Eclipse plugins
for code coverage

About Eclipse

- **Eclipse** is a multi-language software development platform comprising an IDE and a plug-in system to extend it. It is written primarily in Java and is used to develop applications in this language and, by means of the various plug-ins, in other languages as well: C/C++, Cobol, Python, Perl, PHP and more.
- Eclipse is **free** and **open source software**.
- The Eclipse SDK includes the Eclipse Java Development Tools, offering an IDE with a built-in incremental Java compiler and a full model of the Java source files. This allows for advanced refactoring techniques and code analysis. The IDE also makes use of a *workspace*, in this case a set of metadata over a flat filespace allowing external file modifications as long as the corresponding workspace "resource" is refreshed afterwards.
- **http://www.eclipse.org/**

Coverlipse

- http://coverlipse.sourceforge.net
- Coverlipse is an Eclipse plugin that visualizes the code coverage of JUnit tests.
- Features:
  - All Uses Coverage
  - Block Coverage (Statement Coverage)
  - *Branch Coverage (Not so soon to come)*
  - Just one test run is needed for evaluation of all coverage criteria
  - Easy way to include/exclude packages from the test
  - Direct feedback in the Eclipse Java Editor
  - Explanation of the results in specialized views

How to install Coverlipse

- http://coverlipse.sourceforge.net/download.php

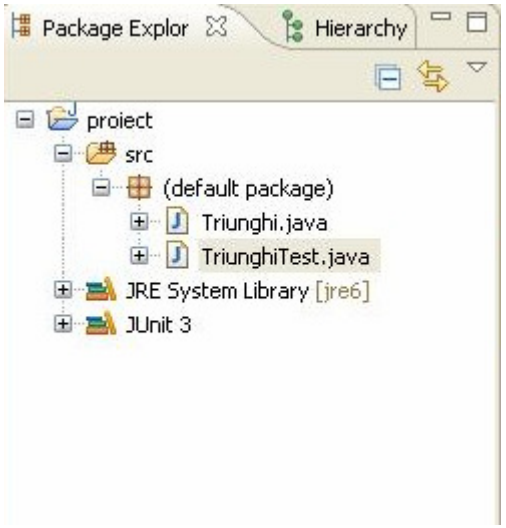
- The eclipse update site is located at http://coverlipse.sf.net/update. When using the update site mechanism you don't need to remove older Coverlipse versions. Eclipse will disable them and only use the newest one.

- *Step-by-step instructions:*
  - In Eclipse, click Help → Software Updates → Find and Install.
  - In the dialog, select Search for new features to install, then Next.
  - In the next step, add a New Remote Site. Name it "Coverlipse update site", the URL is http://coverlipse.sf.net/update/
  - Press Finish. Eclipse now searches for the Coverlipse feature to install and shows that to you.

Using Eclipse

- **Download Eclipse**: http://www.eclipse.org/downloads/
- Unzip the archive
- Double click **eclipse.exe**
- Create a new project, which will contain a class Triunghi, used for triangle classification: **File → New → Java Project**
- In the Package Explorer **right click** on "project_name" **→New → Class** (edit the java class Triunghi.java)
- Make test class:
  In the Package Explorer, **right click** on a java class name (Triunghi.java) **→ New → JUnit Test Case** (**!** Choose New JUnit 3.x test)
- Run test: **right click** on class test (TriunghiTest.java) **→ Run As**
  → 1. JUnit Test or
  → 2. JUnit w/Coverlipse, if you have Coverlipse installed

