Auto-SCST

About Auto-SCST:

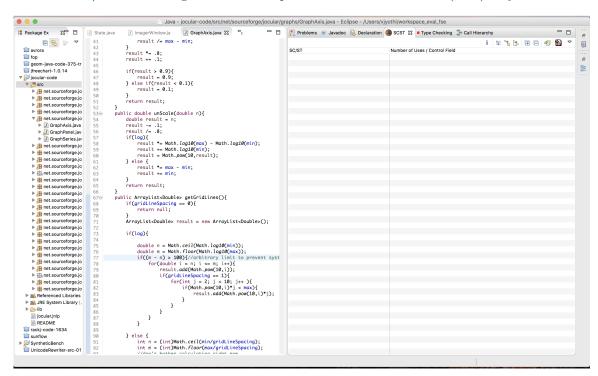
Auto-SCST is a user guided automatic refactoring Eclipse plug-in. It performs "Replace Type Code with State (ST)/Replace Type Code with Subclass (SC)" refactoring followed by "Replace Conditional with Polymorphism" refactoring (RCP).

It follows a 3 phase refactoring approach:

- 1. Identifying refactoring opportunities automatically.
- 2. Collecting developers choice(s) such as the selection of the refactoring opportunity to apply refactoring and subclasses' names.
- 3. Automatic Refactoring.

Installation Steps:

- 1. Download the plug-in (jar file) and move to the plugins folder of your Eclipse application.
- 2. Start your Eclipse from command line using command "./Eclipse -vmargs -Xmx4000m -Xms512m". You can choose the parameters according to your machine's configuration. You can also set these parameter in the configuration file of Eclipse.
- 3. A new entry SCST will be added to the menu bar of your Eclipse.
- 4. Click on SC/ST Refactoring item in the drop down list to see a new tab (view) on your editor.



Options in Auto-SCST

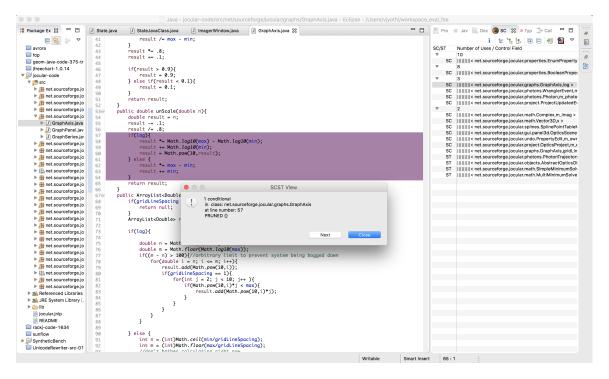
- 1. Identification of refactoring opportunities:
 - i Select the src folder or the any source package of your Java project on which you want to apply RCP refactoring.
 - ii Click on the "i" button in the tab to start the identification step.
 - iii Results are displayed in the tab after the identification is done.

```
Package Ex ⊠ □ □
                                                                                                                                                                                                       ☐ ImagerWindow.java 🎇
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              Pro @ Jav 🚇 Dec 📵 SC 🔀 Typ 👺 Call 🖳 🗖
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               i 🖫 🐈 🕒 🖃 😭
                         ₽

    Copyright (c) 2013, Kenneth MacCallum, Bryan Matthews
    All rights reserved.

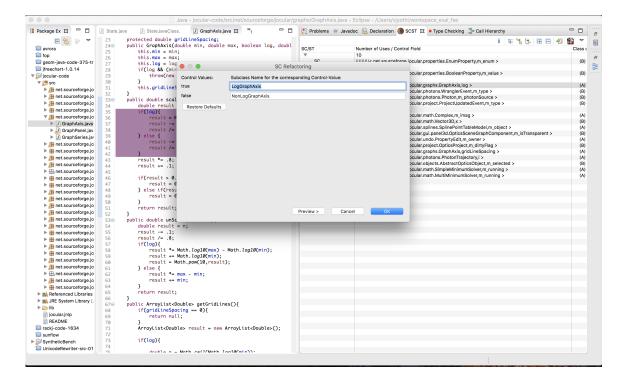
                                                                                                    * Redistribution and use in source and binary forms, with or without modification, are permitted provided that t
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     in not sourceforge, in met sourceforge, in met sourceforge, in it is 
                                                                                                    * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANT
                                                                                       10 * THIS SOFTWARE IS PROVIDED BY T
11 package net.sourceforge.jocular;
13
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             13
14⊕ import java.awt.BorderLayout;[
                                                                                                ##SuppressWarnings("serial")
public class Imager#indow extends JFrame implements ProjectUpdatedListener, WranglerListener{
static final String MAIN_ICOM_PATH = "/net/sourceforge/jocular/icoms/jocular_icon48.png";
                                                                                                               OpticsProject m_project;
ArrayList<OutputObject> m_objects;
                                                                                                               final ImagerMenuBar m_menuBar;
                                                                                                               ImagerToolBar m_toolBar;
ImagerStatusBar m_statusBar;
JTabbedPane m_tobbedPane;
//ImagerPanel m_imagerPanel;
BufferedImage m_image;
Jocular m_app;
Jocular m_app;
ArroyList<OutputPanel> m_outputPanels = new ArroyList<OutputPanel>();
                                                                                                                            m_app = app;
m_menuBar = new ImagerMenuBar();
m_tabbedPane = new JTabbedPane();
m_tabbedPane.addChangeListener(new ChangeListener(){
                                                                                                                                          @Override
public void stateChanged(ChangeEvent e) {
    updatePanels();
                                                                                                                          3);
                                                                                                                             initializeDisplay();
```

iv Double-click on any entry in the displayed list to see all the conditional (IF/SWITCH) statements associated with the refactoring opportunity.



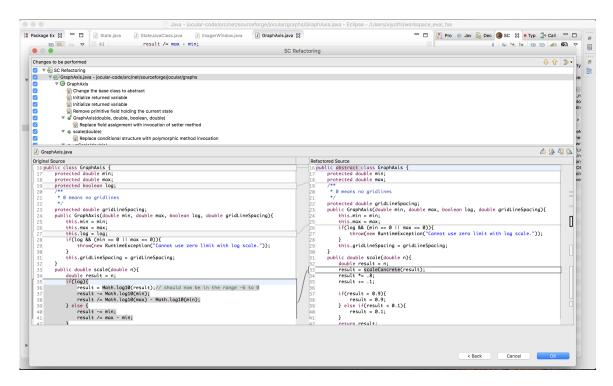
2. Developer's intervention:

- i You can select any entry in the list and click on refactoring button in the tab to apply the RCP refactoring.
- ii Choose your own subclass names or the default names provided by Auto-SCST.



3. Refactoring:

- i Click on "Preview" button to see the set of changes that will be made on the code if refactoring is applied.
- ii Click on "OK" button to apply the RCP refactoring.



Note: (1) The tool may throw some errors when tested on new projects (other than tested benchmarks) if the projects use many external libraries. (2) You may need to restart your eclipse after each run of the tool. The tool is still a prototype implementation and can easily be extended to handle all cases.