetType(), getLabelPosition(),
51         getAltStyle());
52 }
53
54 @Override
55 public AltAutomaton clone() {
```

The Outline view on the right shows the class structure for `org.svvl.goal.core.aut.alt`, including `AltAutomaton` and its methods: `serialVersionUID : long`, `AltAutomaton(AlphabetType, Position, AltStyle)`, `clone() : AltAutomaton`, `newInstance() : Automaton`, and `reorder() : void`.

The Problems view at the bottom shows 11 errors, 1,776 warnings, and 5 others. The table below summarizes the error counts:

Description	Resource	Path	Location	Type
▶ 11 errors (11 items)				
▶ 100 of 1776 warnings (100 of 1776 items)				
▶ 5 infos (5 items)				

At the bottom of the IDE, the status bar shows "Writable", "Smart Insert", and "1 : 1".

Perspective Java Browsing

The screenshot shows the Eclipse IDE in the Perspective Java Browsing view. The main editor displays the source code of `AltAutomaton.java`. The right-hand side shows the Types and Members views.

```
36     * @param atype
37     *         the alphabet type of this alternating automaton
38     * @param lpos
39     *         the label position of this alternating automaton
40     * @param style
41     *         the alternating style, either CNF or DNF, of this alternating
42     *         automaton
43     */
44     public AltAutomaton(AlphabetType atype, Position lpos, AltStyle style) {
45         super(atype, lpos, style);
46     }
47
48     @Override
49     protected Automaton newInstance() {
50         return new AltAutomaton(getAlphabetType(), getLabelPosition(),
51                                 getAltStyle());
52     }
53
54     @Override
55     public AltAutomaton clone() {
56         return (AltAutomaton) super.clone();
57     }
58
59     @Override
60     public void reorder() {
61         super.reorder();
62
63         int gcid = gsid;
64         int dec = 0;
```

Types View:

- AltAutomaton
- AltConnector
- AltState
- AltStyle
- AltStyleConversion
- AltTransition
- CNFNABWCreator
- CNFNACWCreator
- CNFNAGRWCreator

Members View:

- serialVersionUID : long
- AltAutomaton(AlphabetType, Position, AltStyle)
- newInstance() : Automaton
- clone() : AltAutomaton
- reorder() : void

Bottom status bar: Writable | Smart Insert | 1 : 1

Perspective

Debug

The screenshot shows the Eclipse IDE in the Debug Perspective. The main window title is "workspace - MyTest/src/Test.java - Eclipse". The interface is divided into several panes:

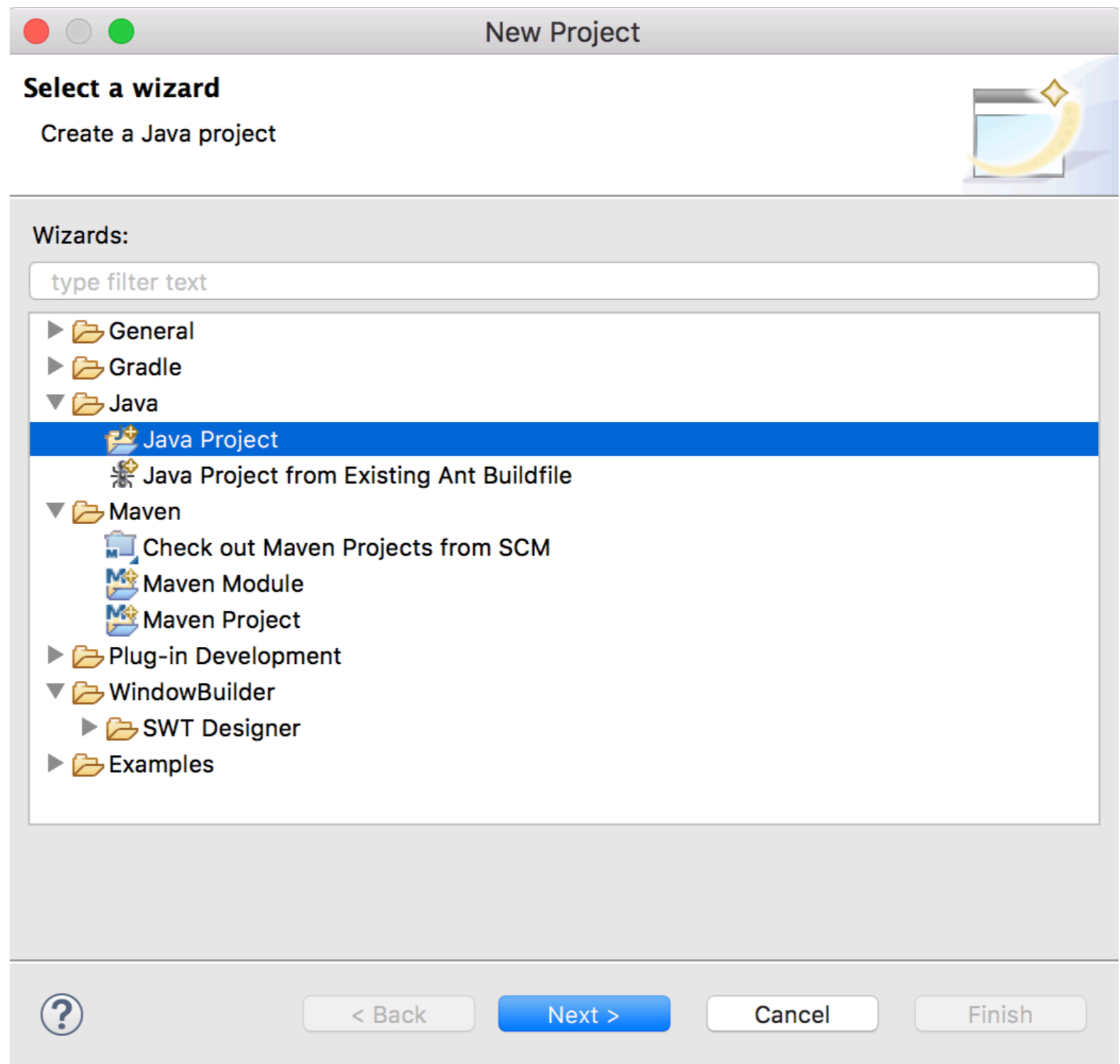
- Debug Console:** Shows the execution stack. The current thread is "Thread [main] (Suspended (breakpoint at line 25 in Test))". The current method is "Test.sort(int[]) line: 25".
- Variables View:** Displays the state of variables. The variable "xs" is an array of integers with the following values:

Name	Value
xs	(id=15)
xs [0]	5
xs [1]	7
xs [2]	1
xs [3]	3
xs [4]	9
- Code Editor:** Shows the source code of "Test.java". The current line of execution is line 25, which is highlighted in green. The code is:

```
20 public class Test {
21
22     public static void sort(int[] xs) {
23         for (int i = 0; i < xs.length - 1; i++) {
24             for (int j = i + 1; j < xs.length; j++) {
25                 if (xs[j] < xs[i]) {
26                     int t = xs[i];
27                     xs[i] = xs[j];
28                     xs[j] = t;
29                 }
30             }
31         }
32     }
33 }
```
- Outline View:** Shows the class structure. The class "Test" is expanded, showing the methods "main(String[]) : void" and "sort(int[]) : void".
- Console:** Shows the output of the application. The text is: "Test [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_25.jdk/Contents/Home/bin/java (Sep 19, 2017, 12:37:47 PM)".

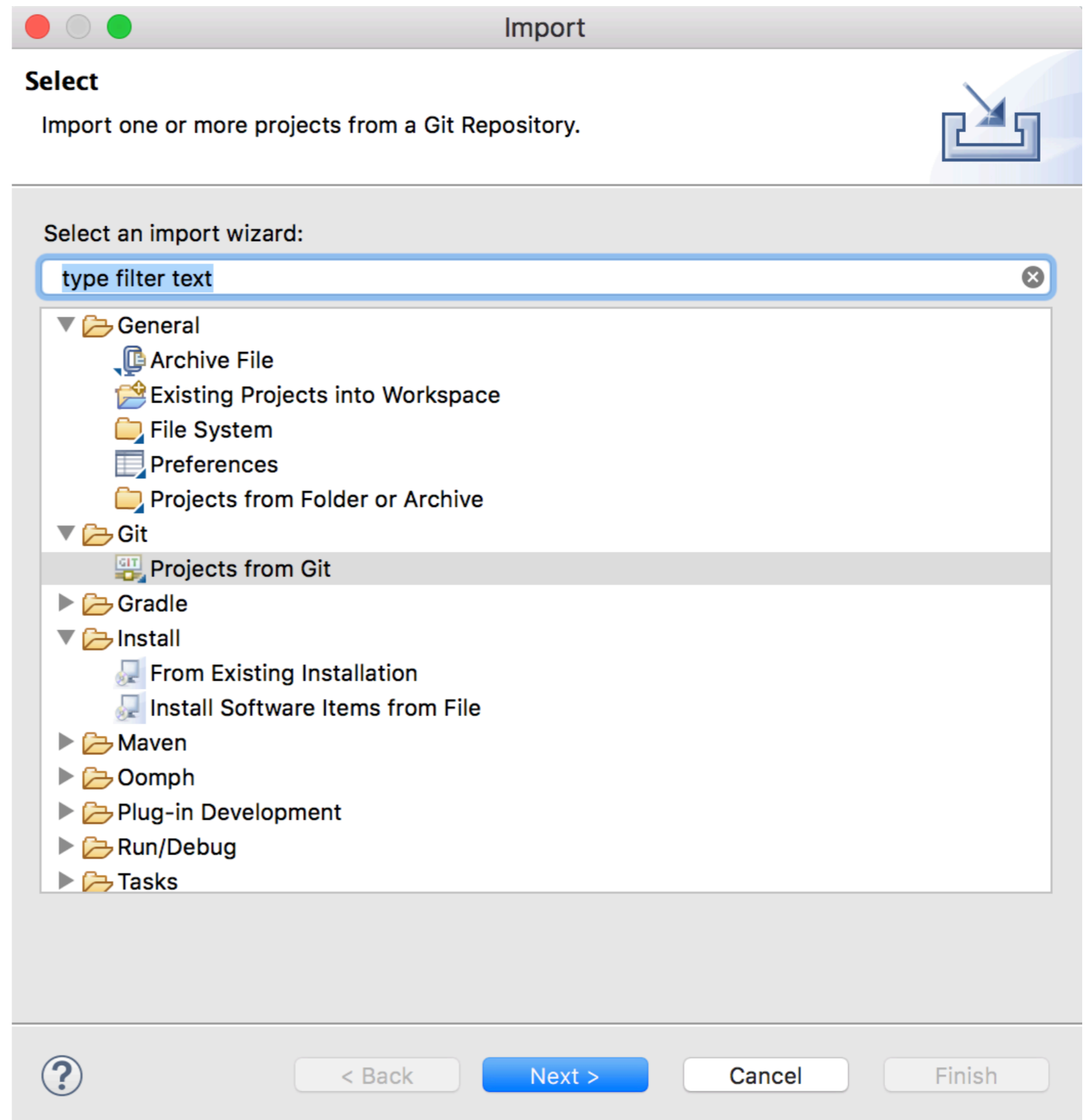
Creating New Projects

File / New / Project...



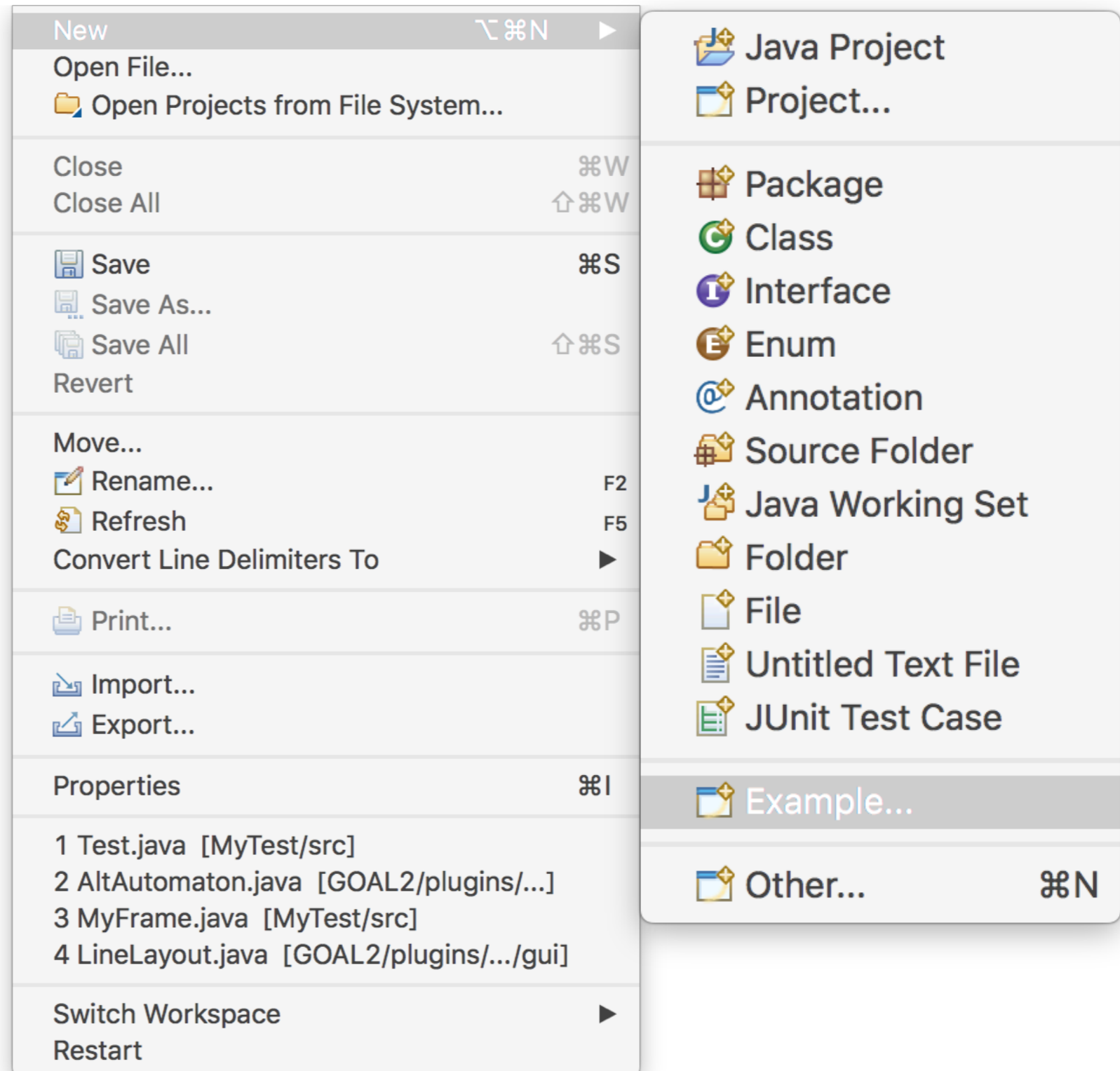
Importing Existing Projects

File / Import...



New Source Files

File / New (⌘N)



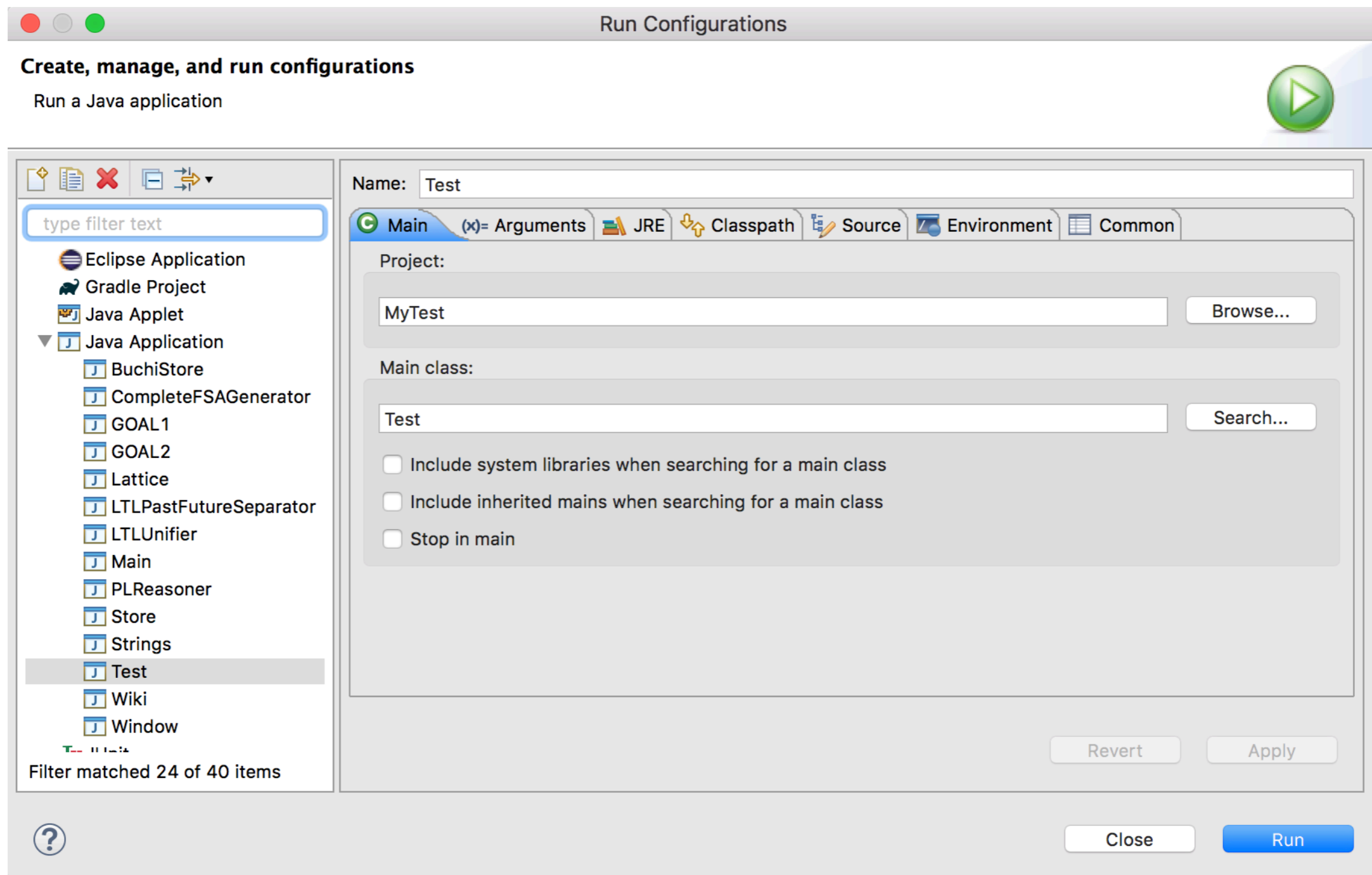
Build Projects

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

Run Projects

Run / Run Configurations...

Run / Run (⇧ ⌘ F11)



Project Management

Right click on a project / Properties

Uniform code style and policy

Properties for MyTest

type filter text

- ▼ Resource
 - Linked Resources
 - Resource Filters
- Builders
- Coverage
- Java Build Path
- ▼ **Java Code Style**
 - Clean Up
 - Code Templates
 - Formatter
 - Organize Imports
- ▼ Java Compiler
 - ▶ Annotation Processing
 - Building
 - Errors/Warnings
 - Javadoc
 - Task Tags
- ▶ Java Editor
 - Javadoc Location
 - Project References

Java Code Style ← ▾ → ▾ ▾

Enable project specific settings [Configure Workspace Settings...](#)

Conventions for variable names:

Variable type	Prefix list	Suffix list
<input checked="" type="checkbox"/> Fields		
<input checked="" type="checkbox"/> ^S Static Fields		
<input checked="" type="checkbox"/> ^S _F Static Final Fields		
<input type="checkbox"/> _L Parameters		
<input type="checkbox"/> _L Local Variables		

Qualify all generated field accesses with 'this.'

Use 'is' prefix for getters that return boolean

Add '@Override' annotation for new overriding methods
([configure compiler option](#) for implementations of interface methods)

Exception variable name in catch blocks:

Edit...

Project Management

Right click on a project / Properties

Uniform code style and policy

The image shows a screenshot of an IDE's 'Properties' dialog for a project, specifically the 'Javadoc' tab. The left sidebar shows a tree view with 'Java Code Style' expanded and 'Javadoc' selected. The main dialog has several sections:

- Enable project specific settings [Configure Workspace Settings...](#)
- Note:** This also affects features like search, refactoring, content assist and missing/unused imports.
- Process Javadoc comments
- Severity levels for problems in Javadoc comments:
- Malformed Javadoc comments:
- Only consider members as visible as:
- Validate tag arguments (@param, @throws, @exception, @see, @link)
- Report non visible references
- Report deprecated references
- Missing tag descriptions:
- Missing Javadoc tags:
- Only consider members as visible as:
- Ignore in overriding and implementing methods
- Ignore method type parameters

API Documents

How would you search for available APIs?

In IDE

The screenshot displays an IDE window with the Automaton class code and its API outline. The code is as follows:

```
1186
1188+ * Inserts a transition to the transition maps.
1193+ private void addToTransitionMaps(Transition t) {}
1220
1222+ * Removes a transition from transition maps.
1227+ private void removeFromTransitionMaps(Transition t) {}
1239
1241+ * Creates a transition to the automaton. The returned transition is
1257+ public Transition createTransition(State from, State to, String symbol) {}
1286
1288+ * Creates a new instance of a transition.
1299 protected abstract Transition newTransition(int id, State from, State to);
1300
1302+ * Creates a new instance of a transition and makes the transition inherit
1317+ protected Transition newTransition(int id, State from, State to,
1323
1325+ * Inserts a transition created outside to this automaton. The transition
1334+ public boolean addTransition(Transition t) {}
1382
1384+ * Returns the transition set of this automaton.
1388+ public Transition[] getTransitions() {}
1393
1395+ * Returns the number of transitions.
1399+ public int getTransitionSize() {}
1404
1406+ * Returns a transition by its ID.
1416+ public Transition getTransitionByID(int id) throws NoSuchElementException {}
1424
```

The API outline on the right side of the IDE shows the following fields and methods:

- FORMULA : String
- serialVersionUID : long
- acc : Acc<?>
- aps : Set<String>
- atype : AlphabetType
- complete_transition : boolean
- from_map : Map<State, Trans
- from_to_map : BinaryMap<Sta
- gsid : int
- gtid : int
- inits : StateSet
- invisible_inits : Stack<StateSe
- invisible_states : Stack<StateS
- invisible_trans : Stack<Transit
- listeners : Set<AutomatonList
- lpos : Position
- states : StateMap
- to_map : Map<State, Transiti
- trans : TransitionSet
- validate_transition_label : boo
- Automaton(AlphabetType, Po
- addAutomatonListener(Autom
- addInitialState(State) : void
- addState(State) : boolean

API Documents

How would you search for available APIs?

In IDE

In browser

The image shows a screenshot of an IDE with a code editor on the left and a 'Method Summary' window on the right. The code editor displays the source code for Automaton.java, showing various methods like `addToTransitionMaps`, `removeFromTransitionMaps`, `createTransition`, `newTransition`, `addTransition`, `getTransitions`, `getTransitionSize`, and `getTransitionByID`. The 'Method Summary' window provides a structured overview of these methods, categorized into 'All Methods', 'Instance Methods', 'Abstract Methods', and 'Concrete Methods'. It lists the modifier and type for each method along with a brief description.

Modifier and Type	Method and Description
void	addAutomatonListener (AutomatonListener listener) Adds an automaton listener to this automaton.
void	addInitialState (State state) Sets a state in this automaton as an initial state.
boolean	addState (State s) Inserts a state created outside to this automaton.
boolean	addState (State s, boolean force) Inserts a state created outside to this automaton.
boolean	addTransition (Transition t) Inserts a transition created outside to this automaton.
Automaton	
void	clone () clone (Automaton aut) Makes this automaton as a clone of another automaton.
void	completeTransitions () Makes transitions complete if they are simplified.
void	completeTransitions (State f, State t) Makes the transitions between two states complete.
boolean	containsEquivalentTransition (Transition t) Returns true if a specified transition is equivalent to an existing transition in this automaton.
boolean	containsInitialState (State s) Checks if a state is an initial state.

Javadoc

⌘⌘J

/** ↩