Running eclipse

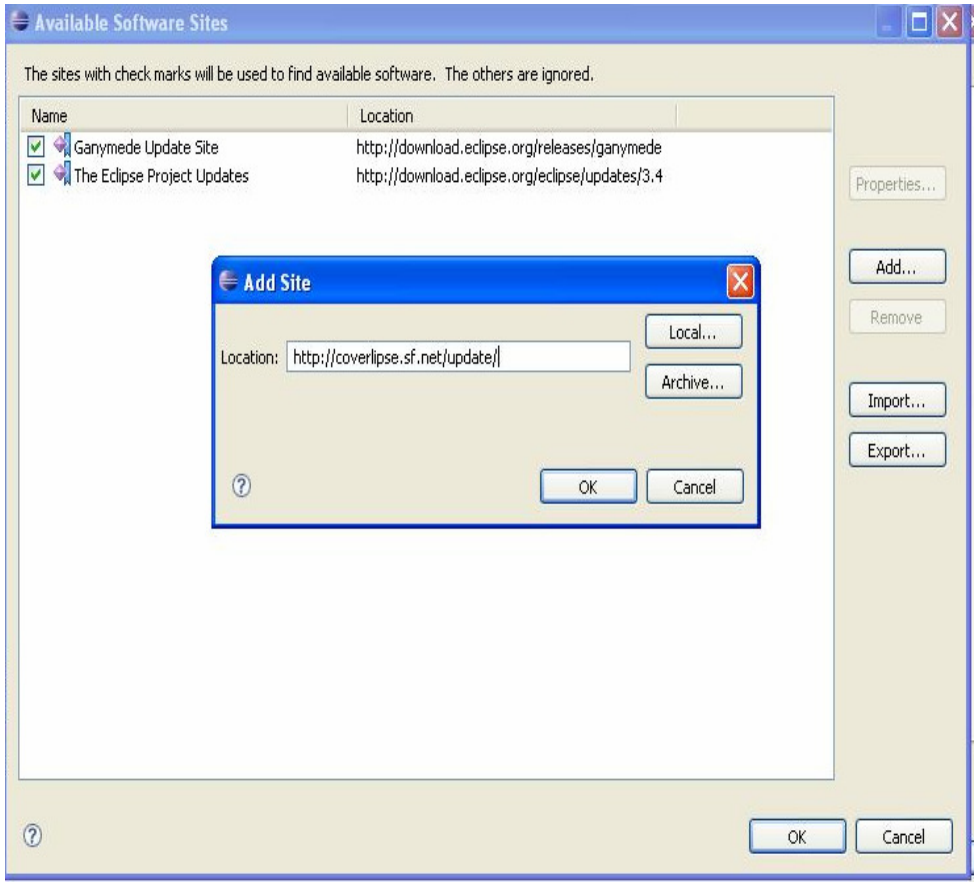


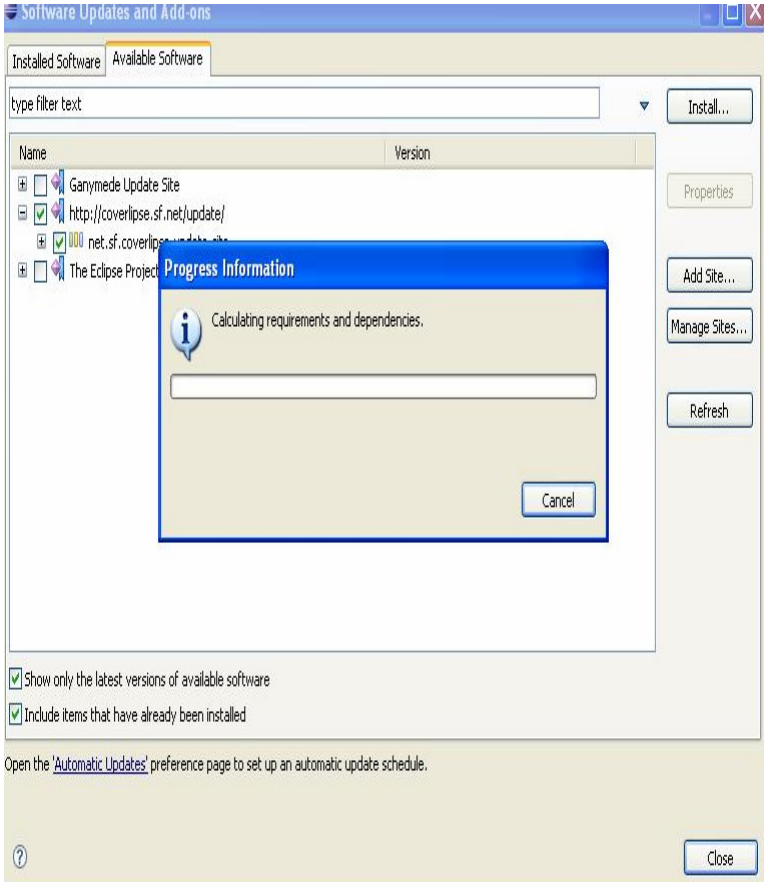
Java Project

JUnit test case

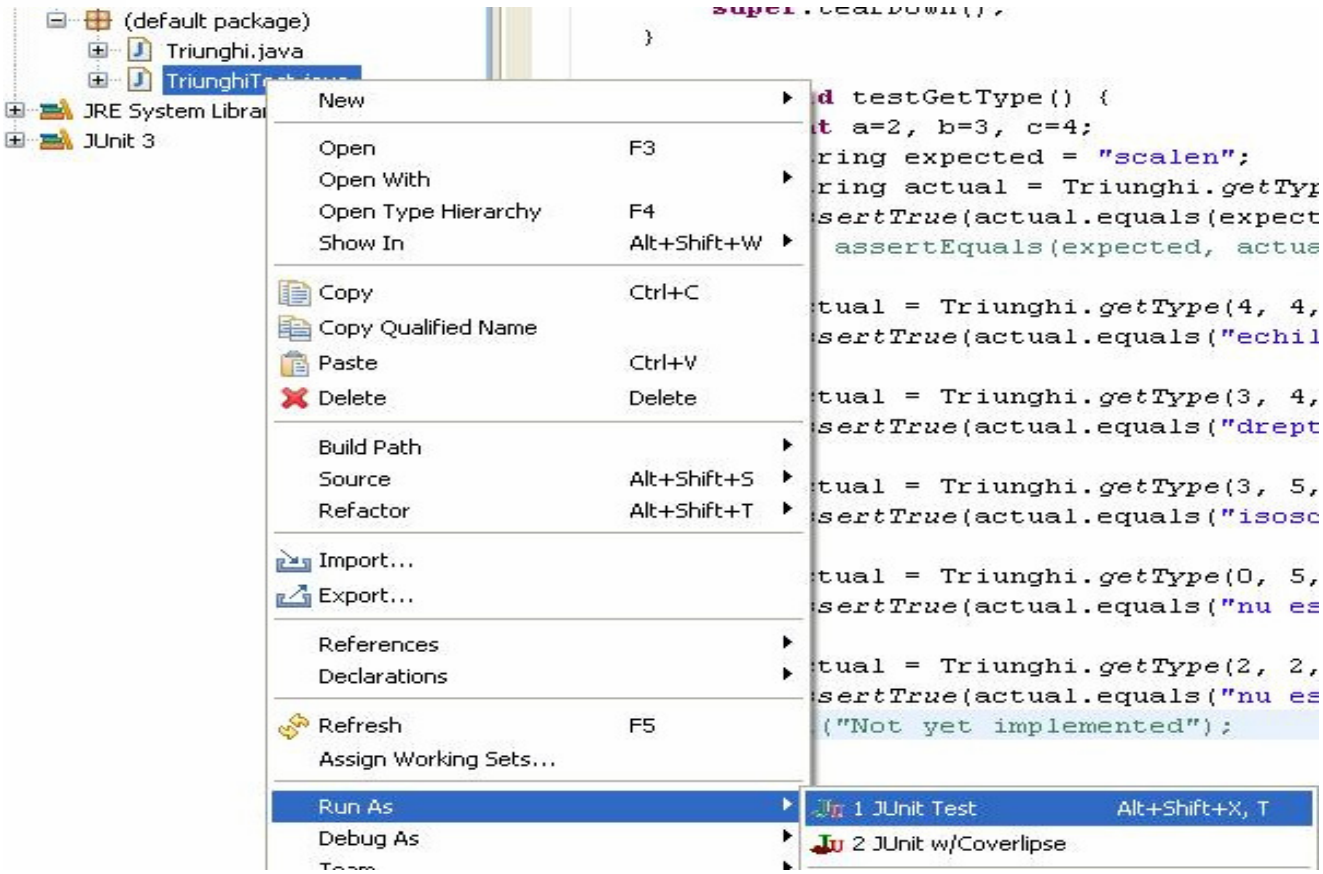
```java
public class Triunghi {

public static String getType(int a, int
    b, int c) {
if (a <= 0 || b <= 0 || c <= 0)
return "nu este triunghi";
else if (a + b <= c || a + c <= b || c +
    b <= a)
return "nu este triunghi";
else if (a == b && b == c)
return "echilateral";
else if (a == b || a == c || b == c)
return "isoscel";
else if (a * a + b * b == c * c || a * a
    + c * c == b * b
|| b * b + c * c == a * a)
return "dreptunghic";
else
return "scalen";
}

public static void main(String[] args) {
// TODO Auto-generated method stub

}

}
```

```java
import junit.framework.TestCase;

public class TriunghiTest extends TestCase {

public void testGetType() {
int a=2, b=3, c=4;
String expected = "scalen";
String actual = Triunghi.getType(a, b, c);
assertEquals(expected, actual);

actual = Triunghi.getType(4, 4, 4);
assertTrue(actual.equals("echilateral"));

actual = Triunghi.getType(3, 4, 5);
assertTrue(actual.equals("dreptunghic"));

actual = Triunghi.getType(3, 5, 5);
assertEquals("isoscel", actual);

actual = Triunghi.getType(0, 5, 5);
assertTrue(actual.equals("nu este
    triunghi"));

actual = Triunghi.getType(2, 2, 5);
assertTrue(actual.equals("nu este
    triunghi"));
}

}
```





