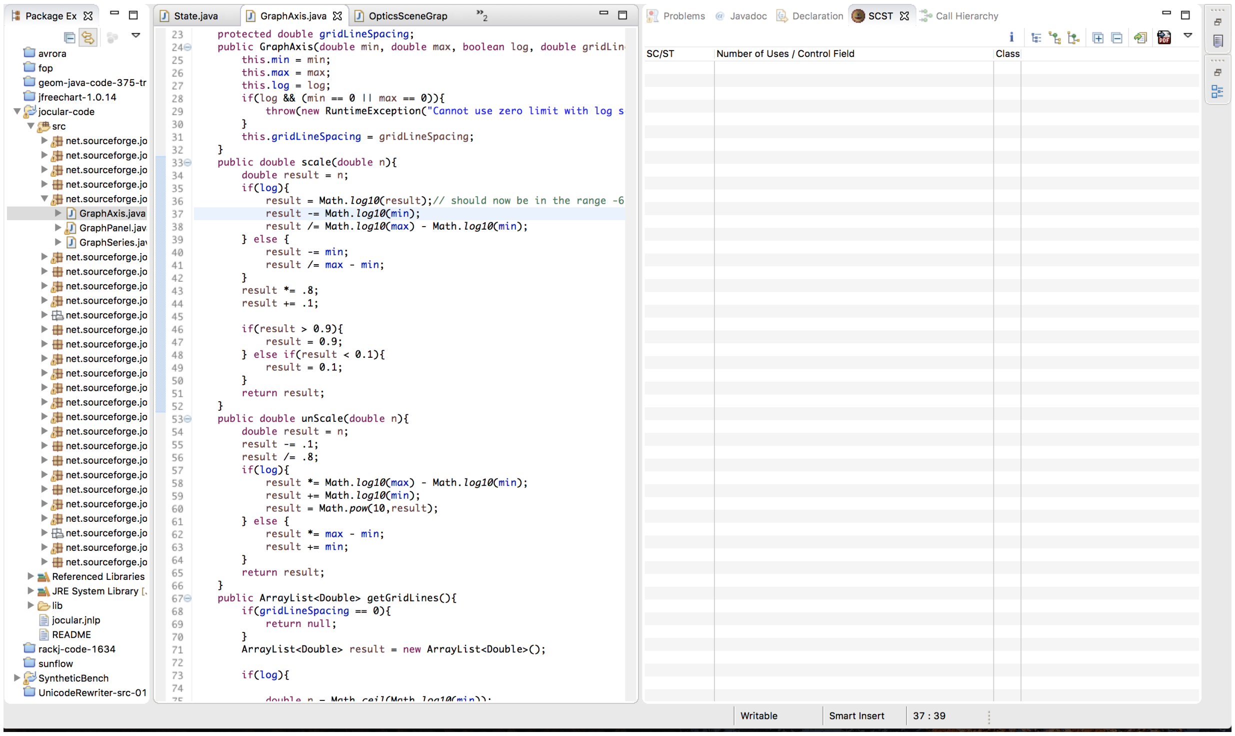
Auto-SCST

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.



```java
    protected double gridLineSpacing;
    public GraphAxis(double min, double max, boolean log, double gridLin
        this.min = min;
        this.max = max;
        this.log = log;
        if(log && (min == 0 || max == 0)){
            throw(new RuntimeException("Cannot use zero limit with log s
        }
        this.gridLineSpacing = gridLineSpacing;
    }
    public double scale(double n){
        double result = n;
        if(log){
            result = Math.log10(result);// should now be in the range -6
            result -= Math.log10(min);
            result /= Math.log10(max) - Math.log10(min);
        } else {
            result -= min;
            result /= max - min;
        }
        result *= .8;
        result += .1;

        if(result > 0.9){
            result = 0.9;
        } else if(result < 0.1){
            result = 0.1;
        }
        return result;
    }
    public double unScale(double n){
        double result = n;
        result -= .1;
        result /= .8;
        if(log){
            result *= Math.log10(max) - Math.log10(min);
            result += Math.log10(min);
            result = Math.pow(10,result);
        } else {
            result *= max - min;
            result += min;
        }
        return result;
    }
    public ArrayList<Double> getGridLines(){
        if(gridLineSpacing == 0){
            return null;
        }
        ArrayList<Double> result = new ArrayList<Double>();

        if(log){

            double n = Math.ceil(Math.log10(min));
```