```
/**
 * Sorts an integer array ascendantly.
 *
 * @param xs
 *         an integer to be sorted
 */
public static void sort(int[] xs) {
    for (int i = 0; i < xs.length - 1; i++) {
        for (int j = i + 1; j < xs.length; j++) {
            if (xs[j] < xs[i]) {
                int t = xs[i];
                xs[i] = xs[j];
                xs[j] = t;
            }
        }
    }
}
```

File / Export... / Java / Javadoc

Javadoc

⇧ ⌘ J

/** ←

```
/**  
 * Sorts an integer array ascendantly.  
 *  
 * @param xs
```

The screenshot shows an IDE window with a class hierarchy on the left and a Javadoc tooltip on the right. The class hierarchy lists methods like clone(), equals(), finalize(), getClass(), hashCode(), notify(), notifyAll(), sort(int[] xs), toString(), wait(), and wait(long timeout). The sort method is highlighted. The tooltip displays the Javadoc for the sort method: "Sorts an integer array ascendantly. Parameters: xs an integer to be sorted".

clone() : Object - Object	Sorts an integer array ascendantly. Parameters: xs an integer to be sorted
equals(Object obj) : boolean - Object	
finalize() : void - Object	
getClass() : Class<?> - Object	
hashCode() : int - Object	
notify() : void - Object	
notifyAll() : void - Object	
sort(int[] xs) : void - Test	
toString() : String - Object	
wait() : void - Object	
wait(long timeout) : void - Object	

Press '⌘0' to show Template Proposals

Press 'Tab' from proposal table or click for focus

File / Export... / Java / Javadoc

Javadoc Tags

- `@author <NAME>`
- `@version <VERSION>`
- `@param <VARIABLE> <DESCRIPTION>`
- `@return <DESCRIPTION>`
- `@deprecated <DESCRIPTION>`
- `@since <VERSION>`
- `@throws <EXCEPTION> <DESCRIPTION>`
- `@exception <EXCEPTION>
<DESCRIPTION>`
- `@see <CLASSPATH>`
- ...

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

Code Generation

Getters/Setters:

Source / Generate Getters and Setters...

Override/Implement:

Source / Overwrite/Implement Methods...

...

Code Generation

Getters/Setters:

```
public class Point {  
    private int x;  
  
    private int y;  
  
    public Point() {  
    }  
  
}
```

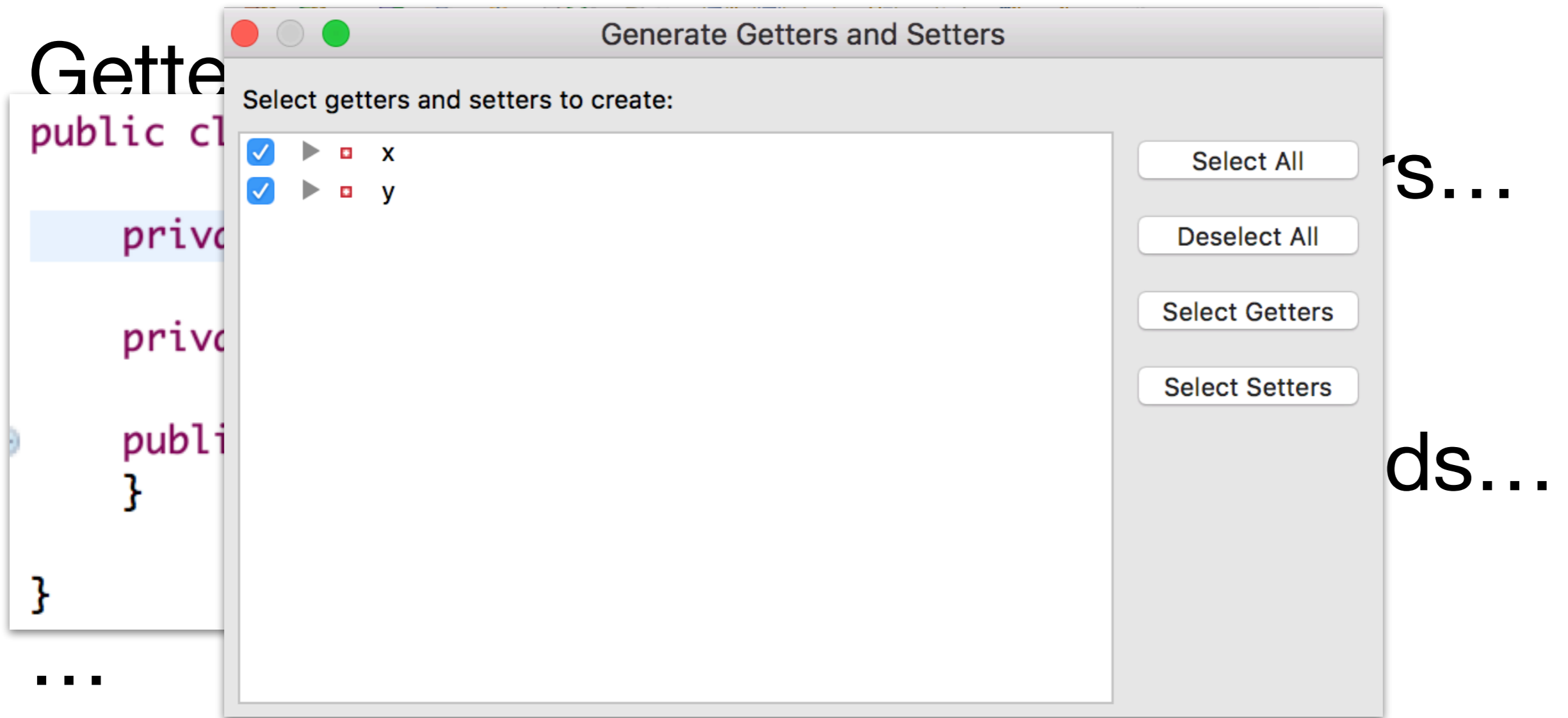
Generate Getters and Setters...

Implementation:

Write/Implement Methods...

...

Code Generation



Code Generation

Getters

```
public class
```

```
private
```

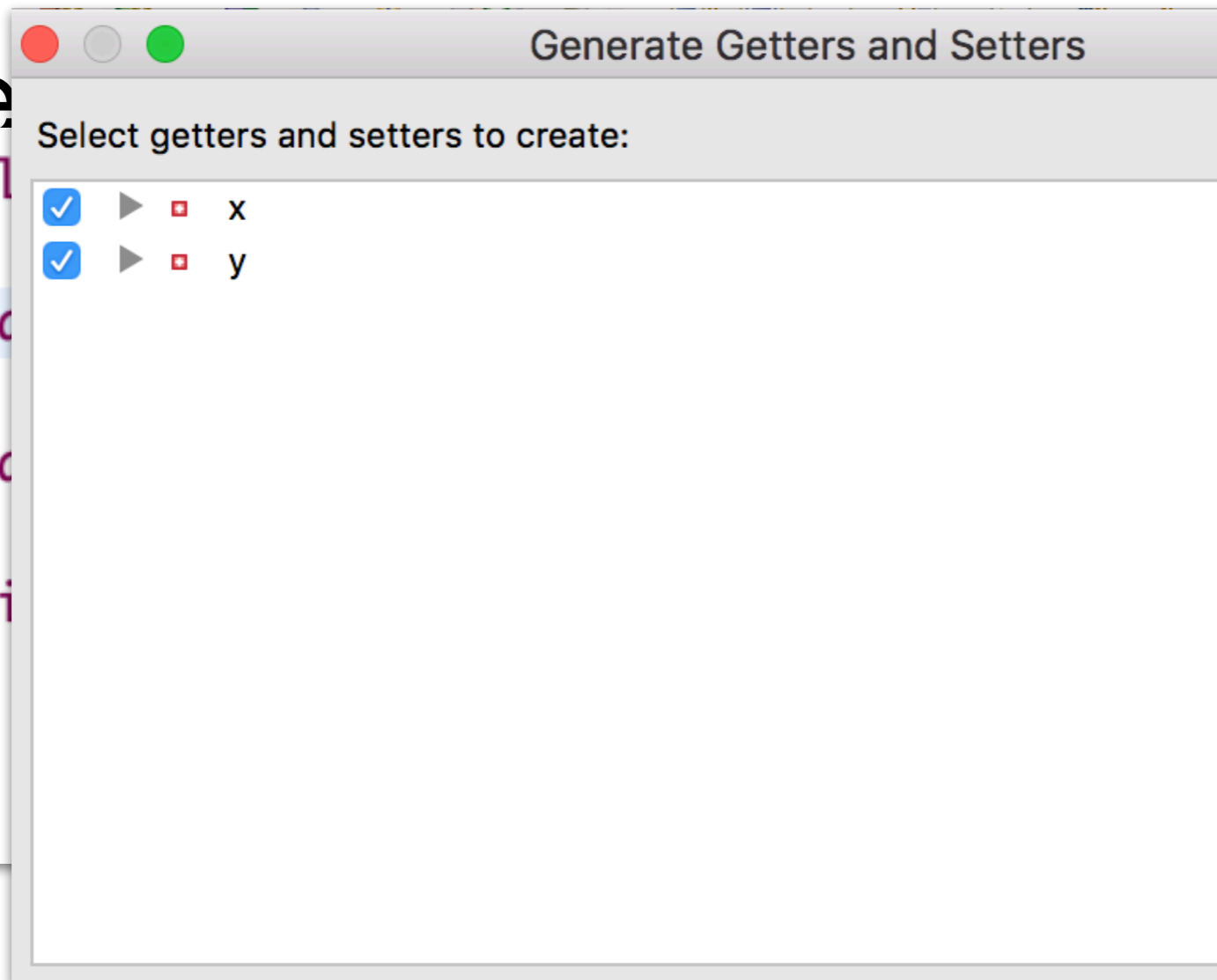
```
private
```

```
public
```

```
}
```

```
}
```

...



```
private int x;

private int y;

/**
 * @return the x
 */
public int getX() {
    return x;
}

/**
 * @param x the x to set
 */
public void setX(int x) {
    this.x = x;
}

/**
 * @return the y
 */
public int getY() {
    return y;
}

/**
 * @param y the y to set
 */
```

Navigation

- Navigate / Open Declaration (F3)
- Navigate / Open Type Hierarchy (F4)
- Navigate / Open Call Hierarchy (^⌘H)

Navigation

- Navigate / Open Declaration (F3)

```
Point p = new Point();  
p.setX(10);
```

- Navigate / Open Type Hierarchy (F4)
- Navigate / Open Call Hierarchy (^⇧H)

Navigation

- Navigate / Open Declaration

```
Point p = new Point();  
p.setX(10);
```

- Navigate / Open Type Hierarchy

- Navigate / Open Call Hierarchy

```
/**  
 * @return the x  
 */  
public int getX() {  
    return x;  
}  
  
/**  
 * @param x the x to set  
 */  
public void setX(int x) {  
    this.x = x;  
}  
  
/**  
 * @return the y  
 */  
public int getY() {  
    return y;  
}
```

Search

- Search / References / Workspace (⇧ ⌘ G)

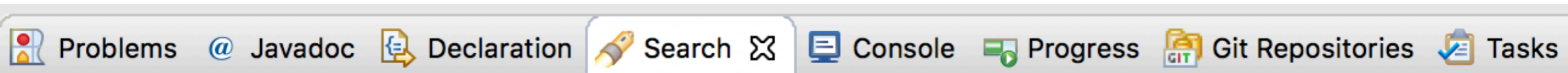
Search

- Search / References / Workspace (⇧ ⌘ G)

```
public class RankConstruction extends AbstractComplementConstruction<FSA, FSA> {
```


Search

- Search / References / Workspace (⇧ ⌘ G)



'org.svtrl.goal.core.comp.rank.RankConstruction' - 48 references in workspace (no JRE) (4 matches filtered from view)

- ▼ > org.svtrl.goal.cmd - plugins/org.svtrl.goal.cmd/source - GOAL2
 - ▼ RankComplementExtension
 - `getOptions(Context, List<Expression>)` (9 matches)
- ▼ org.svtrl.goal.core.comp.rank - plugins/org.svtrl.goal.core/source - GOAL2
 - ▼ RankConstruction
 - `RANK_STATE`
- ▼ org.svtrl.goal.gui.action - plugins/org.svtrl.goal.gui/source - GOAL2
 - ▼ RankComplementAction (1 match)
 - ◆ `getConstruction(FSA, Properties)` (2 matches)
 - ◆ `getConstructionClass()` (2 matches)
 - ▶ StepByStepRankComplementAction
- ▶ org.svtrl.goal.gui.pref - plugins/org.svtrl.goal.gui/source - GOAL2

Source

- Source / Format (⇧ ⌘ F)
- Source / Organize Imports (⇧ ⌘ O)
- Source / Toggle Comment (⌘ /)

Source