Installing Coverlipse plugin



Testing with Coverlipse

Test results with Coverlipse

Packa | Hierarc | JUnit

Finished after 0,172 seconds

Runs: 1/1 Errors: 0 Failures: 0

TriunghiTest (0,000 s)
testGetType (0,000 s)

Failure Trace

Triunghi.java | TriunghiTest.java

```
    }

    protected void tearDown() throws Exception {
        super.tearDown();
    }

    public void testGetType() {
            int a=2, b=3, c=4;
            String expected = "scalen";
            String actual = Triunghi.getType(a, b, c);
            assertTrue(actual.equals(expected));
            // assertEquals(expected, actual);

            actual = Triunghi.getType(4, 4, 4);
            assertTrue(actual.equals("echilateral"));

            actual = Triunghi.getType(3, 4, 5);
            assertTrue(actual.equals("dreptunghic"));
```

Outline

- import declaratic
- TriunghiTest
  - TriunghiTes
  - setUp()
  - tearDown()
  - testGetTyp

Problems | Javadoc | Declaration | Coverlipse Class View | Coverlipse Markers View | Show definitions

Content Description of CoverageMarkerView

Message	L...	covered uses	uncovered uses	Resour
This line was not covered	1			Triungh
This line was fully covered	4			Triungh
This line was fully covered	5			Triungh
This line was fully covered	6			Triungh
This line was fully covered	7			Triungh
This line was fully covered	8			Triungh
This line was fully covered	9			Triungh
This line was fully covered	10			Triungh
This line was fully covered	11			Triungh
This line was fully covered	12			Triungh

