```java
package lehmansrulesgraphsdisplay;

import java.util.ArrayList;

public class GraphVisualizer {
    private History h;

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

    public ChartPanel []firstRuleGraphGenerator(){
        ChartPanel []panels= new ChartPanel[2];
        DefaultCategoryDataset objDataset = new DefaultCategoryDataset();
        ArrayList<VersionInfo> versionlist = h.getVersions();
        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);
        }
        JFreeChart objChart = ChartFactory.createBarChart(
                "Changes Bar Chart",
                "Version ID",
                "Changes",
                objDataset,
                PlotOrientation.VERTICAL,
                true,
                true,
                false
            );
        panels[0] = new ChartPanel(objChart);
        DefaultCategoryDataset objDataset2 = new DefaultCategoryDataset();
        int [][]table;
        if(versionlist.size()<=0){
            return null;
        }
        int first= versionlist.get(0).yearOfDate();
        int last = versionlist.get(versionlist.size()-1).yearOfDate();
        int size = last-first+1;
        table = new int[size][2];
        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);

        }
        JFreeChart objChart2 = ChartFactory.createBarChart(
                "Version per Year Bar Chart",
                "Version ID",
                "Versions",
                objDataset2,
                PlotOrientation.VERTICAL,
                true,
                true,
                false
            );
        panels[1] = new ChartPanel(objChart2);

        return panels;

    }

    public ChartPanel[] secondRuleGraphGenerator()
    {
        ChartPanel []panels= new ChartPanel[2];
        DefaultCategoryDataset objDataset = new DefaultCategoryDataset();
        ArrayList<VersionInfo> versionlist = h.getVersions();
        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);
        }
        JFreeChart objChart = ChartFactory.createLineChart(
                "Complexity Line Chart",
                "Version ID",
                "Complexity",
                objDataset,
                PlotOrientation.VERTICAL,
                true,
                true,
                false
            );
        panels[0] = new ChartPanel(objChart);
        DefaultCategoryDataset objDataset2 = new DefaultCategoryDataset();
        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();
            String xaxis = versionlist.get(i).getID();
            objDataset2.setValue(dstructs,"Structs", xaxis);
        }
        JFreeChart objChart2 = ChartFactory.createBarChart(
                "Maintainance Bar Chart",
                "Version ID",
                "Maintainance",
                objDataset2,
                PlotOrientation.VERTICAL,
                true,
                true,
                false
            );
        panels[1] = new ChartPanel(objChart2);
        return panels;
    }



    public ChartPanel[] thirdRuleGraphGenerator()
    {
        ChartPanel []panels= new ChartPanel[2];
        DefaultCategoryDataset objDataset = new DefaultCategoryDataset();
        ArrayList<VersionInfo> versionlist = h.getVersions();
        int i;
        for(i=0; i<versionlist.size(); i++){
            double dops = versionlist.get(i).getopsMetrics().getGrowthRate();
            String xaxis = versionlist.get(i).getID();
            objDataset.setValue(dops,"Operations", xaxis);
        }
        JFreeChart objChart = ChartFactory.createLineChart(
                "Growth Rate Line Chart",
                "Version ID",
                "Growth Rate",
                objDataset,
                PlotOrientation.VERTICAL,
                true,
                true,
                false
            );
        panels[0] = new ChartPanel(objChart);
        DefaultCategoryDataset objDataset2 = new DefaultCategoryDataset();
        for(i=0; i<versionlist.size(); i++){
            double dstructs = versionlist.get(i).getstructsMetrics().getGrowthRate();
            String xaxis = versionlist.get(i).getID();
            objDataset2.setValue(dstructs,"Structs", xaxis);
        }
        JFreeChart objChart2 = ChartFactory.createLineChart(
                "Growth Rate Bar Chart",
                "Version ID",
                "Growth Rate",
                objDataset2,
                PlotOrientation.VERTICAL,
                true,
                true,
                false
            );
        panels[1] = new ChartPanel(objChart2);

        return panels;
    }

    public ChartPanel[] fourthRuleGraphGenerator()
    {
        ChartPanel []panels= new ChartPanel[2];
        DefaultCategoryDataset objDataset = new DefaultCategoryDataset();
        ArrayList<VersionInfo> versionlist = h.getVersions();
        int i;
        for(i=0; i<versionlist.size(); i++){
            double dops = versionlist.get(i).getopsMetrics().getWorkRate();
            String xaxis = versionlist.get(i).getID();
            objDataset.setValue(dops,"Operations", xaxis);
        }
        JFreeChart objChart = ChartFactory.createLineChart(
                "Work Rate Line Chart",
                "Version ID",
                "Work Rate",
                objDataset,
                PlotOrientation.VERTICAL,
                true,
                true,
                false
            );
        panels[0] = new ChartPanel(objChart);
        DefaultCategoryDataset objDataset2 = new DefaultCategoryDataset();
        for(i=0; i<versionlist.size(); i++){
            double dstructs = versionlist.get(i).getstructsMetrics().getWorkRate();
            String xaxis = versionlist.get(i).getID();
            objDataset2.setValue(dstructs,"Structs", xaxis);
        }
        JFreeChart objChart2 = ChartFactory.createLineChart(
                "Work Rate Bar Chart",
                "Version ID",
                "Work Rate",
                objDataset2,
                PlotOrientation.VERTICAL,
                true,
                true,
                false
            );
        panels[1] = new ChartPanel(objChart2);
        return panels;
    }

    public ChartPanel[] fifthRuleGraphGenerator()
    {
        return thirdRuleGraphGenerator();
    }

    public ChartPanel[] sixthRuleGraphGenerator()
    {

        ChartPanel []panels= new ChartPanel[2];
        DefaultCategoryDataset objDataset = new DefaultCategoryDataset();
        ArrayList<VersionInfo> versionlist = h.getVersions();
        int i;
        for(i=0; i<versionlist.size(); i++){
            double dops = versionlist.get(i).getopsMetrics().getNumber();
            String xaxis = versionlist.get(i).getID();
            objDataset.setValue(dops,"Operations", xaxis);
        }
        JFreeChart objChart = ChartFactory.createLineChart(
                "Work Rate Line Chart",
                "Version ID",
                "Work Rate",
                objDataset,
                PlotOrientation.VERTICAL,
                true,
                true,
                false
            );
        panels[0] = new ChartPanel(objChart);
        DefaultCategoryDataset objDataset2 = new DefaultCategoryDataset();
        for(i=0; i<versionlist.size(); i++){
            double dstructs = versionlist.get(i).getstructsMetrics().getNumber();
            String xaxis = versionlist.get(i).getID();
            objDataset2.setValue(dstructs,"Structs", xaxis);
        }
        JFreeChart objChart2 = ChartFactory.createLineChart(
                "Work Rate Bar Chart",
                "Version ID",
                "Work Rate",
                objDataset2,
                PlotOrientation.VERTICAL,
                true,
                true,
                false
            );
        panels[1] = new ChartPanel(objChart2);

        return panels;
    }

    public ChartPanel[] seventhRuleGraphGenerator()
    {
        return null;
    }

    public ChartPanel[] eighthRuleGraphGenerator()
    {
        return null;
    }
}
```