```
public void sort(int[] xs) {  
    for (int i=0; i <xs.length-1;i++) {  
        for (int j=i+1;j<xs.length; j++) {  
            if (xs[j] < xs[i]) {  
                int t = xs[i];  
                xs[i] = xs[j];  
                xs[j] = t;  
            }  
        }  
    }  
}
```

⌘O)

- Source / Toggle Comment (⌘/)

Source

```
public void sort(int[] xs) {  
    for (int i=0; i <xs.length-1;i++) {  
        for (int j=i+1;j<xs.length; j++) {  
            if (xs[j] < xs[i]) {  
                int t = xs[i];  
                xs[i] = xs[j];  
                xs[j] = t;  
            }  
        }  
    }  
}
```

```
public void sort(int[] xs) {  
    for (int i = 0; i < xs.length - 1; i++) {  
        for (int j = i + 1; j < xs.length; j++) {  
            if (xs[j] < xs[i]) {  
                int t = xs[i];  
                xs[i] = xs[j];  
                xs[j] = t;  
            }  
        }  
    }  
}
```

- Source / Toggle

Refactor

- Refactor / Rename... ($\lrcorner \text{⌘} R$)
- Refactor / Move... ($\lrcorner \text{⌘} V$)

Refactor

```
public void sort(int[] xs) {  
    for (int i = 0; i < xs.length - 1; i++) {  
        for (int j = 0; j < xs.length - 1 - i; j++) {  
            if (xs[j] > xs[j + 1]) {  
                int t = xs[j];  
                xs[j] = xs[j + 1];  
                xs[j + 1] = t;  
            }  
        }  
    }  
}
```

```
public static final void main(String[] args) {  
    Test t = new Test();  
    int[] xs = { 5, 7, 1, 6, 3, 9, 4, 2, 8 };  
    t.sort(xs);  
}
```

Original

Refactor

```
public void sort(int[] xs) {
```

```
public void bubbleSort(int[] xs) {  
    for (int i = 0; i < xs.length - 1; i++) {  
        for (int j = 0; j < xs.length - 1 - i; j++) {  
            if (xs[j] > xs[j + 1]) {  
                int t = xs[j];  
                xs[j] = xs[j + 1];  
                xs[j + 1] = t;  
            }  
        }  
    }  
}
```

```
public static final void main(String[] args) {  
    Test t = new Test();  
    int[] xs = { 5, 7, 1, 6, 3, 9, 4, 2, 8 };  
    t.sort(xs);  
}
```

Rename

Refactor

```
public void sort(int[] xs) {  
public void bubbleSort(int[] xs) {  
public void sort(int[] xs) {  
    for (int i = 0; i < xs.length - 1; i++) {  
        for (int j = 1 - i; j++ < xs.length - i - 1; j++) {  
            if (xs[j] > xs[j + 1]) {  
                int t = xs[j];  
                xs[j] = xs[j + 1];  
                xs[j + 1] = t;  
            }  
        }  
    }  
}  
}  
public static final void main(String[] args) {  
    Test t = new Test();  
    int[] xs = { 5, 7, 1, 6, 3, 9, 4, 2, 8 };  
    t.sort(xs);  
}
```

Refactor / Rename...

Refactor

```
public void sort(int[] xs) {  
    public void bubbleSort(int[] xs) {  
        public void sort(int[] xs) {  
            public void bubbleSort(int[] xs) {  
                for (int i = 0; i < xs.length - 1; i++) {  
                    for (int j = i + 1; j < xs.length - 1 - i; j++) {  
                        Press ↵ to refactor. Options... ▼  
                        if (xs[j] > xs[j + 1]) {  
                            int t = xs[j];  
                            xs[j] = xs[j + 1];  
                            xs[j + 1] = t;  
                        }  
                    }  
                }  
            }  
        }  
    }  
}  
public class Test {  
    public Test() {}  
    public void sort(int[] xs) {}  
    public void bubbleSort(int[] xs) {}  
}  
public static final void main(String[] args) {  
    Test t = new Test();  
    int[] xs = { 5, 7, 1, 6, 3, 9, 4, 2, 8 };  
    t.bubbleSort(xs);  
}
```

Refactor / Rename...

Others

- Quick Fix (⌘ 1)
- Shortcuts reference (⇧ ⌘ L)

Others

 44

```
System.out.println(xs);
```

- Quick Fix (⌘ 1)
- Shortcuts reference (⇧ ⌘ L)

Others

44 `System.out.println(xs);`

44 `System.out.println(xs);`
45 `System.out.println(xs);`
46 `}`
47
48 `}`
49

- Change to 'print(..)'
- Change to 'println(..)'
- Add cast to 'System.out'
- Rename in file (⌘2 R)

...
t.bubbleSort(xs);
System.out.print(xs);
System.out.println(Arrays.toString(xs));
...

Problems @ Javadoc Dec

<terminated> Test [Java Application]
[1, 2, 3, 4, 5, 6, 7, 8, 9]

Press 'Tab' from proposal table or click for focus

Others

```

44
45
46 }
47
48 }
49

```

Problems @ Javadoc De

```

<terminated> Test [Java Application]
[1, 2, 3, 4, 5, 6, 7, 8, 9]

```

Activate Editor	⌘F12
Activate Task	⌘F9
Add Artifact to Target Platform	⇧⌘A
Add Block Comment	^⌘/
Add Import	⇧⌘M
Add Javadoc Comment	⇧⌘J
All Instances	⇧⌘N
Backward History	⇧⌘←
Build All	⌘B
Change Method Signature	⇧⌘C
Close	⌘W
Close All	⇧⌘W
Collapse	⌘Numpad_Subtract
Collapse All	⇧⌘Numpad_Divide
Commit...	⇧⌘3
Content Assist	⌘0
Context Information	⇧⌘Space
Copy	⌘C
Copy Lines	⇧⌘↓
Correct Indentation	⌘I
Coverage Eclipse Application	⇧⌘E E
Coverage JUnit Plug-in Test	⇧⌘E P
Coverage JUnit Test	⇧⌘E T

Press "⇧⌘L" to open the preference page.