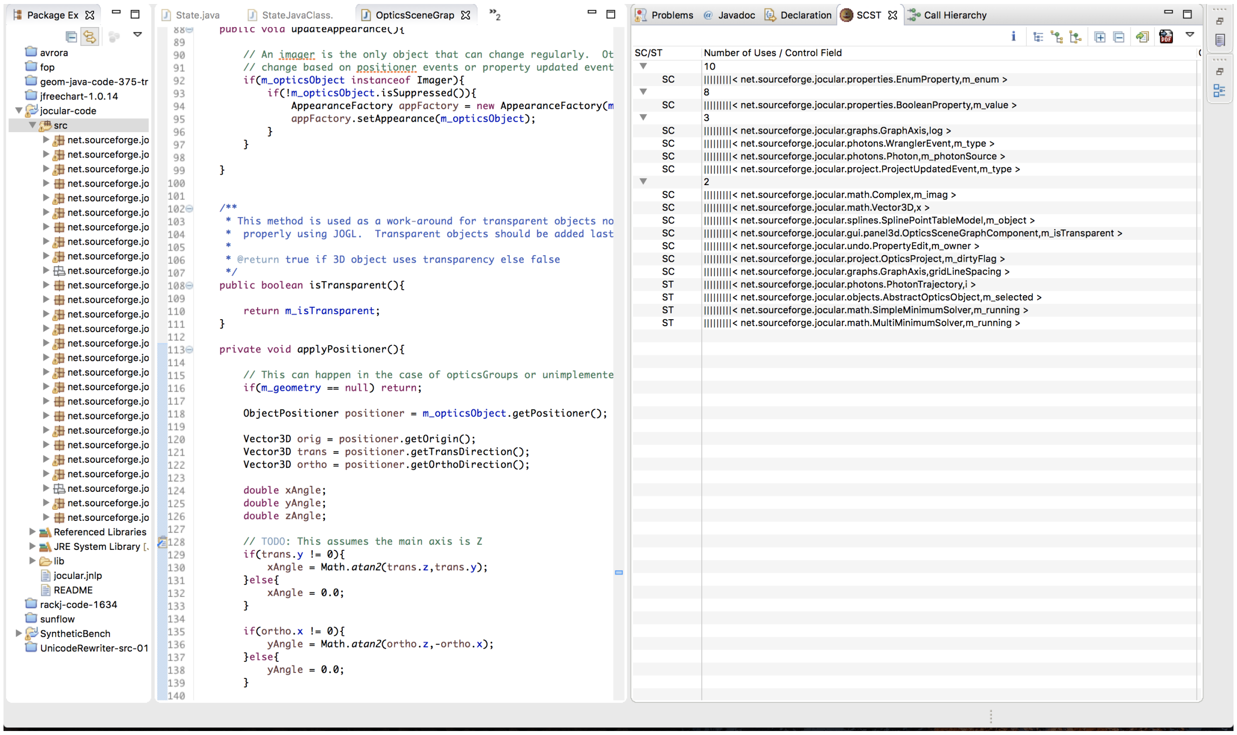
Options in Auto-SCST

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.