JCoverage

- http://cms.jcoverage.com/
- Eclipse plugin (not free)
- Features:

  - Line coverage

  - Branch coverage

  - LOC, number of methods

  - Cyclomatic complexity (1-10 low, 11-20 moderate, 21-50 high, 51+ very high)

  - Viewer integrated with Eclipse;

  - Can export statistics in HTML

  - ***Instructions for installing JCoverage*** *(the same as for installing Coverlipse, just changing the update link http://eclipse.jcoverage.com):*

  http://cms.jcoverage.com/products/eclipse-plugin/installation-instructions.html

Installing JCoverage

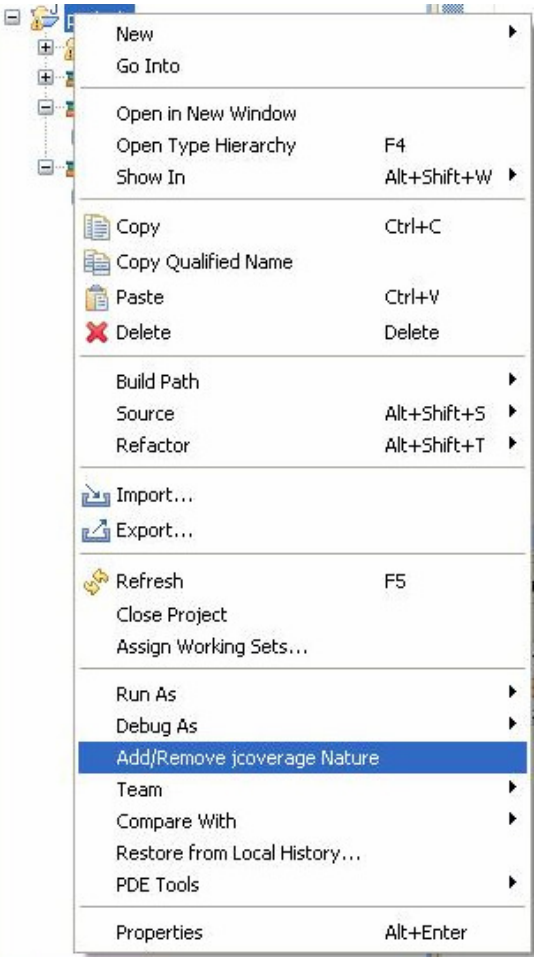
- A short tutorial

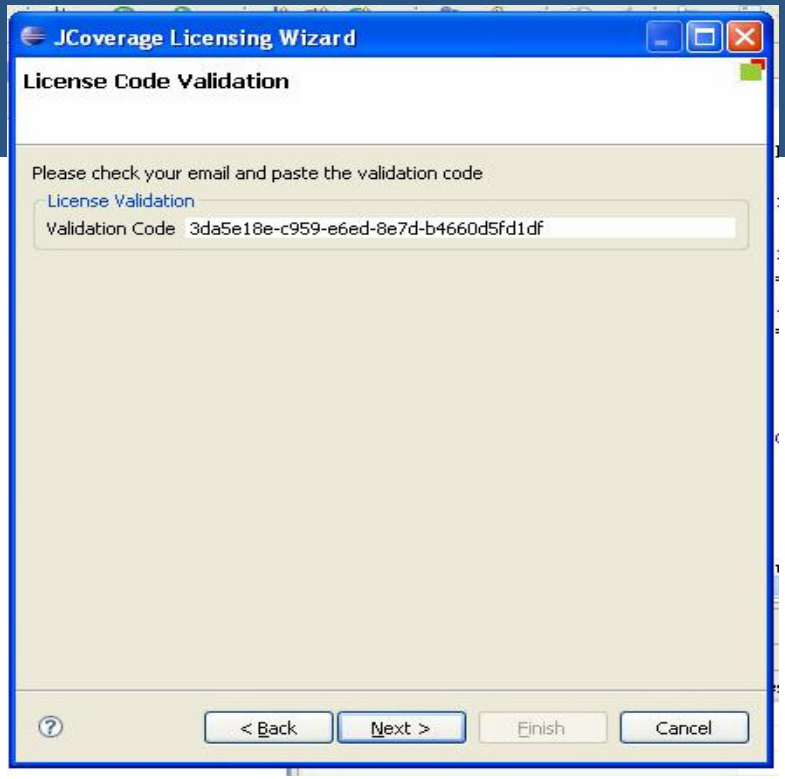
  http://cms.jcoverage.com/products/eclipse-plugin/two-minute-tutorial.html