```

rays.toString(xs));

```

Tab' from proposal table or click for focus

Other Languages

- Eclipse CDT for C/C++
- Eclipse PDT for PHP
- Eclipse JSDT for Javascript
- PyDev for Python
- Scala IDE for Scala

Other Features

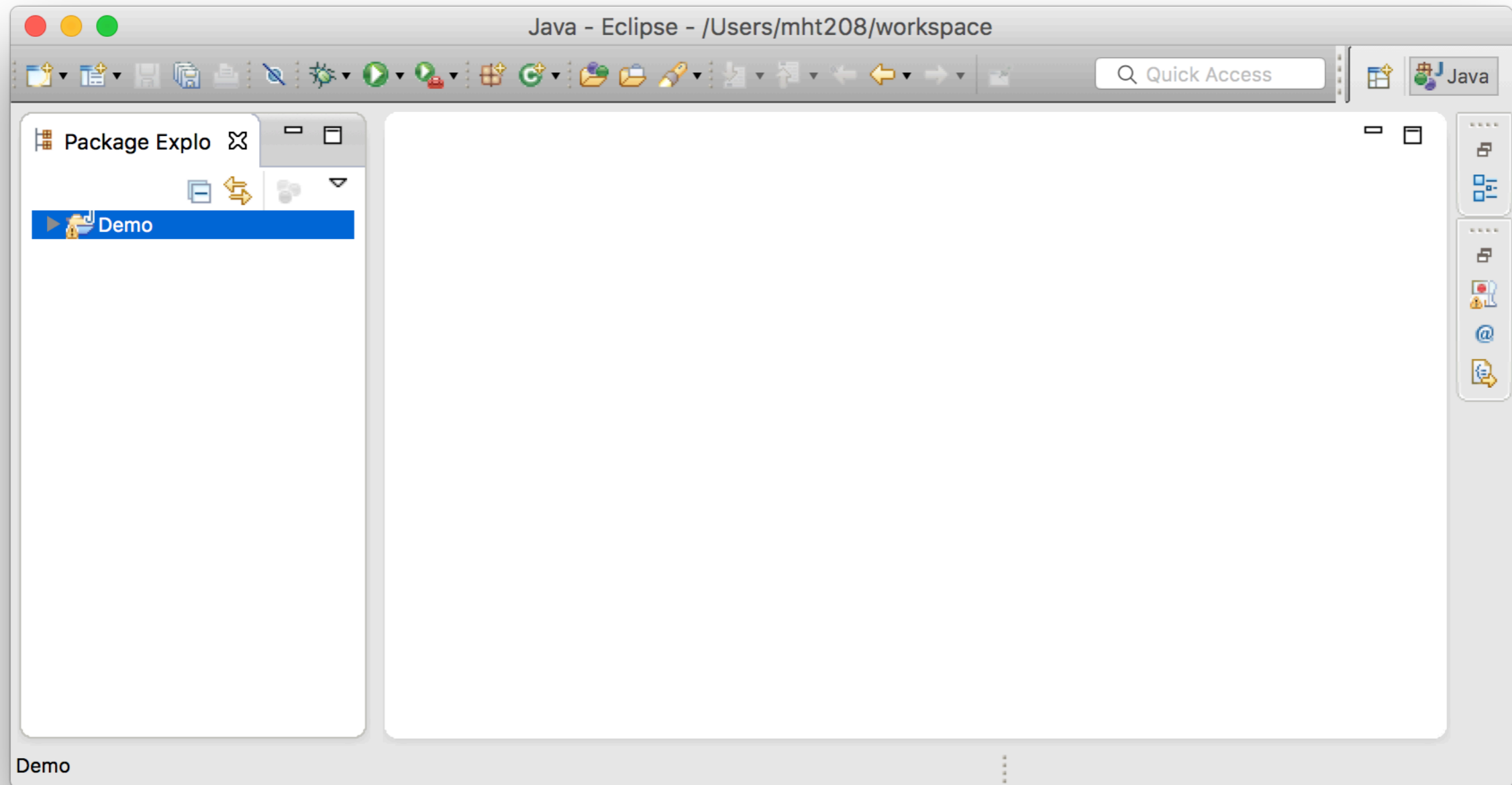
- Debugging
- UML diagrams and code generation
 - UML Designer, UML to Java code generator
- Task management
 - Mylyn
- Issue tracking
 - Bugzilla, JIRA, Redmine, ...

Other Features

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

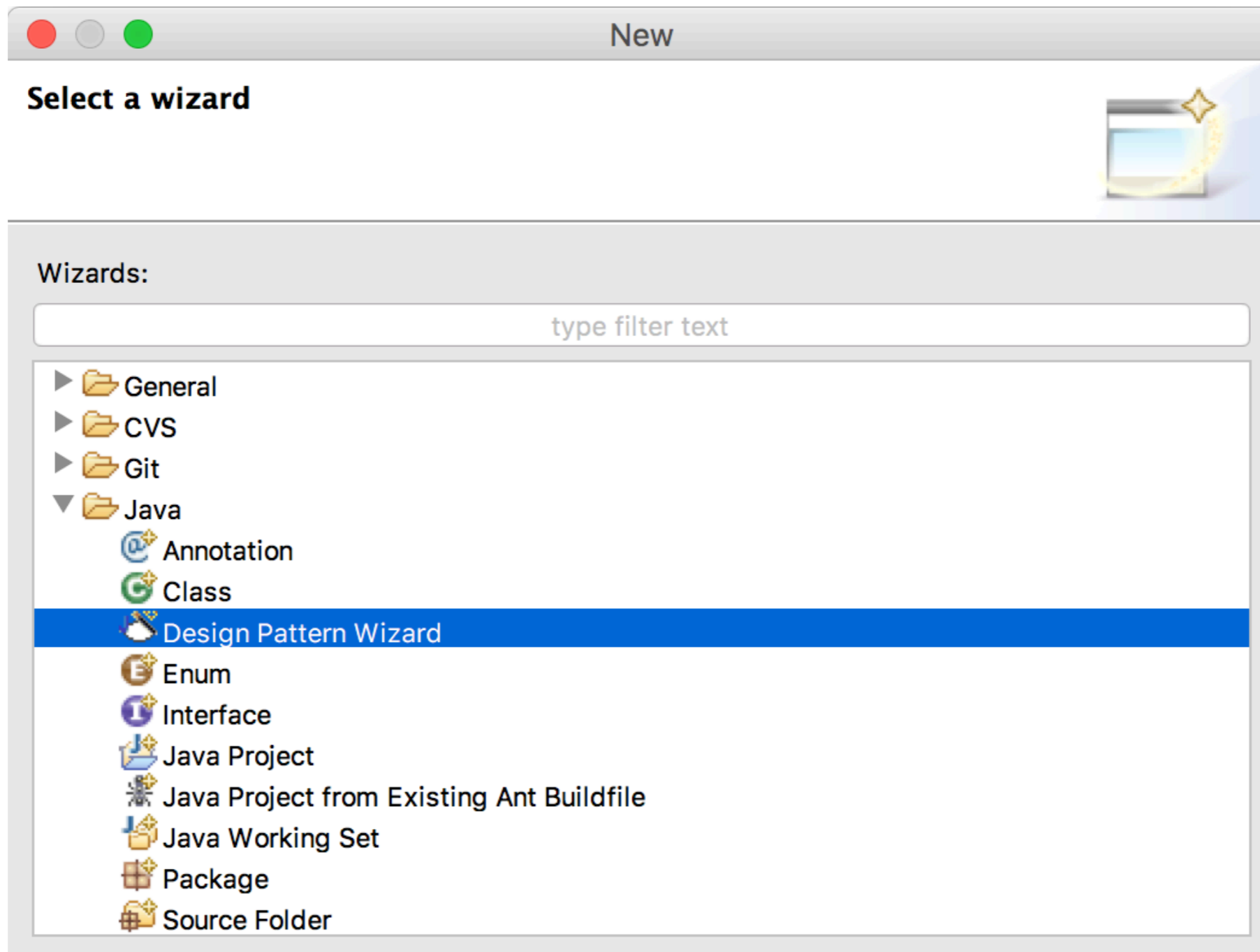
Design Patterns

with Eclipse Juno+PatternBox (obsolete)



Design Patterns

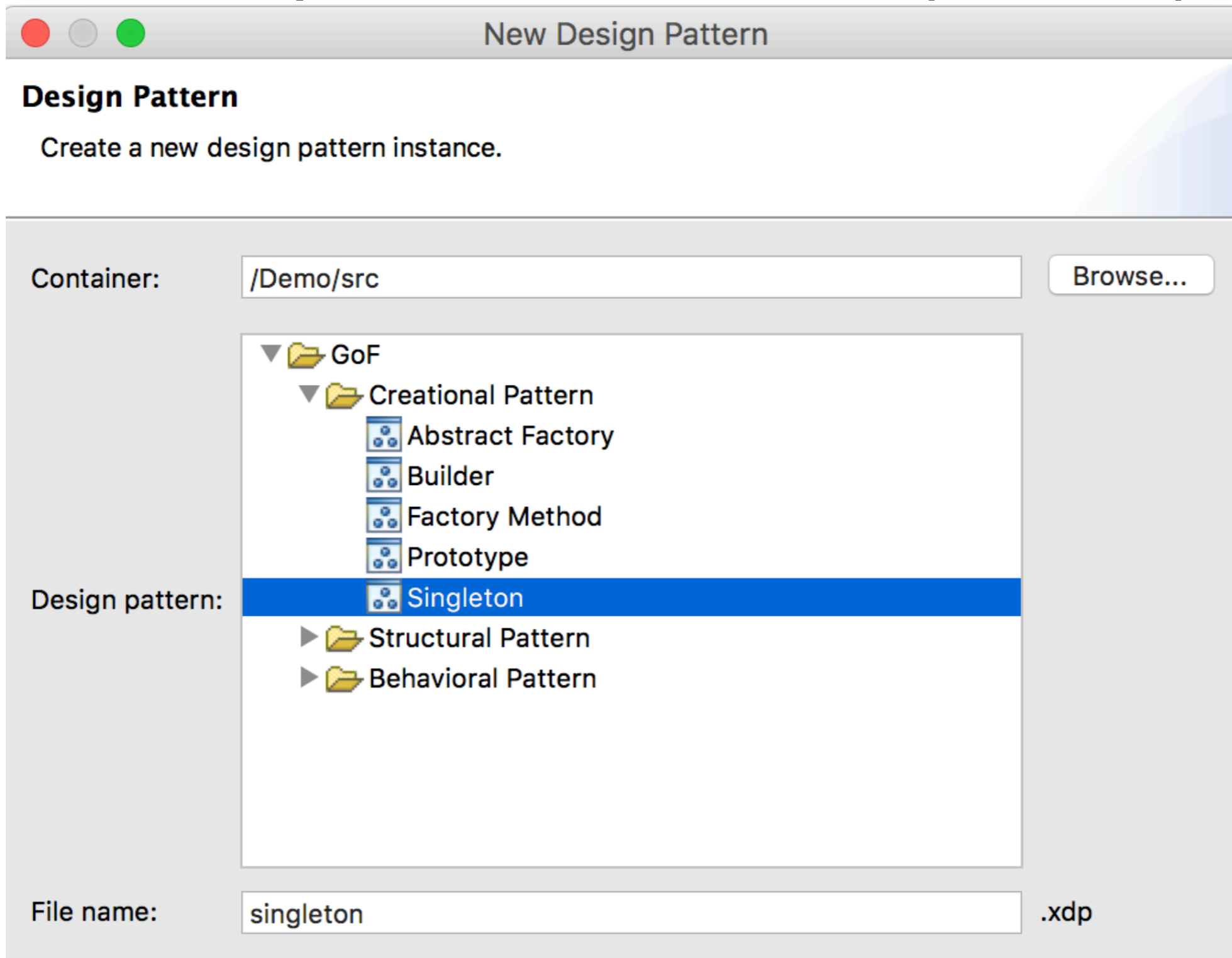
with Eclipse Juno+PatternBox (obsolete)



File / New / Other... / Java / Design Pattern Wizard

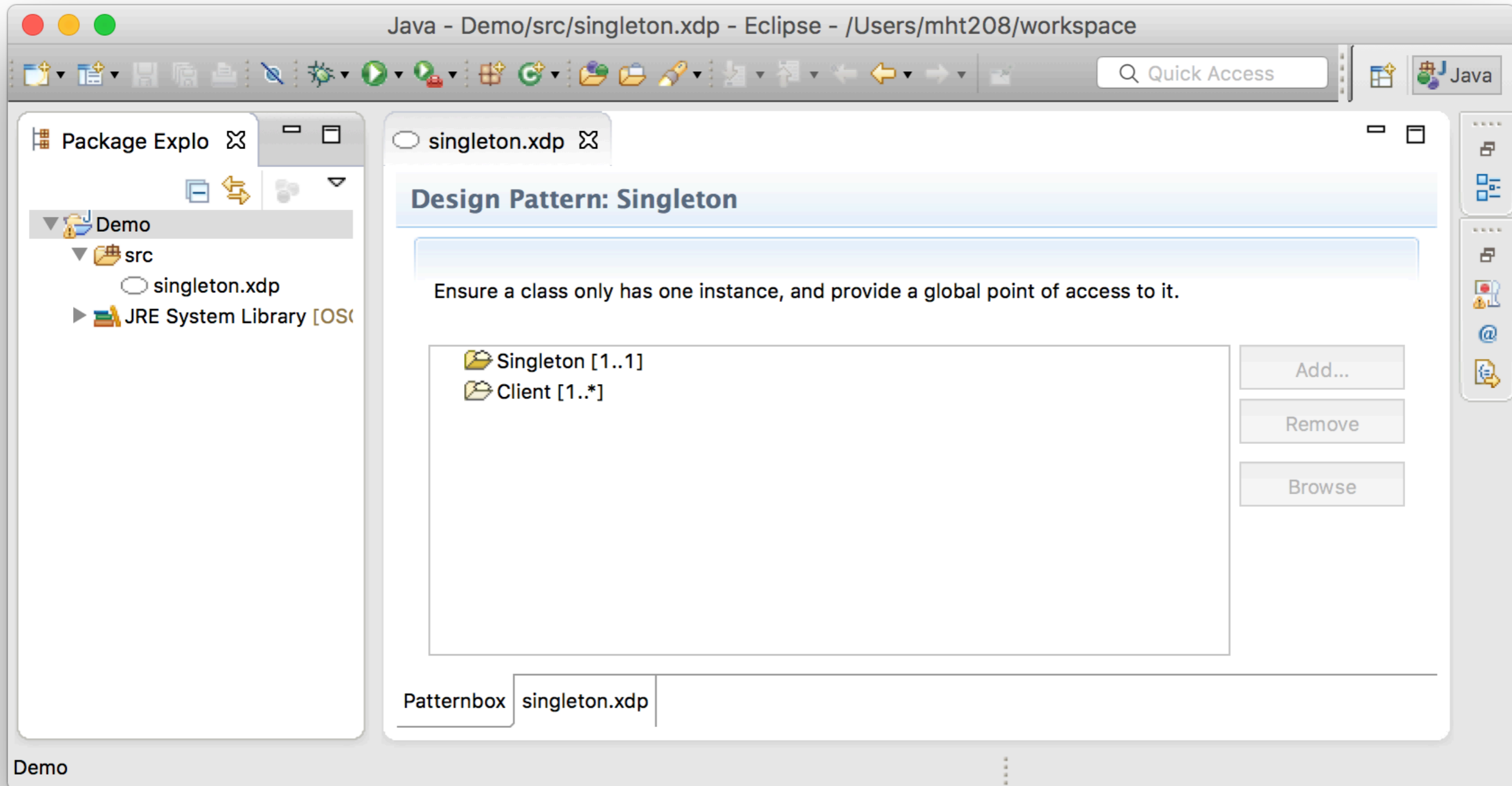
Design Patterns

with Eclipse Juno+PatternBox (obsolete)



Design Patterns

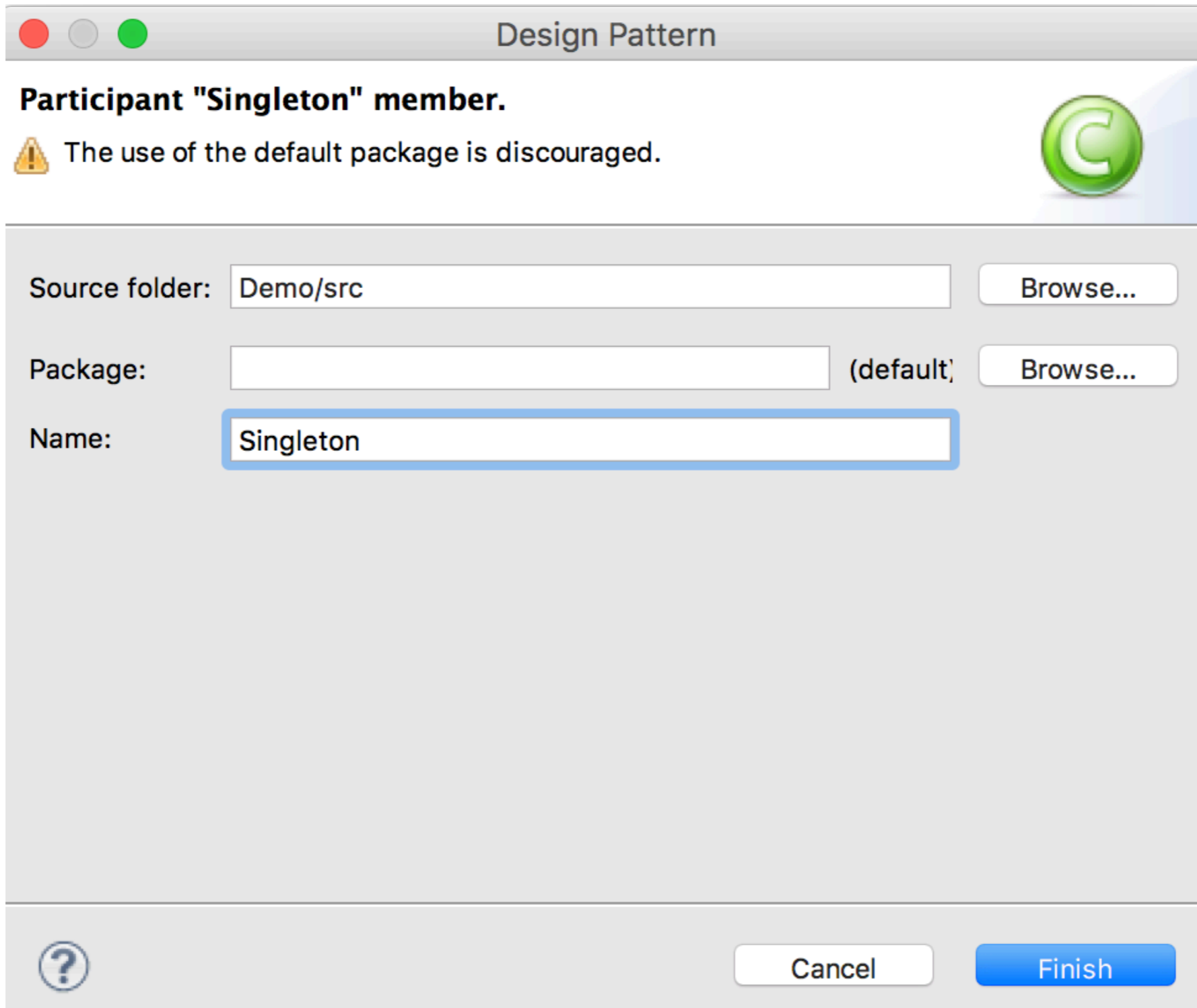
with Eclipse Juno+PatternBox (obsolete)



Add a new singleton

Design Patterns

with Eclipse Juno+PatternBox (obsolete)



The image shows a dialog box titled "Design Pattern" with a standard macOS-style title bar (red, yellow, and green buttons). The main content area has a light blue header with the text "Participant 'Singleton' member." and a warning icon (yellow triangle with an exclamation mark) followed by the text "The use of the default package is discouraged." To the right of this header is a green circular icon with a white 'C' inside. Below the header, there are three input fields and two "Browse..." buttons. The first row is "Source folder:" with a text box containing "Demo/src" and a "Browse..." button. The second row is "Package:" with an empty text box, "(default)" text, and a "Browse..." button. The third row is "Name:" with a text box containing "Singleton" that has a blue selection border. At the bottom of the dialog, there is a question mark icon on the left, a "Cancel" button in the center, and a "Finish" button on the right.

Design Pattern

Participant "Singleton" member.

⚠ The use of the default package is discouraged.

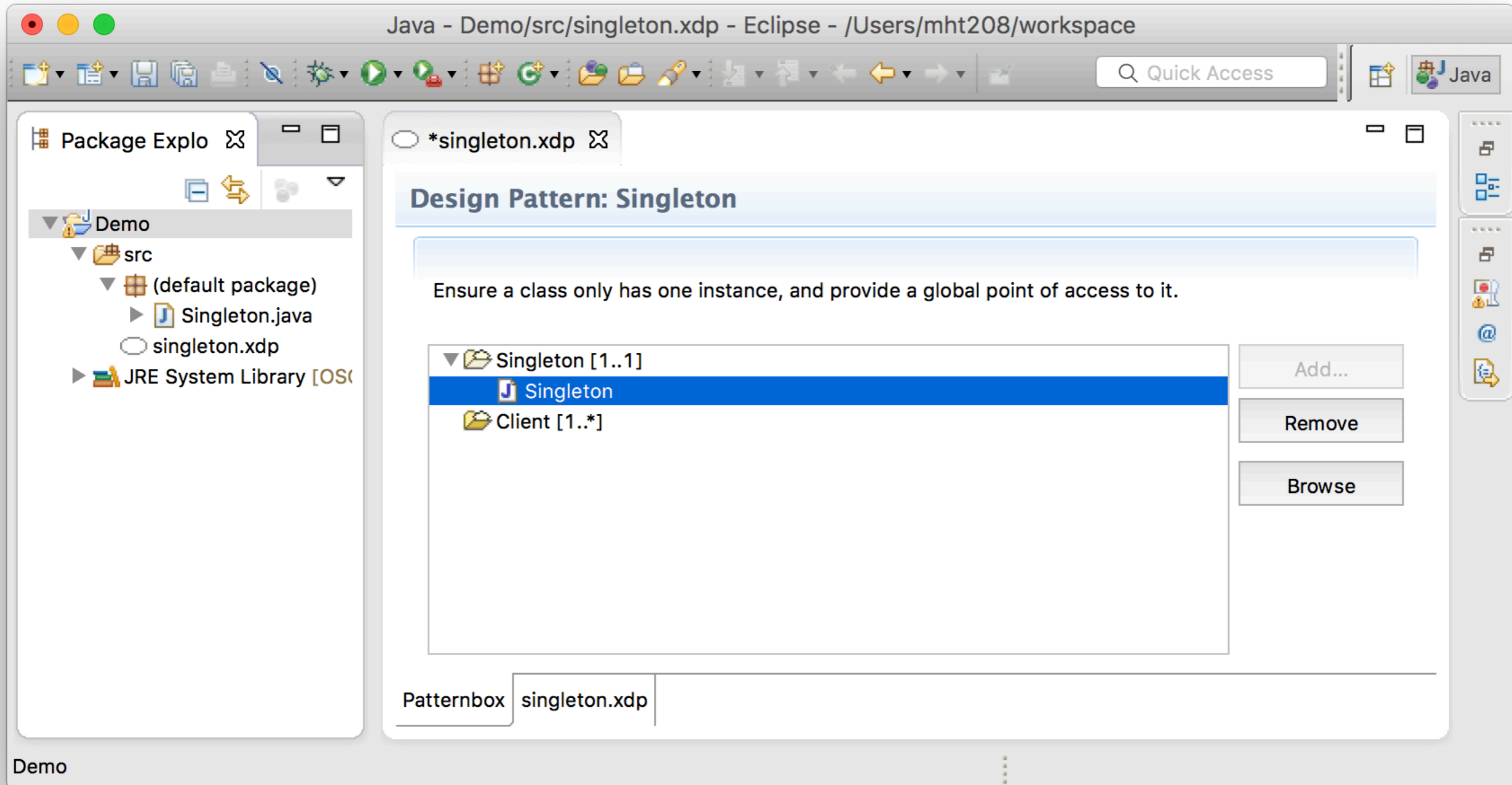
Source folder:

Package: (default)

Name:

Design Patterns

with Eclipse Juno+PatternBox (obsolete)



Design Patterns

with Eclipse Juno+PatternBox (obsolete)