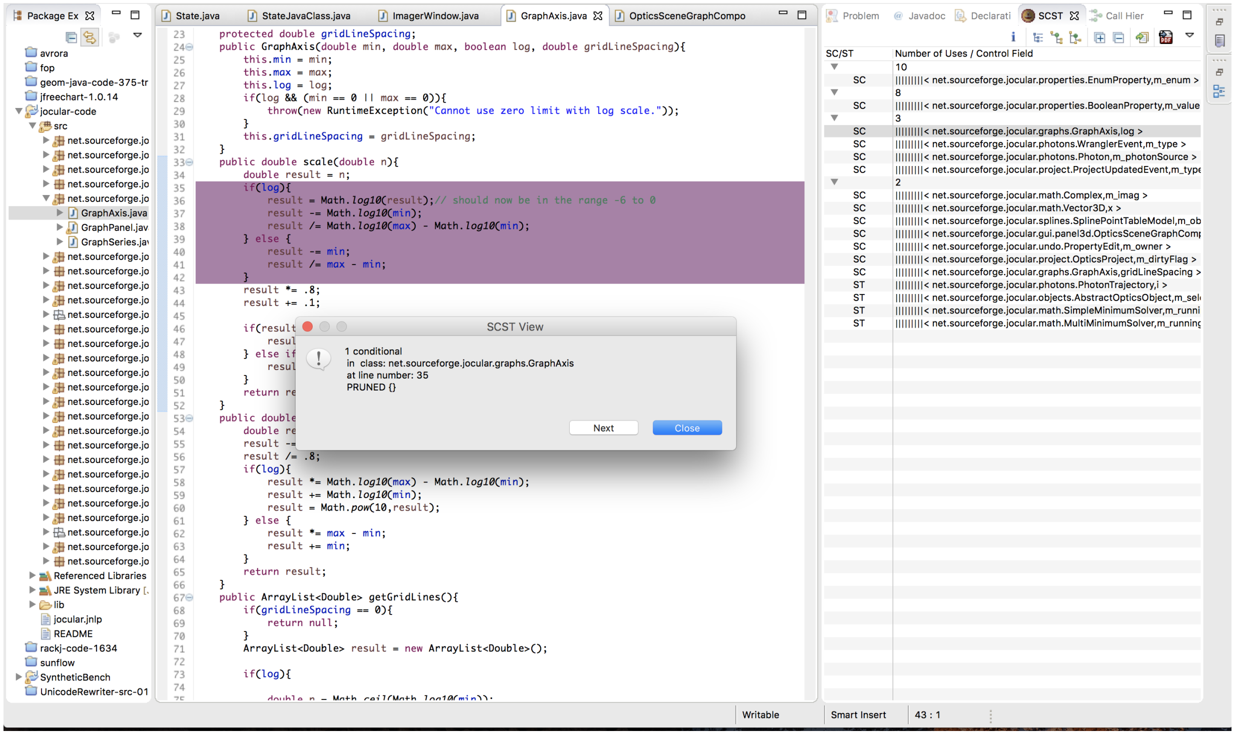
   iii Results are displayed in the tab after the identification is done.



```
    public void updateAppearance(){

        // An imager is the only object that can change regularly.  Ot
        // change based on positioner events or property updated event
        if(m_opticsObject instanceof Imager){
            if(!m_opticsObject.isSuppressed()){
                AppearanceFactory appFactory = new AppearanceFactory(m
                appFactory.setAppearance(m_opticsObject);
            }
        }

    }


    /**
     * This method is used as a work-around for transparent objects no
     *  properly using JOGL.  Transparent objects should be added last
     *
     * @return true if 3D object uses transparency else false
     */
    public boolean isTransparent(){

        return m_isTransparent;
    }

    private void applyPositioner(){

        // This can happen in the case of opticsGroups or unimplemente
        if(m_geometry == null) return;

        ObjectPositioner positioner = m_opticsObject.getPositioner();

        Vector3D orig = positioner.getOrigin();
        Vector3D trans = positioner.getTransDirection();
        Vector3D ortho = positioner.getOrthoDirection();

        double xAngle;
        double yAngle;
        double zAngle;

        // TODO: This assumes the main axis is Z
        if(trans.y != 0){
            xAngle = Math.atan2(trans.z,trans.y);
        }else{
            xAngle = 0.0;
        }

        if(ortho.x != 0){
            yAngle = Math.atan2(ortho.z,-ortho.x);
        }else{
            yAngle = 0.0;
        }