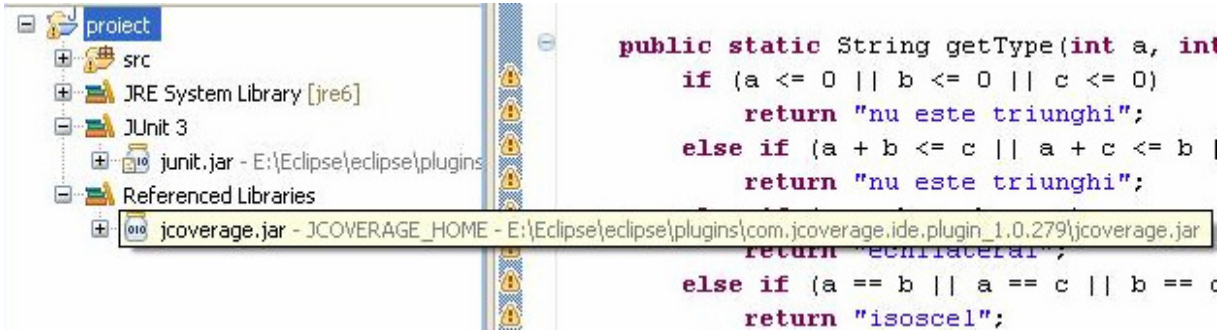
- *Add the **jcoverage** nature to any eclipse project that you wish to instrument:*