```
public class Singleton {  
  
    /** unique instance */  
    private static Singleton sInstance = null;  
  
    /**  
     * Private constuctor  
     */  
    private Singleton() {  
        super();  
    }  
  
    /**  
     * Get the unique instance of this class.  
     */  
    public static synchronized Singleton getUniqueInstance() {  
  
        if (sInstance == null) {  
            sInstance = new Singleton();  
        }  
  
        return sInstance;  
    }  
}
```


EGit

(with Eclipse Oxygen)

Ming-Hsien Tsai
Academia Sinica

SDM 2020

Use Git with Eclipse

- Create you project in Eclipse
- Switch to a terminal window
- `git init`, `git add`, `git commit`, ...

Use Git with Eclipse

- Create you project in Eclipse

Thank you for listening ?

- git init, git add, git commit, ...

Use Git in Eclipse

- EGit
 - Eclipse Team provider for the Git version control system
 - On top of the JGit Java implementation of Git
 - Available in the Eclipse Marketplace
 - Usually installed by default

Configuration

Eclipse / Preferences... (⌘,) / Team / Git

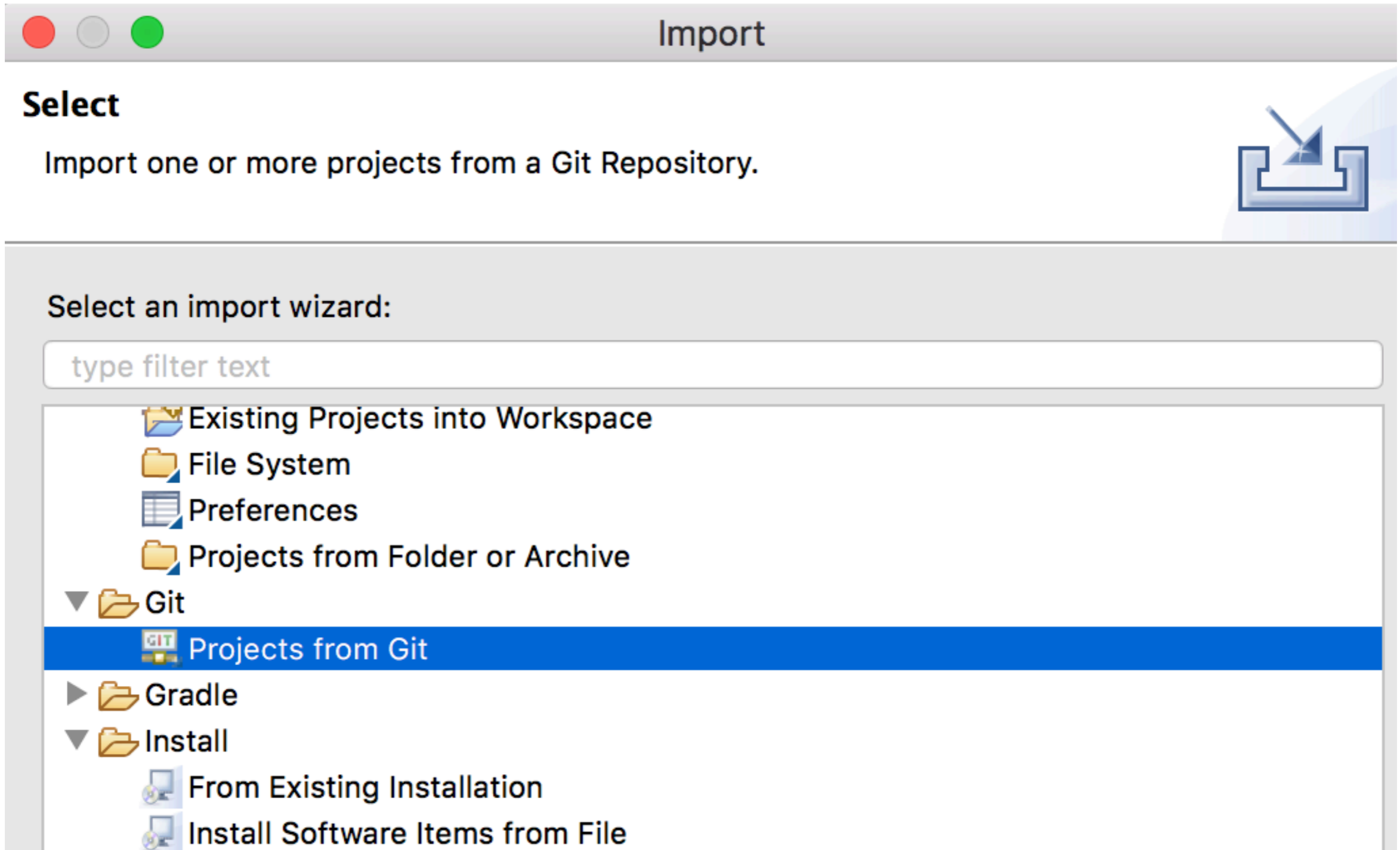
- ▶ Install/Update
- ▶ Java
 - JPF Preferences
- ▶ Maven
- ▶ Mylyn
- ▶ Oomph
- ▶ Plug-in Development
- ▶ Run/Debug
- ▼ Team
 - File Content
 - ▼ Git
 - Committing
 - Configuration**
 - Confirmations and Wa
 - Date Format
 - History
 - Label Decorations
 - Projects
 - Staging View
 - Synchronize
 - Window Cache
 - Ignored Resources
 - Models

Key	Value
▼ alias	
ci	commit
co	checkout
glog	log --graph --decorate --pretty=on
st	status
▼ http	
postBuffer	2M
sslVerify	false
▼ user	
email	mhtsai208@gmail.com
name	Ming-Hsien Tsai