```

SC/ST	Number of Uses / Control Field
▼	10
SC	|||||||||< net.sourceforge.jocular.properties.EnumProperty,m_enum >
▼	8
SC	|||||||||< net.sourceforge.jocular.properties.BooleanProperty,m_value >
▼	3
SC	|||||||||< net.sourceforge.jocular.graphs.GraphAxis,log >
SC	|||||||||< net.sourceforge.jocular.photons.WranglerEvent,m_type >
SC	|||||||||< net.sourceforge.jocular.photons.Photon,m_photonSource >
SC	|||||||||< net.sourceforge.jocular.project.ProjectUpdatedEvent,m_type >
▼	2
SC	|||||||||< net.sourceforge.jocular.math.Complex,m_imag >
SC	|||||||||< net.sourceforge.jocular.math.Vector3D,x >
SC	|||||||||< net.sourceforge.jocular.splines.SplinePointTableModel,m_object >
SC	|||||||||< net.sourceforge.jocular.gui.panel3d.OpticsSceneGraphComponent,m_isTransparent >
SC	|||||||||< net.sourceforge.jocular.undo.PropertyEdit,m_owner >
SC	|||||||||< net.sourceforge.jocular.project.OpticsProject,m_dirtyFlag >
SC	|||||||||< net.sourceforge.jocular.graphs.GraphAxis,gridLineSpacing >
ST	|||||||||< net.sourceforge.jocular.photons.PhotonTrajectory,i >
ST	|||||||||< net.sourceforge.jocular.objects.AbstractOpticsObject,m_selected >
ST	|||||||||< net.sourceforge.jocular.math.SimpleMinimumSolver,m_running >
ST	|||||||||< net.sourceforge.jocular.math.MultiMinimumSolver,m_running >

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



```
    protected double gridLineSpacing;
    public GraphAxis(double min, double max, boolean log, double gridLineSpacing){
        this.min = min;
        this.max = max;
        this.log = log;
        if(log && (min == 0 || max == 0)){
            throw(new RuntimeException("Cannot use zero limit with log scale."));
        }
        this.gridLineSpacing = gridLineSpacing;
    }
    public double scale(double n){
        double result = n;
        if(log){
            result = Math.log10(result);// should now be in the range -6 to 0
            result -= Math.log10(min);
            result /= Math.log10(max) - Math.log10(min);
        } else {
            result -= min;
            result /= max - min;
        }
        result *= .8;
        result += .1;

        if(result
            resul
        } else if
            resul
        }
        return re
    }
    public double
        double re
        result -=
        result /= .8;
        if(log){
            result *= Math.log10(max) - Math.log10(min);
            result += Math.log10(min);
            result = Math.pow(10,result);
        } else {
            result *= max - min;
            result += min;
        }
        return result;
    }
    public ArrayList<Double> getGridLines(){
        if(gridLineSpacing == 0){
            return null;
        }
        ArrayList<Double> result = new ArrayList<Double>();

        if(log){

            double n = Math.ceil(Math.log10(min));
```

