```java
package lehmansrulesgraphsdisplay;

import java.util.ArrayList;

public class RefactoredGraphVisualizer {
    private History h;

    public RefactoredGraphVisualizer(History h){
        this.h = h;
    }

    public ChartPanel []firstRuleGraphGenerator(){
        ChartPanel []panels= new ChartPanel[2];
        DefaultCategoryDataset objDataset = new DefaultCategoryDataset();
        DefaultCategoryDataset objDataset2 = new DefaultCategoryDataset();
        ArrayList<VersionInfo> versionlist = h.getVersions();

        constructChangesDataset(objDataset, versionlist);
        panels[0] = createBarChartPanel("Version per Year Bar Chart", "Version ID",
"changes", objDataset);

        constructVersionsDataset(versionlist, objDataset2);
        panels[1] = createBarChartPanel("Version per Year Bar Chart", "Version ID",
"Versions", objDataset2);

        return panels;
    }

    private void constructChangesDataset(DefaultCategoryDataset objDataset,
            ArrayList<VersionInfo> versionlist) {
        int i;
        for(i=0; i<versionlist.size(); i++){
            double dops = versionlist.get(i).getopsChanges().getNumOfChanges();
            String xaxis = versionlist.get(i).getID();
            objDataset.setValue(dops,"Operations", xaxis);
        }
        for(i=0; i<versionlist.size(); i++){
            double dstructs = versionlist.get(i).getstructsChanges().getNumOfChanges();
            String xaxis = versionlist.get(i).getID();
            objDataset.setValue(dstructs,"Structs", xaxis);
        }
    }

    private void constructVersionsDataset(ArrayList<VersionInfo> versionlist,
            DefaultCategoryDataset objDataset2) {
        int [][]table;
        if(versionlist.size()<=0){
            return;
        }
        int first= versionlist.get(0).yearOfDate();
        int last = versionlist.get(versionlist.size()-1).yearOfDate();
        int size = last-first+1;
        table = new int[size][2];
        int i;
        for(i=0; i<size; i++){
            table[i][0] = first + i;
            table[i][1] = 0;
        }

        for(i=0; i<versionlist.size(); i++){
            int year = versionlist.get(i).yearOfDate();
```

```
            for(int j=0; j<size; j++){
                if(table[j][0]==year){
                    table[j][1]++;
                    break;
                }
            }
        }

        for(i=0; i<size; i++){
            String xaxis = "'"+(table[i][0]-2000);
            objDataset2.setValue(table[i][1],"Year Versions", xaxis);

        }
    }

    public ChartPanel[] secondRuleGraphGenerator()
    {
        ChartPanel []panels= new ChartPanel[2];
        DefaultCategoryDataset objDataset = new DefaultCategoryDataset();
        ArrayList<VersionInfo> versionlist = h.getVersions();
        int i;

        constructComplexitiesDataset(objDataset, versionlist);
        panels[0] = createLineChartPanel("Complexity Line Chart", "Version
ID","Complexity", objDataset);

        DefaultCategoryDataset objDataset2 = new DefaultCategoryDataset();
        constructMaintenanceActivitiesDataset(versionlist, objDataset2);
        panels[1] = createLineChartPanel("Maintainance Bar Chart", "Version ID",
"Maintainance", objDataset2);

        return panels;
    }

    private void constructComplexitiesDataset(
            DefaultCategoryDataset objDataset,
            ArrayList<VersionInfo> versionlist) {
        int i;
        for(i=0; i<versionlist.size(); i++){
            double dops = versionlist.get(i).getopsMetrics().getComplexity();
            String xaxis = versionlist.get(i).getID();
            objDataset.setValue(dops,"Operations", xaxis);
        }
        for(i=0; i<versionlist.size(); i++){
            double dstructs = versionlist.get(i).getstructsMetrics().getComplexity();
            String xaxis = versionlist.get(i).getID();
            objDataset.setValue(dstructs,"Structs", xaxis);
        }
    }

    private void constructMaintenanceActivitiesDataset(
            ArrayList<VersionInfo> versionlist,
            DefaultCategoryDataset objDataset2) {
        int i;
        for(i=0; i<versionlist.size(); i++){
            double dops = versionlist.get(i).getopsChanges().getMaintenanceActivities();
            String xaxis = versionlist.get(i).getID();
            objDataset2.setValue(dops,"Operations", xaxis);
        }
        for(i=0; i<versionlist.size(); i++){
            double dstructs =
```

```
   versionlist.get(i).getstructsChanges().getMaintenanceActivities();
132             String xaxis = versionlist.get(i).getID();
133             objDataset2.setValue(dstructs,"Structs", xaxis);
134         }
135     }
136 
137     public ChartPanel[] thirdRuleGraphGenerator()
138     {
139         ChartPanel []panels= new ChartPanel[2];
140         DefaultCategoryDataset objDataset = new DefaultCategoryDataset();
141         DefaultCategoryDataset objDataset2 = new DefaultCategoryDataset();
142         ArrayList<VersionInfo> versionlist = h.getVersions();
143 
144         constructOperationsGrowthRateDataset(objDataset, versionlist);
145         panels[0] = createLineChartPanel("Operations Growth Rate Line Chart", "Version
   ID","Growth Rate", objDataset);
146 
147         constructStructGrowthRateDataset(versionlist, objDataset2);
148         panels[1] = createLineChartPanel("Structs Growth Rate Line Chart", "Version
   ID","Growth Rate", objDataset2);
149 
150         return panels;
151     }
152 
153     private void constructStructGrowthRateDataset(
154             ArrayList<VersionInfo> versionlist,
