

RefactoredGraphVisualizer2.java

```

1 package lehmanrulesgraphsdisplay;
2
3 import java.util.ArrayList;
4
17
18 public class RefactoredGraphVisualizer2 {
19     private History h;
20
21     public RefactoredGraphVisualizer2(History h){
22         this.h = h;
23     }
24
25     public ChartPanel []firstRuleGraphGenerator(){
26         ChartPanel []panels= new ChartPanel[2];
27         DefaultCategoryDataset objDataset = new DefaultCategoryDataset();
28         DefaultCategoryDataset objDataset2 = new DefaultCategoryDataset();
29         ArrayList<VersionInfo> versionlist = h.getVersions();
30
31         constructChangesDataset(objDataset, versionlist);
32         panels[0] = createBarChartPanel(new ChartTitles("Version per Year Bar Chart",
33 "Version ID", "changes"), objDataset);
34
35         constructVersionsDataset(versionlist, objDataset2);
36         panels[1] = createBarChartPanel(new ChartTitles("Version per Year Bar Chart",
37 "Version ID", "Versions"), objDataset2);
38
39         return panels;
40     }
41
42     private void constructChangesDataset(DefaultCategoryDataset objDataset,
43 ArrayList<VersionInfo> versionlist) {
44         int i;
45         for(i=0; i<versionlist.size(); i++){
46             double dops = versionlist.get(i).getopsChanges().getNumOfChanges();
47             String xaxis = versionlist.get(i).getID();
48             objDataset.setValue(dops,"Operations", xaxis);
49         }
50         for(i=0; i<versionlist.size(); i++){
51             double dstructs = versionlist.get(i).getstructsChanges().getNumOfChanges();
52             String xaxis = versionlist.get(i).getID();
53             objDataset.setValue(dstructs,"Structs", xaxis);
54         }
55     }
56
57     private void constructVersionsDataset(ArrayList<VersionInfo> versionlist,
58 DefaultCategoryDataset objDataset2) {
59         int [][]table;
60         if(versionlist.size()<=0){
61             return;
62         }
63         int first= versionlist.get(0).yearOfDate();
64         int last = versionlist.get(versionlist.size()-1).yearOfDate();
65         int size = last-first+1;
66         table = new int[size][2];
67         int i;
68         for(i=0; i<size; i++){
69             table[i][0] = first + i;
70             table[i][1] = 0;
71         }
72
73         for(i=0; i<versionlist.size(); i++){
74             int year = versionlist.get(i).yearOfDate();

```

RefactoredGraphVisualizer2.java

```

73         for(int j=0; j<size; j++){
74             if(table[j][0]==year){
75                 table[j][1]++;
76                 break;
77             }
78         }
79     }
80
81     for(i=0; i<size; i++){
82         String xaxis = ""+(table[i][0]-2000);
83         objDataset2.setValue(table[i][1],"Year Versions", xaxis);
84     }
85 }
86
87
88 public ChartPanel[] secondRuleGraphGenerator()
89 {
90     ChartPanel []panels= new ChartPanel[2];
91     DefaultCategoryDataset objDataset = new DefaultCategoryDataset();
92     ArrayList<VersionInfo> versionlist = h.getVersions();
93     int i;
94
95     constructComplexitiesDataset(objDataset, versionlist);
96     panels[0] = createLineChartPanel(new ChartTitles("Complexity Line Chart", "Version
ID", "Complexity"), objDataset);
97
98     DefaultCategoryDataset objDataset2 = new DefaultCategoryDataset();
99     constructMaintenanceActivitiesDataset(versionlist, objDataset2);
100    panels[1] = createLineChartPanel(new ChartTitles("Maintainance Bar Chart", "Version
ID", "Maintainance"), objDataset2);
101
102    return panels;
103 }
104
105 private void constructComplexitiesDataset(
106     DefaultCategoryDataset objDataset,
107     ArrayList<VersionInfo> versionlist) {
108     int i;
109     for(i=0; i<versionlist.size(); i++){
110         double dops = versionlist.get(i).getopsMetrics().getComplexity();
111         String xaxis = versionlist.get(i).getID();
112         objDataset.setValue(dops,"Operations", xaxis);
113     }
114     for(i=0; i<versionlist.size(); i++){
115         double dstructs = versionlist.get(i).getstructsMetrics().getComplexity();
116         String xaxis = versionlist.get(i).getID();
117         objDataset.setValue(dstructs,"Structs", xaxis);
118     }
119 }
120
121 private void constructMaintenanceActivitiesDataset(
122     ArrayList<VersionInfo> versionlist,
123     DefaultCategoryDataset objDataset2) {
124     int i;
125     for(i=0; i<versionlist.size(); i++){
126         double dops = versionlist.get(i).getopsChanges().getMaintenanceActivities();
127         String xaxis = versionlist.get(i).getID();
128         objDataset2.setValue(dops,"Operations", xaxis);
129     }
130     for(i=0; i<versionlist.size(); i++){
131         double dstructs =

```

RefactoredGraphVisualizer2.java

```

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(new ChartTitles("Operations Growth Rate Line
Chart", "Version ID", "Growth Rate"), objDataset);
146
147     constructStructGrowthRateDataset(versionlist, objDataset2);
148     panels[1] = createLineChartPanel(new ChartTitles("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(new ChartTitles("Operations Work Rate Line Chart",
"Version ID", "Work Rate"), objDataset);
183
184     constructStructsWorkRateDataset(versionlist, objDataset2);
185     panels[1] = createLineChartPanel(new ChartTitles("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(new ChartTitles("Operations Size Line Chart",
227 "Version ID", "Work Rate"), objDataset);
228
229         constructStructsSizeDataset(objDataset2, versionlist);
230         panels[1] = createLineChartPanel(new ChartTitles("Structs Size Line Chart",
231 "Version ID", "Work Rate"), objDataset2);
232
233         return panels;
234     }
235
236     private void constructStructsSizeDataset(
237         DefaultCategoryDataset objDataset2,
238         ArrayList<VersionInfo> versionlist) {
239
240         for(int i=0; i<versionlist.size(); i++){
241             double dstructs = versionlist.get(i).getstructsMetrics().getNumber();
242             String xaxis = versionlist.get(i).getID();
243             objDataset2.setValue(dstructs, "Structs", xaxis);
244         }
245     }
246
247     private void constructOperationsSizeDataset(
248         DefaultCategoryDataset objDataset,
249         ArrayList<VersionInfo> versionlist) {

```

RefactoredGraphVisualizer2.java

```

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(ChartTitles parameterObject,
DefaultCategoryDataset objDataset2){
265     JFreeChart objChart2 = ChartFactory.createBarChart(
266         parameterObject.getTitle(),
267         parameterObject.getxAxisTitle(),
268         parameterObject.getyAxisTitle(),
269         objDataset2,
270         PlotOrientation.VERTICAL,
271         true,
272         true,
273         false
274     );
275     return new ChartPanel(objChart2);
276 }
277
278 private ChartPanel createLineChartPanel(ChartTitles parameterObject,
DefaultCategoryDataset objDataset2){
279     JFreeChart objChart2 = ChartFactory.createLineChart(
280         parameterObject.getTitle(),
281         parameterObject.getxAxisTitle(),
282         parameterObject.getyAxisTitle(),
283         objDataset2,
284         PlotOrientation.VERTICAL,
285         true,
286         true,
287         false
288     );
289     return new ChartPanel(objChart2);
290 }
291 }
292

```