

# RefactoredGraphVisualizerBeforeMovingStuff.java

```

1 package lehmanrulesgraphsdisplay;
2
3 import java.util.ArrayList;
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18 public class RefactoredGraphVisualizerBeforeMovingStuff {
19     public History h;
20
21     public RefactoredGraphVisualizerBeforeMovingStuff(History h){
22         this.h = h;
23     }
24
25     public ChartPanel []firstRuleGraphGenerator(){
26         ChartPanel []panels= new ChartPanel[2];
27
28         DefaultCategoryDataset objDataset = h.createChangesDataset();
29         panels[0] = createBarChartPanel("Version per Year Bar Chart", "Version ID",
30 "changes", objDataset);
31
32         DefaultCategoryDataset objDataset2 = h.createVersionsDataset();
33         panels[1] = createBarChartPanel("Version per Year Bar Chart", "Version ID",
34 "Versions", objDataset2);
35
36         return panels;
37     }
38
39     private ChartPanel createBarChartPanel(String title, String xAxisTitle, String
40 yAxisTitle, DefaultCategoryDataset objDataset2){
41         JFreeChart objChart2 = ChartFactory.createBarChart(
42             title,
43             xAxisTitle,
44             yAxisTitle,
45             objDataset2,
46             PlotOrientation.VERTICAL,
47             true,
48             true,
49             false
50 );
51         return new ChartPanel(objChart2);
52     }
53
54     public ChartPanel[] secondRuleGraphGenerator()
55     {
56         ChartPanel []panels= new ChartPanel[2];
57         DefaultCategoryDataset objDataset = new DefaultCategoryDataset();
58         ArrayList<VersionInfo> versionlist = h.getVersions();
59         int i;
60         for(i=0; i<versionlist.size(); i++){
61             double dops = versionlist.get(i).getopsMetrics().getComplexity();
62             String xaxis = versionlist.get(i).getID();
63             objDataset.setValue(dops,"Operations", xaxis);
64         }
65         for(i=0; i<versionlist.size(); i++){
66             double dstructs = versionlist.get(i).getstructsMetrics().getComplexity();
67             String xaxis = versionlist.get(i).getID();
68             objDataset.setValue(dstructs,"Structs", xaxis);
69         }
70         JFreeChart objChart = ChartFactory.createLineChart(
71             "Complexity Line Chart",
72             "Version ID",
73             "Complexity",
74

```

# RefactoredGraphVisualizerBeforeMovingStuff.java

```

73         objDataset,
74         PlotOrientation.VERTICAL,
75         true,
76         true,
77         false
78     );
79     panels[0] = new ChartPanel(objChart);
80     DefaultCategoryDataset objDataset2 = new DefaultCategoryDataset();
81     for(i=0; i<versionlist.size(); i++){
82         double dops = versionlist.get(i).getopsChanges().getMaintenanceActivities();
83         String xaxis = versionlist.get(i).getID();
84         objDataset2.setValue(dops,"Operations", xaxis);
85     }
86     for(i=0; i<versionlist.size(); i++){
87         double dstructs =
versionlist.get(i).getstructsChanges().getMaintenanceActivities();
88         String xaxis = versionlist.get(i).getID();
89         objDataset2.setValue(dstructs,"Structs", xaxis);
90     }
91     JFreeChart objChart2 = ChartFactory.createBarChart(
92         "Maintenance Bar Chart",
93         "Version ID",
94         "Maintenance",
95         objDataset2,
96         PlotOrientation.VERTICAL,
97         true,
98         true,
99         false
100    );
101    panels[1] = new ChartPanel(objChart2);
102    return panels;
103 }
104
105
106
107 public ChartPanel[] thirdRuleGraphGenerator()
108 {
109     ChartPanel []panels= new ChartPanel[2];
110     DefaultCategoryDataset objDataset = new DefaultCategoryDataset();
111     ArrayList<VersionInfo> versionlist = h.getVersions();
112     int i;
113     for(i=0; i<versionlist.size(); i++){
114         double dops = versionlist.get(i).getopsMetrics().getGrowthRate();
115         String xaxis = versionlist.get(i).getID();
116         objDataset.setValue(dops,"Operations", xaxis);
117     }
118     JFreeChart objChart = ChartFactory.createLineChart(
119         "Growth Rate Line Chart",
120         "Version ID",
121         "Growth Rate",
122         objDataset,
123         PlotOrientation.VERTICAL,
124         true,
125         true,
126         false
127     );
128     panels[0] = new ChartPanel(objChart);
129     DefaultCategoryDataset objDataset2 = new DefaultCategoryDataset();
130     for(i=0; i<versionlist.size(); i++){
131         double dstructs = versionlist.get(i).getstructsMetrics().getGrowthRate();
132         String xaxis = versionlist.get(i).getID();
133         objDataset2.setValue(dstructs,"Structs", xaxis);