155             DefaultCategoryDataset objDataset2) {
156 
157         for(int i=0; i<versionlist.size(); i++){
158             double dstructs = versionlist.get(i).getstructsMetrics().getGrowthRate();
159             String xaxis = versionlist.get(i).getID();
160             objDataset2.setValue(dstructs,"Structs", xaxis);
161         }
162     }
163 
164     private void constructOperationsGrowthRateDataset(DefaultCategoryDataset objDataset,
165             ArrayList<VersionInfo> versionlist) {
166 
167         for(int i=0; i<versionlist.size(); i++){
168             double dops = versionlist.get(i).getopsMetrics().getGrowthRate();
169             String xaxis = versionlist.get(i).getID();
170             objDataset.setValue(dops,"Operations", xaxis);
171         }
172     }
173 
174     public ChartPanel[] fourthRuleGraphGenerator()
175     {
176         ChartPanel []panels= new ChartPanel[2];
177         DefaultCategoryDataset objDataset = new DefaultCategoryDataset();
178         DefaultCategoryDataset objDataset2 = new DefaultCategoryDataset();
179         ArrayList<VersionInfo> versionlist = h.getVersions();
180 
181         constructOperationsWorkRateDataset(objDataset, versionlist);
182         panels[0] = createLineChartPanel("Operations Work Rate Line Chart", "Version
   ID","Work Rate", objDataset);
183 
184         constructStructsWorkRateDataset(versionlist, objDataset2);
185         panels[1] = createLineChartPanel("Structs Work Rate Line Chart", "Version ID","Work
   Rate", objDataset2);
186 
187         return panels;
```

```
188     }
189 
190     private void constructStructsWorkRateDataset(
191             ArrayList<VersionInfo> versionlist,
192             DefaultCategoryDataset objDataset2) {
193 
194         for(int i=0; i<versionlist.size(); i++){
195             double dstructs = versionlist.get(i).getstructsMetrics().getWorkRate();
196             String xaxis = versionlist.get(i).getID();
197             objDataset2.setValue(dstructs,"Structs", xaxis);
198         }
199     }
200 
201     private void constructOperationsWorkRateDataset(
202             DefaultCategoryDataset objDataset,
203             ArrayList<VersionInfo> versionlist) {
204 
205         for(int i=0; i<versionlist.size(); i++){
206             double dops = versionlist.get(i).getopsMetrics().getWorkRate();
207             String xaxis = versionlist.get(i).getID();
208             objDataset.setValue(dops,"Operations", xaxis);
209         }
210     }
211 
212     public ChartPanel[] fifthRuleGraphGenerator()
213     {
214         return thirdRuleGraphGenerator();
215     }
216 
217     public ChartPanel[] sixthRuleGraphGenerator()
218     {
219 
220         ChartPanel []panels= new ChartPanel[2];
221         DefaultCategoryDataset objDataset = new DefaultCategoryDataset();
222         DefaultCategoryDataset objDataset2 = new DefaultCategoryDataset();
223         ArrayList<VersionInfo> versionlist = h.getVersions();
224 
225         constructOperationsSizeDataset(objDataset, versionlist);
226         panels[0] = createLineChartPanel("Operations Size Line Chart", "Version ID","Work
    Rate", objDataset);
227 
228         constructStructsSizeDataset(objDataset2, versionlist);
229         panels[1] = createLineChartPanel("Structs Size Line Chart", "Version ID","Work
    Rate", objDataset2);
230 
231         return panels;
232     }
233 
234     private void constructStructsSizeDataset(
235             DefaultCategoryDataset objDataset2,
236             ArrayList<VersionInfo> versionlist) {
237         for(int i=0; i<versionlist.size(); i++){
238             double dstructs = versionlist.get(i).getstructsMetrics().getNumber();
239             String xaxis = versionlist.get(i).getID();
240             objDataset2.setValue(dstructs,"Structs", xaxis);
241         }
242     }
243 
244     private void constructOperationsSizeDataset(
245             DefaultCategoryDataset objDataset,
246             ArrayList<VersionInfo> versionlist) {
```

```
247         for(int i=0; i<versionlist.size(); i++){
248             double dops = versionlist.get(i).getopsMetrics().getNumber();
249             String xaxis = versionlist.get(i).getID();
250             objDataset.setValue(dops,"Operations", xaxis);
251         }
252     }
253 
254     public ChartPanel[] seventhRuleGraphGenerator()
255     {
256         return null;
257     }
258 
259     public ChartPanel[] eighthRuleGraphGenerator()
260     {
261         return null;
262     }
263 
264     private ChartPanel createBarChartPanel(String title, String xAxisTitle, String
    yAxisTitle, DefaultCategoryDataset objDataset2){
265         JFreeChart objChart2 = ChartFactory.createBarChart(
266                 title,
267                 xAxisTitle,
268                 yAxisTitle,
269                 objDataset2,
270                 PlotOrientation.VERTICAL,
271                 true,
272                 true,
273                 false
274             );
275         return new ChartPanel(objChart2);
276     }
277 
278     private ChartPanel createLineChartPanel(String title, String xAxisTitle, String
    yAxisTitle, DefaultCategoryDataset objDataset2){
279         JFreeChart objChart2 = ChartFactory.createLineChart(
280                 title,
281                 xAxisTitle,
282                 yAxisTitle,
283                 objDataset2,
284                 PlotOrientation.VERTICAL,
285                 true,
286                 true,
287                 false
288             );
289         return new ChartPanel(objChart2);
290     }
291 }
292 
```