

```

1 package manageData;
2
3
4
5 import org.jfree.chart.*;
12
13 public class Laws {
14     History history;
15     Report report;
16     public Laws(History history, Report report){
17         this.history=history;
18         this.report=report;
19     }
20     public void evaluateLaw1(){
21         DefaultCategoryDataset objDataset = new DefaultCategoryDataset();
22         for(int i=0;i<this.history.idCounter;i++){
23             objDataset.setValue(this.history.returnOperationChanges(i),"Q1",Integer.toString
24                 g(i+1));
25         }
26         JFreeChart objChart = ChartFactory.createBarChart(
27             "MTurk:Operation Changes",
28             "version ID",
29             "number of changes",
30             objDataset,
31             PlotOrientation.VERTICAL,
32             true,
33             true,
34             false
35         );
36         ChartFrame frame = new ChartFrame("MTurk:Operation Changes", objChart);
37         frame.pack();
38         frame.setVisible(true);
39         objDataset = new DefaultCategoryDataset();
40         for(int i=0;i<this.history.idCounter;i++){
41             objDataset.setValue(this.history.returnDataStructuresChanges(i),"Q1",Integer.to
42                 String(i+1));
43         }
44         objChart = ChartFactory.createBarChart(
45             "MTurk:Data Structures Changes",
46             "version ID",
47             "number of changes",
48             objDataset,
49             PlotOrientation.VERTICAL,
50             true,
51             true,
52             false
53         );
54         frame = new ChartFrame("MTurk:Data Structures Changes", objChart);
55         frame.pack();
56         frame.setVisible(true);
57         objDataset = new DefaultCategoryDataset();
58         int times[][]=this.history.retrunNumberOfOpPerYear();
59         for(int i=0;i<times.length;i++){
60             objDataset.setValue(times[i][1],"Q1",Integer.toString(times[i][0]));
61         }
62         objChart = ChartFactory.createBarChart(
63             "MTurk:versions per year",
64             "version ID",
65             "number of versions",
66             objDataset,
67             PlotOrientation.VERTICAL,

```

long

good - would have been long

long

good - would have been long

```

66         true,
67         true,
68         false
69     );
70     frame = new ChartFrame("MTurk:versions per year", objChart);
71     frame.pack();
72     frame.setVisible(true);
73 }
74
75 public void evaluateLaw2(){
76     DefaultCategoryDataset dataset = new DefaultCategoryDataset( );
77     for(int i=0;i<this.history.idCounter;i++){
78         dataset.addValue(this.history.returnComplexity0(i),"complexity",Integer.toString
79 g(i+1));
80     }
81     JFreeChart lineChart = ChartFactory.createLineChart(
82         "MTurk:Operation complexity",
83         "version ID","complexity",
84         dataset,
85         PlotOrientation.VERTICAL,
86         true,true,false
87     );
88     ChartFrame frame = new ChartFrame("MTurk:Operation complexity", lineChart);
89     frame.pack();
90     frame.setVisible(true);
91     dataset = new DefaultCategoryDataset( );
92     for(int i=0;i<this.history.idCounter;i++){
93         dataset.addValue(this.history.returnComplexityDS(i),"complexity",Integer.toString(i+1));
94     }
95     lineChart = ChartFactory.createLineChart(
96         "MTurk:DataStructure complexity",
97         "version ID","complexity",
98         dataset,
99         PlotOrientation.VERTICAL,
100         true,true,false);
101     frame = new ChartFrame("MTurk:DataStructure complexity", lineChart);
102     frame.pack();
103     frame.setVisible(true);
104     DefaultCategoryDataset objDataset = new DefaultCategoryDataset();
105     for(int i=0;i<this.history.idCounter;i++){
106         objDataset.setValue(this.history.returnMaintenance(i),"Q1",Integer.toString(i+1));
107     }
108     JFreeChart objChart = ChartFactory.createBarChart(
109         "MTurk:Maintenance Actions",
110         "version ID",
111         "number of changes",
112         objDataset,
113         PlotOrientation.VERTICAL,
114         true,
115         true,
116         false
117     );
118     frame = new ChartFrame("MTurk:Maintenance Actions", objChart);
119     frame.pack();
120     frame.setVisible(true);
121 }
122
123 public void evaluateLaw3(){
124     DefaultCategoryDataset dataset1 = new DefaultCategoryDataset( );