SCST View

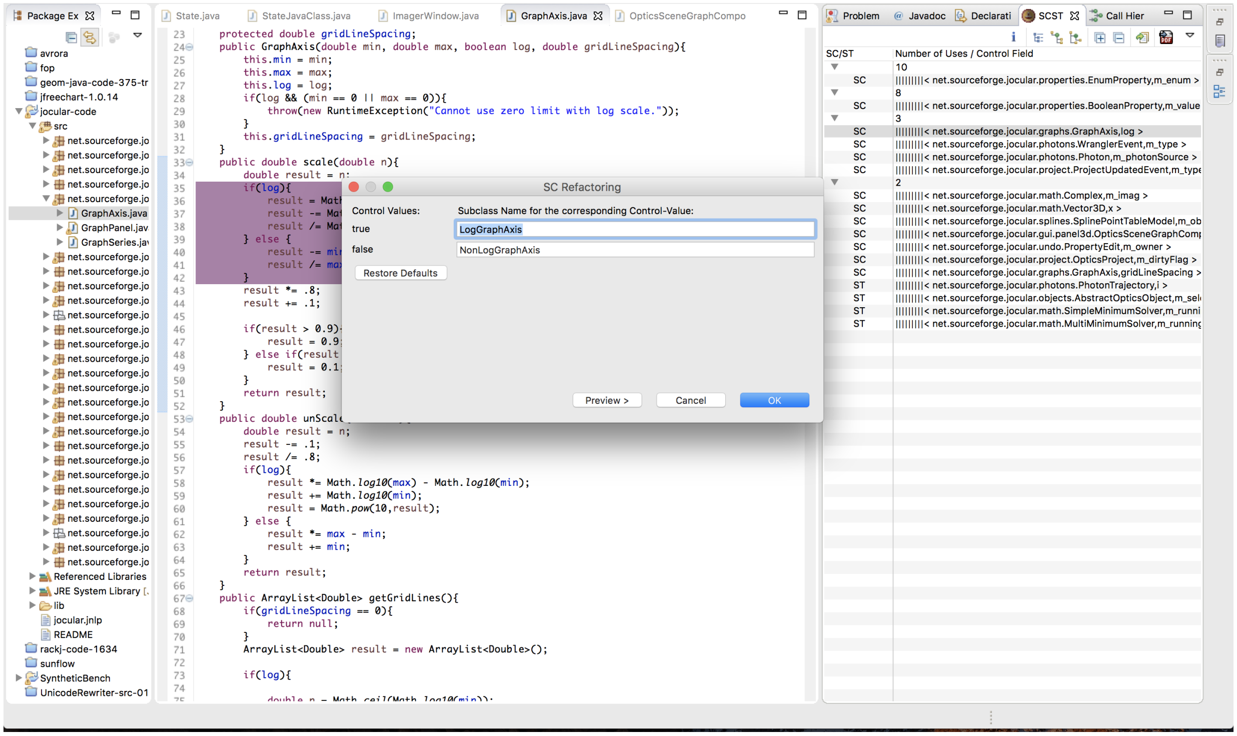
1 conditional
in class: net.sourceforge.jocular.graphs.GraphAxis
at line number: 35
PRUNED {}

SC/ST	Number of Uses / Control Field
▼	10
SC	|||||||||< net.sourceforge.jocular.properties.EnumProperty,m_enum >
▼	8
SC	|||||||||< net.sourceforge.jocular.properties.BooleanProperty,m_value
▼	3
SC	|||||||||< net.sourceforge.jocular.graphs.GraphAxis,log >
SC	|||||||||< net.sourceforge.jocular.photons.WranglerEvent,m_type >
SC	|||||||||< net.sourceforge.jocular.photons.Photon,m_photonSource >
SC	|||||||||< net.sourceforge.jocular.project.ProjectUpdatedEvent,m_type
▼	2
SC	|||||||||< net.sourceforge.jocular.math.Complex,m_imag >
SC	|||||||||< net.sourceforge.jocular.math.Vector3D,x >
SC	|||||||||< net.sourceforge.jocular.splines.SplinePointTableModel,m_ob
SC	|||||||||< net.sourceforge.jocular.gui.panel3d.OpticsSceneGraphComp
SC	|||||||||< net.sourceforge.jocular.undo.PropertyEdit,m_owner >
SC	|||||||||< net.sourceforge.jocular.project.OpticsProject,m_dirtyFlag >
SC	|||||||||< net.sourceforge.jocular.graphs.GraphAxis,gridLineSpacing >
ST	|||||||||< net.sourceforge.jocular.photons.PhotonTrajectory,i >
ST	|||||||||< net.sourceforge.jocular.objects.AbstractOpticsObject,m_sel
ST	|||||||||< net.sourceforge.jocular.math.SimpleMinimumSolver,m_runni
ST	|||||||||< net.sourceforge.jocular.math.MultiMinimumSolver,m_running