```

# RefactoredGraphVisualizerBeforeMovingStuff.java

```

134     }
135     JFreeChart objChart2 = ChartFactory.createLineChart(
136         "Growth Rate Bar Chart",
137         "Version ID",
138         "Growth Rate",
139         objDataset2,
140         PlotOrientation.VERTICAL,
141         true,
142         true,
143         false
144     );
145     panels[1] = new ChartPanel(objChart2);
146
147     return panels;
148 }
149
150 public ChartPanel[] fourthRuleGraphGenerator()
151 {
152     ChartPanel []panels= new ChartPanel[2];
153     DefaultCategoryDataset objDataset = new DefaultCategoryDataset();
154     ArrayList<VersionInfo> versionlist = h.getVersions();
155     int i;
156     for(i=0; i<versionlist.size(); i++){
157         double dops = versionlist.get(i).getopsMetrics().getWorkRate();
158         String xaxis = versionlist.get(i).getID();
159         objDataset.setValue(dops,"Operations", xaxis);
160     }
161     JFreeChart objChart = ChartFactory.createLineChart(
162         "Work Rate Line Chart",
163         "Version ID",
164         "Work Rate",
165         objDataset,
166         PlotOrientation.VERTICAL,
167         true,
168         true,
169         false
170     );
171     panels[0] = new ChartPanel(objChart);
172     DefaultCategoryDataset objDataset2 = new DefaultCategoryDataset();
173     for(i=0; i<versionlist.size(); i++){
174         double dstructs = versionlist.get(i).getstructsMetrics().getWorkRate();
175         String xaxis = versionlist.get(i).getID();
176         objDataset2.setValue(dstructs,"Structs", xaxis);
177     }
178     JFreeChart objChart2 = ChartFactory.createLineChart(
179         "Work Rate Bar Chart",
180         "Version ID",
181         "Work Rate",
182         objDataset2,
183         PlotOrientation.VERTICAL,
184         true,
185         true,
186         false
187     );
188     panels[1] = new ChartPanel(objChart2);
189     return panels;
190 }
191
192 public ChartPanel[] fifthRuleGraphGenerator()
193 {
194     return thirdRuleGraphGenerator();
195 }

```

```

196
197 public ChartPanel[] sixthRuleGraphGenerator()
198 {
199
200     ChartPanel []panels= new ChartPanel[2];
201     DefaultCategoryDataset objDataset = new DefaultCategoryDataset();
202     ArrayList<VersionInfo> versionlist = h.getVersions();
203     int i;
204     for(i=0; i<versionlist.size(); i++){
205         double dops = versionlist.get(i).getopsMetrics().getNumber();
206         String xaxis = versionlist.get(i).getID();
207         objDataset.setValue(dops,"Operations", xaxis);
208     }
209     JFreeChart objChart = ChartFactory.createLineChart(
210         "Work Rate Line Chart",
211         "Version ID",
212         "Work Rate",
213         objDataset,
214         PlotOrientation.VERTICAL,
215         true,
216         true,
217         false
218     );
219     panels[0] = new ChartPanel(objChart);
220     DefaultCategoryDataset objDataset2 = new DefaultCategoryDataset();
221     for(i=0; i<versionlist.size(); i++){
222         double dstructs = versionlist.get(i).getstructsMetrics().getNumber();
223         String xaxis = versionlist.get(i).getID();
224         objDataset2.setValue(dstructs,"Structs", xaxis);
225     }
226     JFreeChart objChart2 = ChartFactory.createLineChart(
227         "Work Rate Bar Chart",
228         "Version ID",
229         "Work Rate",
230         objDataset2,
231         PlotOrientation.VERTICAL,
232         true,
233         true,
234         false
235     );
236     panels[1] = new ChartPanel(objChart2);
237
238     return panels;
239 }
240
241 public ChartPanel[] seventhRuleGraphGenerator()
242 {
243     return null;
244 }
245
246 public ChartPanel[] eighthRuleGraphGenerator()
247 {
248     return null;
249 }
250 }
251

```