```
git config --global --add user.name "NAME"  
git config --global --add user.email "EMAIL"
```

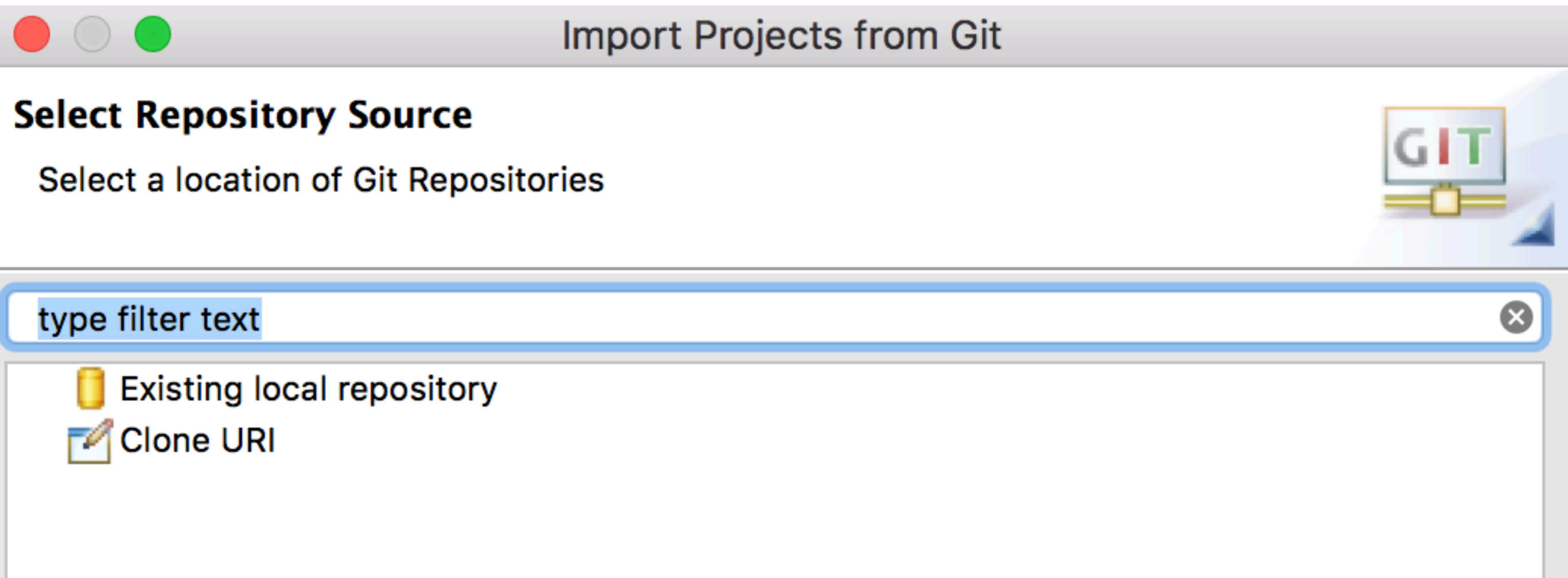
Clone

File / Import... / Git / Projects from Git



Clone

File / Import... / Git / Projects from Git




Clone

File / Import... / Git / Projects from Git

Import Projects from Git

Source Git Repository

Enter the location of the source repository.



Location

URI:

Host:

Repository path:

Connection

Protocol:

Port:

Authentication

User:

Password:

Store in Secure Store

Clone

File / Import... / Git / Projects from Git



Import Projects from Git







Branch Selection

Select branches to clone from remote repository. Remote tracking branches will be created to track updates for these branches in the remote repository.



Branches of <https://github.com/groovy/groovy-eclipse.git>:

type filter text

-  extract-groovy-sources
-  greclipse_batch_2_2_1
-  master
-  plexus-3.0
-  wip1727
-  working_build

Clone

File / Import... / Git / Projects from Git

Import Projects from Git

Local Destination

Configure the local storage location for groovy-eclipse.



Destination

Directory:

Browse

Initial branch:

Clone submodules

Configuration

Remote name:

Clone

File / Import... / Git / Projects from Git

Cloning from <https://github.com/groovy/groovy-eclipse.git>

Select a wizard to use for importing projects

Depending on the wizard, you may select a directory to determine the wizard's scope



Wizard for project import

- Import existing Eclipse projects
- Import using the New Project wizard
- Import as general project

 Working Tree - /Users/mht208/.git/groovy-eclipse


Clone

File / Import... / Git / Projects from Git

Cloning from <https://github.com/groovy/groovy-eclipse.git>


Import Projects

Import projects from a Git repository



Projects:

type filter text to filter unselected projects

 groovy-eclipse (/Users/mht208/.git/groovy-eclipse)

Search for nested projects

Select All

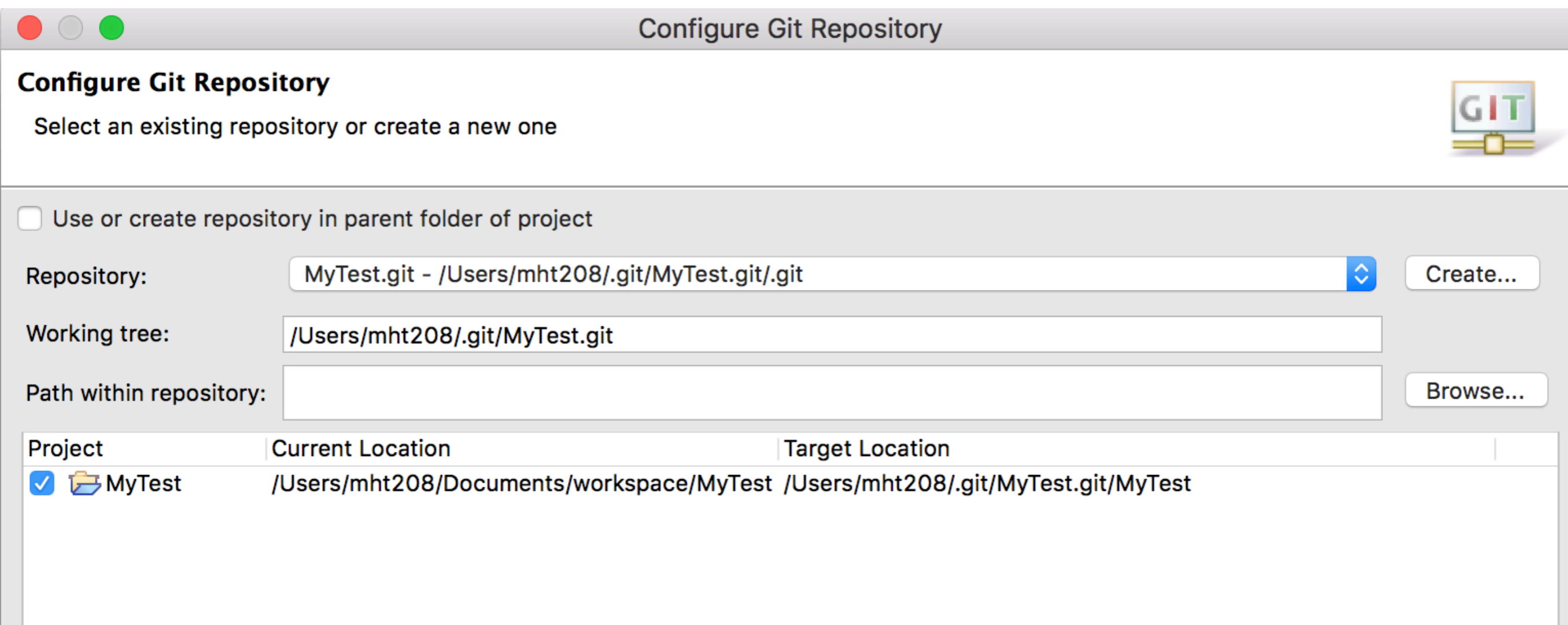
Deselect All

Share Projects

- Project Popup Menu / Team / Share Project...
- Make an initial commit
- Configure push
- Push to remote repositories

Share Projects

- Project Popup Menu / Team / Share Project...



Share Projects

- Project Popup Menu / Team / Share Project...

```
DHCP-22126 .git $ ls -a MyTest.git
```

```
.      ..      .git    MyTest
```

```
DHCP-22126 .git $ ls -a MyTest.git/MyTest
```

```
.      .settings      debug      simple
..     LICENSE       lattice.dot  splash.png
.DS_Store  Lattice.java.backup  lattice.png  src
.classpath  Lattice.java.debug  lattice.svg  wiki
.gitignore  a.dot              lib
.project    bin                num_containment_ce
```