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.



```
    protected double gridLineSpacing;
    public GraphAxis(double min, double max, boolean log, double gridLineSpacing){
        this.min = min;
        this.max = max;
        this.log = log;
        if(log && (min == 0 || max == 0)){
            throw(new RuntimeException("Cannot use zero limit with log scale."));
        }
        this.gridLineSpacing = gridLineSpacing;
    }
    public double scale(double n){
        double result = n;
        if(log){
            result = Math
            result -= Ma
            result /= Ma
        } else {
            result -= mi
            result /= ma
        }
        result *= .8;
        result += .1;

        if(result > 0.9)
            result = 0.9
        } else if(result
            result = 0.1
        }
        return result;
    }
    public double unScale
        double result = n;
        result -= .1;
        result /= .8;
        if(log){
            result *= Math.log10(max) - Math.log10(min);
            result += Math.log10(min);
            result = Math.pow(10,result);
        } else {
            result *= max - min;
            result += min;
        }
        return result;
    }
    public ArrayList<Double> getGridLines(){
        if(gridLineSpacing == 0){
            return null;
        }
        ArrayList<Double> result = new ArrayList<Double>();

        if(log){

            double n = Math.ceil(Math.log10(min));
```

SC Refactoring

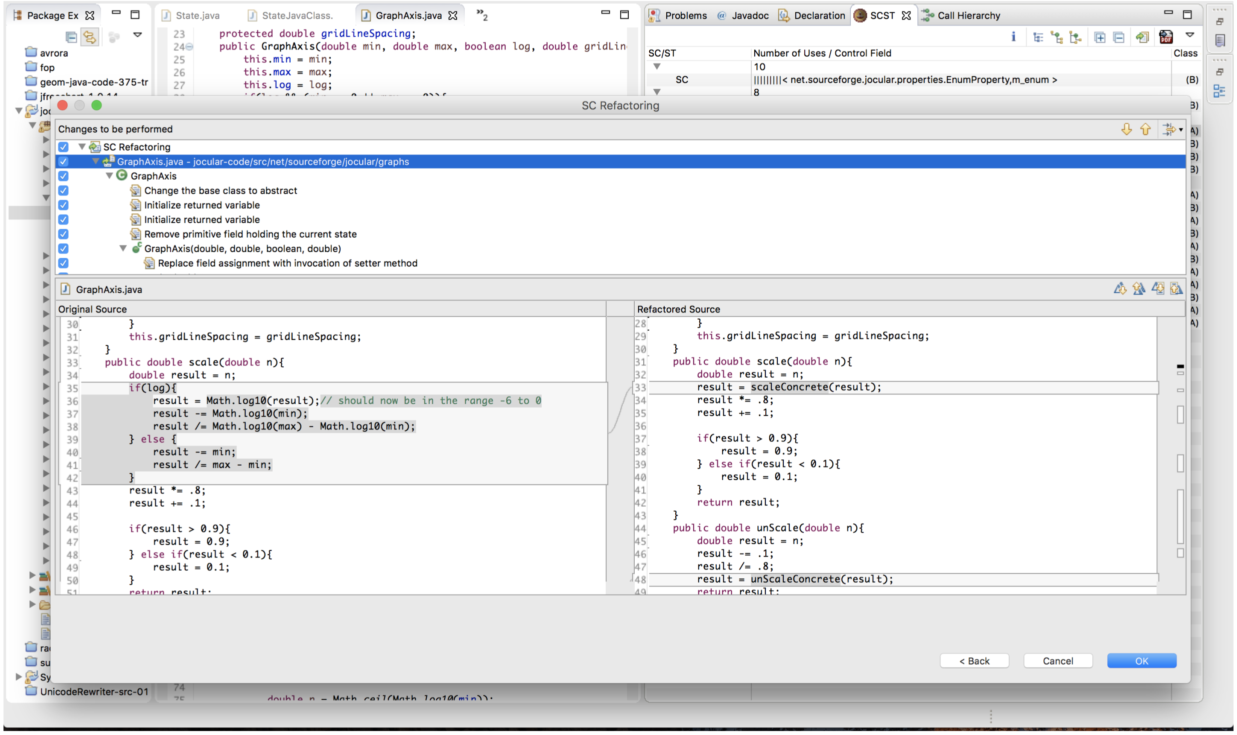
Control Values:	Subclass Name for the corresponding Control-Value:
true	LogGraphAxis
false	NonLogGraphAxis