```

```

124     DefaultCategoryDataset dataset2 = new DefaultCategoryDataset( );
125     for(int i=0;i<this.history.idCounter;i++){
126         dataset1.addValue(this.history.returnGrowthOperations(i),"growth",Integer.toString(i+1));
127         dataset2.addValue(this.history.returnGrowthDataStructures(i),"growth",Integer.toString(i+1));
128     }
129     JFreeChart lineChart1 = ChartFactory.createLineChart(
130         "MTurk:Operation growth",
131         "version ID","growth",
132         dataset1,
133         PlotOrientation.VERTICAL,
134         true,true,false);
135     ChartFrame frame = new ChartFrame("MTurk:Operation growth", lineChart1);
136     frame.pack();
137     frame.setVisible(true);
138     JFreeChart lineChart2 = ChartFactory.createLineChart(
139         "MTurk:Data Structures growth",
140         "version ID","growth",
141         dataset2,
142         PlotOrientation.VERTICAL,
143         true,true,false);
144     frame = new ChartFrame("MTurk:Data Structures growth", lineChart2);
145     frame.pack();
146     frame.setVisible(true);
147 }
148
149 public void evaluateLaw4(){
150     DefaultCategoryDataset dataset1 = new DefaultCategoryDataset( );
151     DefaultCategoryDataset dataset2 = new DefaultCategoryDataset( );
152     for(int i=0;i<this.history.idCounter;i++){
153         dataset1.addValue(this.history.returnEmploymentRateO(i),"Work
Rate",Integer.toString(i+1));
154         dataset2.addValue(this.history.returnEmploymentRateDS(i),"Work
Rate",Integer.toString(i+1));
155     }
156     JFreeChart lineChart1 = ChartFactory.createLineChart(
157         "MTurk:Operations Work Rate",
158         "version ID","Work Rate(days)",
159         dataset1,
160         PlotOrientation.VERTICAL,
161         true,true,false);
162     ChartFrame frame = new ChartFrame("MTurk:Operations Work Rate", lineChart1);
163     frame.pack();
164     frame.setVisible(true);
165     JFreeChart lineChart2 = ChartFactory.createLineChart(
166         "MTurk:Data Structures Work Rate",
167         "version ID","Work Rate(days)",
168         dataset2,
169         PlotOrientation.VERTICAL,
170         true,true,false);
171     frame = new ChartFrame("MTurk:Data Structures Work Rate", lineChart2);
172     frame.pack();
173     frame.setVisible(true);
174 }
175
176 public void evaluateLaw5(){
177     evaluateLaw3();
178 }
179
180 public void evaluateLaw6(){

```

```

181     DefaultCategoryDataset dataset1 = new DefaultCategoryDataset( );
182     DefaultCategoryDataset dataset2 = new DefaultCategoryDataset( );
183     for(int i=0;i<this.history.idCounter;i++){
184         dataset1.addValue(this.history.returnNumberOfOperations(i),"growth",Integer.toS
tring(i+1));
185         dataset2.addValue(this.history.returnNumberOfDataStructures(i),"growth",Integer
.toString(i+1));
186     }
187     JFreeChart lineChart1 = ChartFactory.createLineChart(
188         "MTurk:Operation Number",
189         "version ID","Number of Operations",
190         dataset1,
191         PlotOrientation.VERTICAL,
192         true,true,false);
193     ChartFrame frame = new ChartFrame("MTurk:Operations Number", lineChart1);
194     frame.pack();
195     frame.setVisible(true);
196     JFreeChart lineChart2 = ChartFactory.createLineChart(
197         "MTurk:Data Structures Number",
198         "version ID","Data Structures Number",
199         dataset2,
200         PlotOrientation.VERTICAL,
201         true,true,false);
202     frame = new ChartFrame("MTurk:Data Structures Number", lineChart2);
203     frame.pack();
204     frame.setVisible(true);
205 }
206
207 public void evaluateLaw7(){
208     if(this.report.getValuation(2)==true && this.report.getValuation(6)==true){
209         this.report.InsertValuation(true,7);
210     }
211 }
212
213 public void evaluateLaw8(){
214     double E[]=new double[this.history.idCounter];
215     double S[]=new double[this.history.idCounter];
216     double s=0,Epsilon;
217     int e;
218     for(int i=0;i<E.length;i++){
219         e=
220         (this.history.returnNumberOfOperations(i)-this.history.returnNumberOfOperations(0));
221         for(int j=0;j<E.length-1;j++){
222             s=s+(1/(this.history.returnNumberOfOperations(i))^2);
223         }
224         E[i]=e/s;
225         s=0;
226         for(int i=0;i<E.length;i++){
227             s=s+E[i];
228         }
229         Epsilon=s/E.length;
230         for(int i=0;i<S.length;i++){
231             if(i==0){
232                 S[i]=this.history.InitialNumberOfOperations;
233             }
234             else{
235                 S[i]=this.history.returnNumberOfOperations(i-1)+(Epsilon/
236                 ((this.history.returnNumberOfOperations(i))*(this.history.returnNumberOfOperations(i))));
237             }

```

```

238     final XYSeries line1 = new XYSeries( "Number of Operations actual" );
239     for(int i=0;i<S.length;i++){
240         line1.add( i+1 , this.history.returnNumberOfOperations(i) );
241     }
242     final XYSeries line2 = new XYSeries( "Number of Operations Estimation" );
243     for(int i=0;i<S.length;i++){
244         line2.add( i+1 , S[i] );
245     }
246     final XYSeriesCollection dataset = new XYSeriesCollection( );
247     dataset.addSeries( line1 );
248     dataset.addSeries( line2 );
249     JFreeChart xylineChart = ChartFactory.createXYLineChart(
250         "MTurk: Number of Operations Estimations " ,
251         "Category" ,
252         "Score" ,
253         dataset ,
254         PlotOrientation.VERTICAL ,
255         true , true , false);
256     ChartFrame frame = new ChartFrame("MTurk: Number of Operations Estimations",
xylineChart);
257     frame.pack();
258     frame.setVisible(true);
259 }
260 }

```