```
DHCP-22126 .git $ █
```

Commit

Project Popup Menu / Team / Commit...

The screenshot shows the Git Staging window in an IDE. The top toolbar includes icons for Problems, Javadoc, Declaration, Search, Console, Progress, Git Repositories, Tasks, Call Hierarchy, and Git Staging. A search box labeled "Filter files" is on the right. The main area is divided into three sections:

- Unstaged Changes (12):** Lists files like `a.ba`, `ch.unifr.goal.complement.zip - plugins`, `input.ba`, and `org.svvl.goal.cmd.strans.zip - plugins`.
- Staged Changes (2):** Lists `AbstractRandomAutomaton.java - plugins/org.svvl.goal.core/source/org/svvl/goal/core` and `TestCommand.java - plugins/org.svvl.goal.cmd/source/org/svvl/goal/cmd`.
- Commit Message:** Contains the message "Change the symbols of classic alphabet." and a "Change-Id" field with a long ID.

At the bottom right, there are two buttons: "Commit and Push..." and "Commit".

Author:

Committer:

Commit

Project Popup Menu / Team / Commit...

The screenshot shows an IDE's commit dialog for a project named "GOAL2 [general-acc]". The dialog is divided into two main sections: "Unstaged Changes (12)" and "Staged Changes (2)".

Unstaged Changes (12):

- a.ba
- ch.unifr.goal.complement.zip - plugins
- input.ba
- org.svrl.goal.cmd.strans.zip - plugins

Staged Changes (2):

- AbstractRandomAutomaton.java - plugins/org.svrl.goal.core/source/org/svrl/goal/core
- TestCommand.java - plugins/org.svrl.goal.cmd/source/org/svrl/goal/cmd

At the bottom of the dialog, there are two buttons: "Commit" and "Commit and Push...". The "Commit" button is highlighted with a red arrow icon.

The IDE's interface is visible in the background, showing the "Problems" view, "Javadoc" view, "Declaration" view, "Search" view, "Console" view, "Progress" view, and "Git Repositories" view. The "Git Repositories" view shows a list of files, including "10000000".

Configure Push

Git Repositories View / Your Local Repository / Remotes / Create Remote...
Project Popup Menu / Team / Remote / Configure Push to Upstream...

The screenshot shows the Eclipse IDE's Git Repositories View. The top toolbar includes icons for Problems, Javadoc, Declaration, Search, Console, Progress, and Git Repositories. The main area displays a tree view of local repositories, including 3DFT, BuchiComplementation, BuchiStore.Admin, BuchiStore.STTT, cpachecker, Eclipse-Markdown-Editor-Plugin, GOAL1, GOAL2, INER2012, jThor, Lattice, and MyTest.git. The 'Remotes' section is expanded, and a context menu is open over the 'Create Remote...' option. The menu includes the text 'Paste Repository Path or URI' and a keyboard shortcut '⌘V'.

Problems @ Javadoc Declaration Search Console Progress Git Repositories

- ▶ 3DFT [NO-HEAD] - /Users/mht208/.git/3DFT/.git
- ▶ > BuchiComplementation [master] - /Users/mht208/.git/BuchiComplementation/.git
- ▶ BuchiStore.Admin [master] - /Users/mht208/.git/BuchiStore.Admin/.git
- ▶ > BuchiStore.STTT [master] - /Users/mht208/.git/BuchiStore.STTT/.git
- ▶ > cpachecker [socket] - /Users/mht208/Work/Sources/cpachecker/.git
- ▶ Eclipse-Markdown-Editor-Plugin [master ↑ 1] - /Users/mht208/.git/Eclipse-Markdown-Editor-Plugin/.git
- ▶ GOAL1 [master] - /Users/mht208/.git/GOAL1/.git
- ▶ > GOAL2 [general-acc] - /Users/mht208/.git/GOAL2/.git
- ▶ > INER2012 [master ↑ 6] - /Users/mht208/.git/INER2012/.git
- ▶ jThor [master] - /Users/mht208/.git/jThor/.git
- ▶ Lattice [master - Bare] - /Users/mht208/.git/Lattice
- ▶ > Lattice [master] - /Users/mht208/Documents/workspace/Lattice/.git
- ▼ > MyTest.git [master] - /Users/mht208/.git/MyTest.git/.git
 - ▶ Branches
 - ▶ Tags
 - ▶ References
 - ▶ Remotes
- ▶ Working Tree - /Users/mht208/.git/
- ▶ > PLTLTranslation [master] - /Users/mht208/.git/PLTLTranslation/.git
- ▶ > ShapeDrawingCore [master] - /Users/mht208/.git/ShapeDrawingCore/.git
- ▶ > ShapeDrawingUI [master] - /Users/mht208/.git/ShapeDrawingUI/.git

Create Remote...
Paste Repository Path or URI ⌘V

Configure Push

Git Repositories View / Your Local Repository / Remotes / Create Remote...
Project Popup Menu / Team / Remote / Configure Push to Upstream...

New Remote

Please enter a name for the new remote

You need to configure the new remote for either fetch or push; you can add configuration for the other direction later

Remote name:

Configure push

Configure fetch



Cancel

OK


Configure Push

Git Repositories View / Your Local Repository / Remotes / Create Remote...
Project Popup Menu / Team / Remote / Configure Push to Upstream...

Select a URI

Source Git Repository

Enter the location of the source repository.



Location

URI:

Host:

Repository path:

Connection

Protocol:

Port:

Authentication

User:

Password:

Store in Secure Store

Configure Push

Git Repositories View / Your Local Repository / Remotes / Create Remote...
Project Popup Menu / Team / Remote / Configure Push to Upstream...
Configure Push

Configure push for remote 'origin'

In order to use a remote for push, you must specify at least one URI and at least one ref mapping



URI:

Change...

Remove

▶ Push URIs

Ref mappings

No Push Refspec, will push currently checked out branch instead.

Add...

Modify...

Delete

Copy

Paste

Advanced...

Save

Dry-Run

Revert

Cancel

Save and Push

Configure Push

Git Repositories View / Your Local Repository / Remotes / Create Remote...
Project Popup Menu / Team / Remote / Configure Push to Upstream...
Configure Push

Configure push for remote 'origin'

In order to use a remote for push, you must specify at least one URI and at least one ref mapping



URI: file:///Users/mht208/.git/MyTestRemove.git

Change...

Remove

► Push URIs

Ref mappings

No Push Refspec, will push currently checked out branch instead.

Add...

Modify...

Delete

Copy

Paste

Advanced...

Add ref mappings (Advanced...)

Save

Dry-Run

Revert

Cancel

Save and Push

Configure Push



Git Repositories View / Your Local Repository / Remotes / Create Remote...
Project Popup Menu / Team / Remote / Configure Push to Upstream...

Add Configured Push Specs

Add All Branches Spec

Add All Tags Spec

Specifications for push

Mode	Source Ref	Destination Ref	Force Update	Remove
 Update	refs/heads/*	refs/heads/*	<input type="checkbox"/>	

Configure Push

Git Repositories View / Your Local Repository / Remotes / Create Remote...
Project Popup Menu / Team / Remote / Configure Push to Upstream...

Configure Push

Configure push for remote 'origin'

In order to use a remote for push, you must specify at least one URI and at least one ref mapping

URI: Change... Remove

► Push URIs

Ref mappings

Add...
Modify...
Delete
Copy
Paste
Advanced...

Save Dry-Run Revert Cancel Save and Push

Configure Push

Git Repositories View / Your Local Repository / Remotes / Create Remote...
Project Popup Menu / Team / Remote / Configure Push to Upstream...

Push Results: MyTest.git - origin

Pushed to MyTest.git - origin



↕ master → master [up to date]

Message Details

Repository <file:///Users/mht208/.git/MyTestRemove.git>

Configure Fetch

Project Popup Menu / Team / Remote / Configure Fetch to Upstream...

Configure fetch for remote 'origin'

⊗ Please provide a ref mapping

Branch:

URI:

Ref mappings



Similar to the configuration for push

Pull

Project Popup Menu / Team / Pull

Pull Result for MyTest.git

Fetch Result



- ▼  master : origin/master [0670b85..a1e8c4f] (1)
 - ▶  0670b85f: Add README. (Ming-Hsien Tsai on 2017-09-20 14:49:16)

Update Result

Result

New HEAD

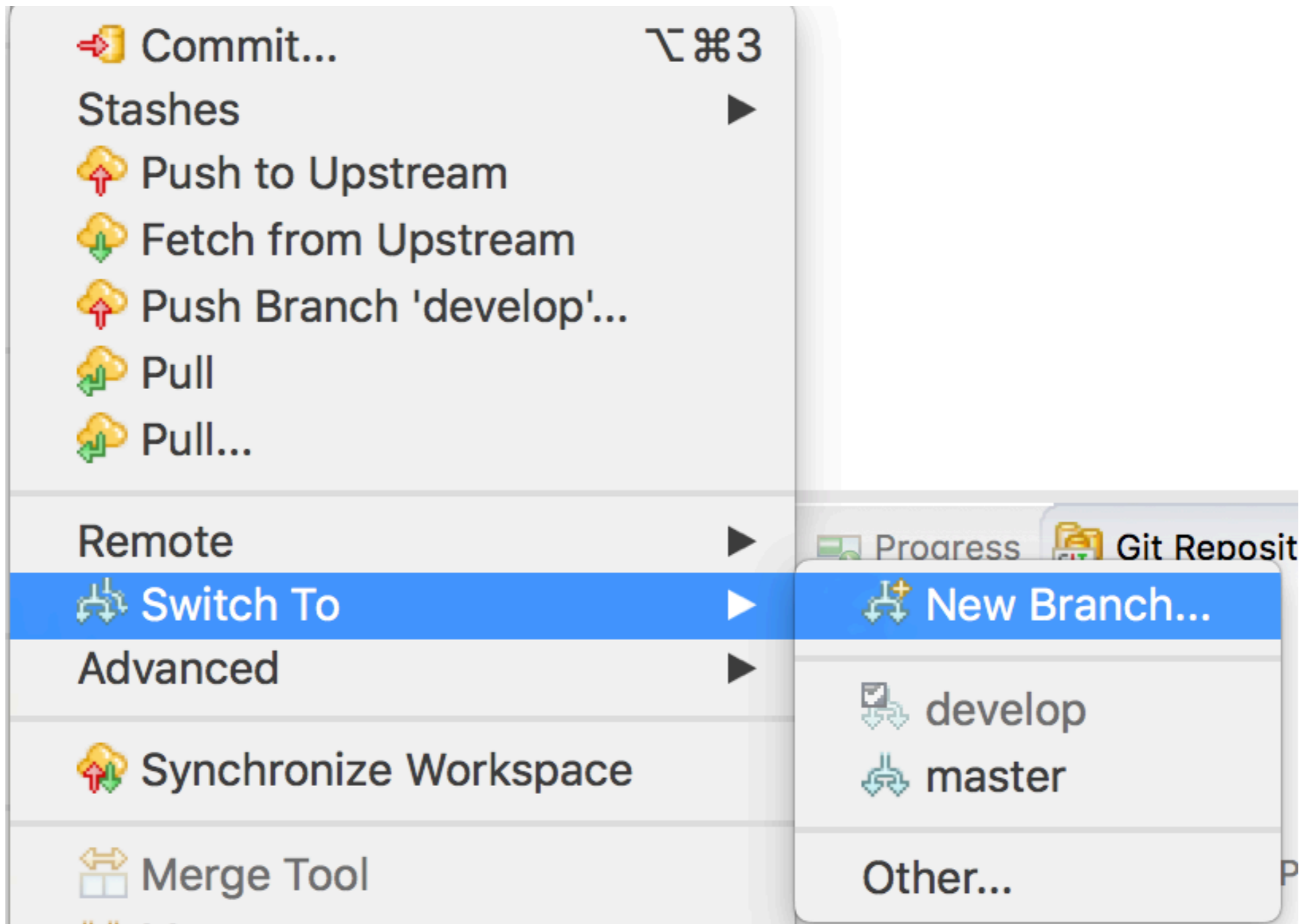
Merge input

-  a1e8c4fa: Initial commit. (Ming-Hsien Tsai on 2017-09-20 14:16:57)
-  0670b85f: Add README. (Ming-Hsien Tsai on 2017-09-20 14:49:16)

Close

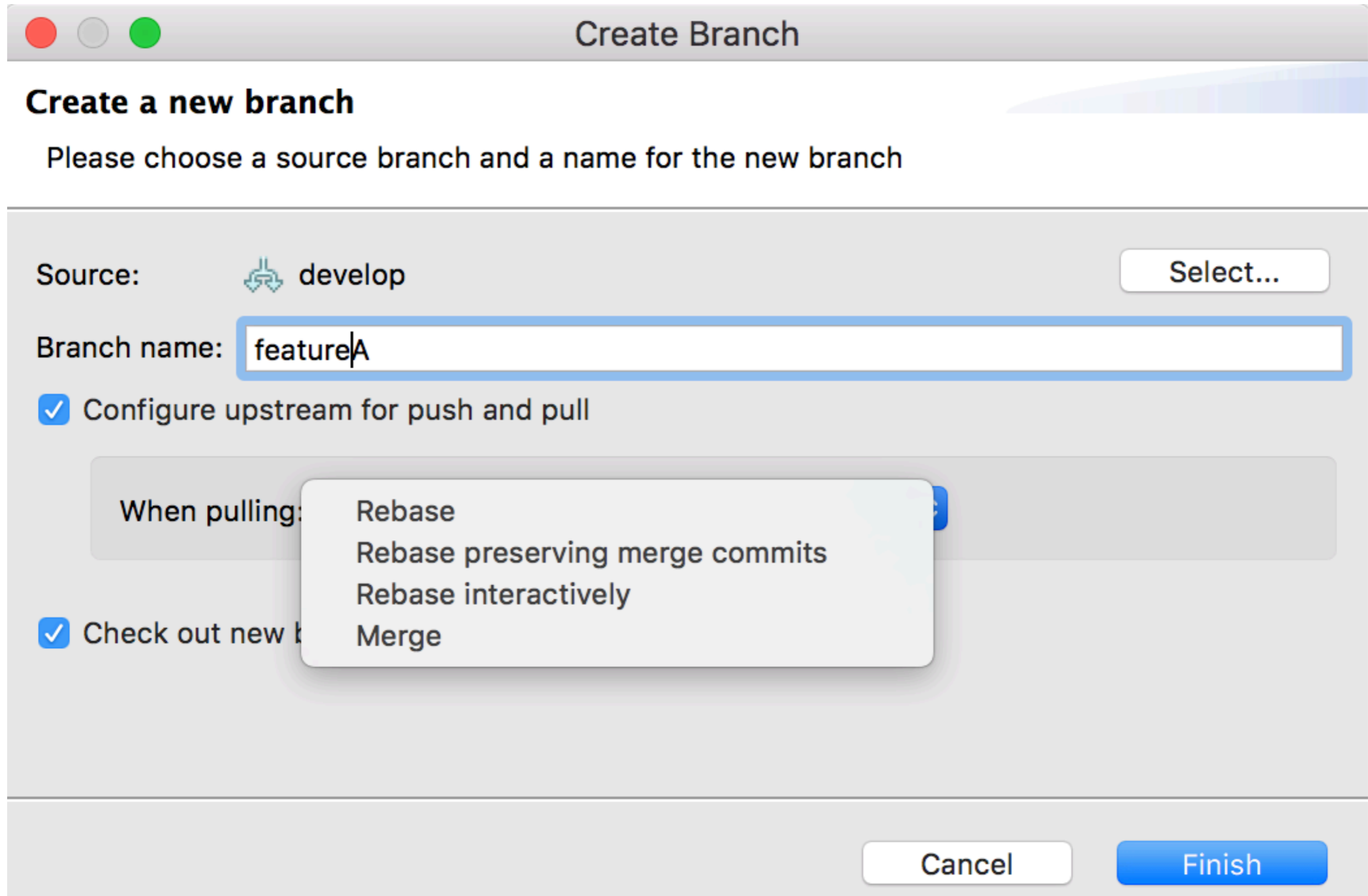
Branches

Project Popup Menu / Team / Switch To



Branches

Project Popup Menu / Team / Switch To



Others

- Revert file
 - File Popup Menu / Replace With
- Rebase
 - Project Popup menu / Team / Rebase...
- Stash
 - Repository Popup Menu / Stashes / Stash Changes...