SC/ST	Number of Uses / Control Field
▼	10
SC	|||||||||< net.sourceforge.jocular.properties.EnumProperty,m_enum >
▼	8
SC	|||||||||< net.sourceforge.jocular.properties.BooleanProperty,m_value
▼	3
SC	|||||||||< net.sourceforge.jocular.graphs.GraphAxis,log >
SC	|||||||||< net.sourceforge.jocular.photons.WranglerEvent,m_type >
SC	|||||||||< net.sourceforge.jocular.photons.Photon,m_photonSource >
SC	|||||||||< net.sourceforge.jocular.project.ProjectUpdatedEvent,m_type
▼	2
SC	|||||||||< net.sourceforge.jocular.math.Complex,m_imag >
SC	|||||||||< net.sourceforge.jocular.math.Vector3D,x >
SC	|||||||||< net.sourceforge.jocular.splines.SplinePointTableModel,m_ob
SC	|||||||||< net.sourceforge.jocular.gui.panel3d.OpticsSceneGraphComp
SC	|||||||||< net.sourceforge.jocular.undo.PropertyEdit,m_owner >
SC	|||||||||< net.sourceforge.jocular.project.OpticsProject,m_dirtyFlag >
SC	|||||||||< net.sourceforge.jocular.graphs.GraphAxis,gridLineSpacing >
ST	|||||||||< net.sourceforge.jocular.photons.PhotonTrajectory,i >
ST	|||||||||< net.sourceforge.jocular.objects.AbstractOpticsObject,m_seli
ST	|||||||||< net.sourceforge.jocular.math.SimpleMinimumSolver,m_runni
ST	|||||||||< net.sourceforge.jocular.math.MultiMinimumSolver,m_running

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.



Changes to be performed

- ☑ SC Refactoring
  - ☑ GraphAxis.java - jocular-code/src/net/sourceforge/jocular/graphs
    - ☑ GraphAxis
      - ☑ Change the base class to abstract
      - ☑ Initialize returned variable
      - ☑ Initialize returned variable
      - ☑ Remove primitive field holding the current state
      - ☑ GraphAxis(double, double, boolean, double)
        - ☑ Replace field assignment with invocation of setter method

GraphAxis.java

Original Source

```java
        }
        this.gridLineSpacing = gridLineSpacing;
    }
    public double scale(double n){
        double result = n;
        if(log){
            result = Math.log10(result);// should now be in the range -6 to 0
            result -= Math.log10(min);
            result /= Math.log10(max) - Math.log10(min);
        } else {
            result -= min;
            result /= max - min;
        }
        result *= .8;
        result += .1;

        if(result > 0.9){
            result = 0.9;
        } else if(result < 0.1){
            result = 0.1;
        }
        return result;
```

Refactored Source

```java
        }
        this.gridLineSpacing = gridLineSpacing;
    }
    public double scale(double n){
        double result = n;
        result = scaleConcrete(result);
        result *= .8;
        result += .1;

        if(result > 0.9){
            result = 0.9;
        } else if(result < 0.1){
            result = 0.1;
        }
        return result;
    }
    public double unScale(double n){
        double result = n;
        result -= .1;
        result /= .8;
        result = unScaleConcrete(result);
        return result;
```

SC/ST | Number of Uses / Control Field | Class
10
SC | ||||||||< net.sourceforge.jocular.properties.EnumProperty,m_enum > | (B)
8

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.