  Right click on project →Add/Remove jcoverage Nature

  Obtain a validation code by fulfilling the

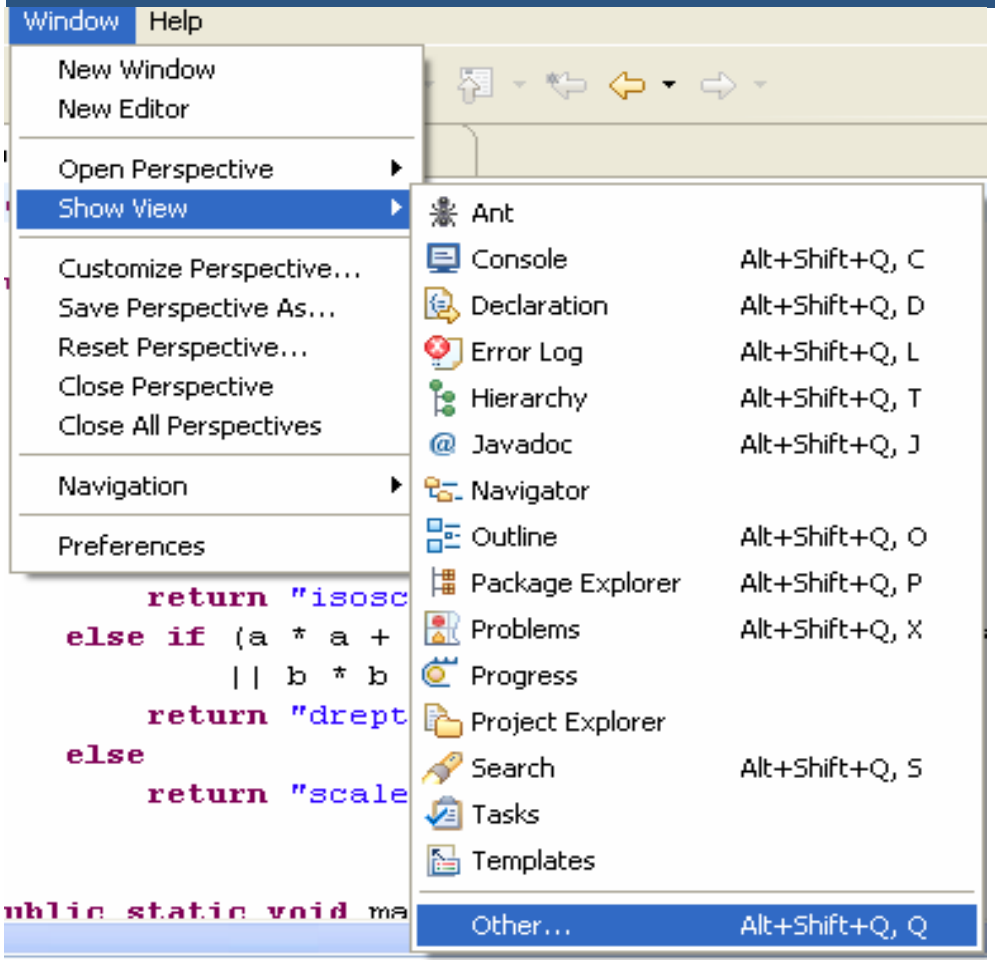


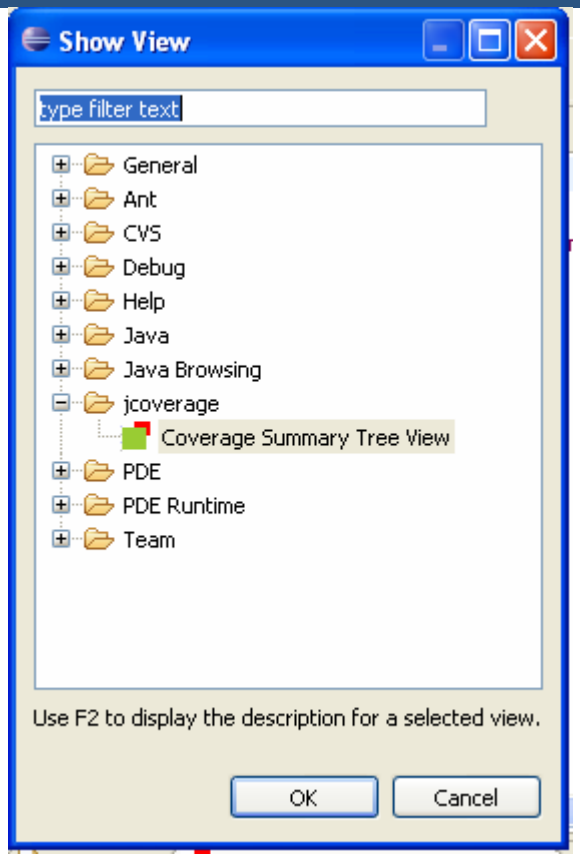




1. ***Add the jcoverage nature to any eclipse project***

Projects that have the jcoverage nature have a jcoverage library added to their classpath.





2. How to display cover information about the eclipse project.

Test results with JCoverage

Problems | Javadoc | Declaration | Coverage Summary Tree View

	Lines	Line Coverage	Branch Cover...	Meth...	Complexity	Avg Method...
proiect	12	0%	0%	3	High (29)	4
	12	0%	0%	3	High (29)	4
Triunghi	12	0%	0%	3	High (29)	4
Triunghi()	0	100%	100%	1	Low (1)	0
getType(int, int, int)	12	0%	0%	1	High (29)	12
main(String[])	0	100%	100%	1	Low (1)	0

Coverage Server

Thank you!