```java
package datamodel;

import java.util.ArrayList;

public class History {
    private String name;
    private int initNumOfOps;
    private int initNumOfStructs;
    private ArrayList <VersionInfo> listofVersions;
    private LehmansRuleRecord []Rules;

    public History(String name, int nops, int nstr){
        listofVersions = new ArrayList<VersionInfo>();
        Rules = new LehmansRuleRecord[8];
        for(int i=0; i<8; i++){
            Rules[i] = new LehmansRuleRecord(i+1);
        }
        this.name = name;
        initNumOfOps = nops;
        initNumOfStructs = nstr;
    }

    public LehmansRuleRecord getRule(int i){
        if(i<=0 || i>8) return null;
        return Rules[i-1];
    }

    public void addVersionInfo(VersionInfo v){
        listofVersions.add(v);
    }

    public int getTotalVersions(){
        return listofVersions.size();
    }

    public ArrayList<VersionInfo> getVersions(){
        return listofVersions;
    }

    public boolean hasUserCheckedAllRules(){
        for(int i=0; i<8; i++){
            if(Rules[i].hasUserCheckedRule()==false) return false;
        }
        return true;
    }

    public String toString(){
        return name;
    }

    public DefaultCategoryDataset createChangesDataset() {
        DefaultCategoryDataset objDataset = new DefaultCategoryDataset();

        for(int i=0; i<getVersions().size(); i++){
            objDataset.setValue(getOperationChangesPerVersion(i),
                    "Operations", getXAxisLabelPerVersion(i));
            objDataset.setValue(getStructsChangesPerVersion(i),
                    "Structs", getXAxisLabelPerVersion(i));
        }
        return objDataset;
    }

```

```java
    public double getOperationChangesPerVersion(int i) {
        return (double) getVersions().get(i).getopsChanges().getNumOfChanges();
    }

    public double getStructsChangesPerVersion(int i) {
        return (double) getVersions().get(i).getstructsChanges().getNumOfChanges();
    }

    public String getXAxisLabelPerVersion(int i) {
        return getVersions().get(i).getID();
    }

    public DefaultCategoryDataset createVersionsDataset() {

        if(getVersions().size()<=0){
            return null;
        }

        int [][] versionsPerYearDistribution = initializeNumOfVersionsPerYear();
        populateNumOfVersionsPerYear(versionsPerYearDistribution);

        DefaultCategoryDataset objDataset2 =
  createVersionsPerYearDataset(versionsPerYearDistribution);

        return objDataset2;
    }

    public int [][] initializeNumOfVersionsPerYear() {
        int [][] versionsPerYearDistribution = new int[getProjectLifetimeInYears()][2];

        for(int i=0; i<getProjectLifetimeInYears(); i++){
            // 1st column holds a specific year label
            versionsPerYearDistribution[i][0] = getYearOfFirstVersion() + i;
            // 2nd column holds no of versions this year
            versionsPerYearDistribution[i][1] = 0;
        }
        return versionsPerYearDistribution;
    }

    public int getProjectLifetimeInYears() {
        return getYearOfLastVersion()-getYearOfFirstVersion()+1;
    }

    public int getYearOfLastVersion() {
        return getVersions().get(getVersions().size()-1).yearOfDate();
    }

    public int getYearOfFirstVersion() {
        return getVersions().get(0).yearOfDate();
    }

    public void populateNumOfVersionsPerYear(int[][] versionsPerYearDistribution) {
        for(int i=0; i<getVersions().size(); i++){
            for(int j=0; j<getProjectLifetimeInYears(); j++){
                if(versionsPerYearDistribution[j][0] == getYearPerVersion(i)){
                    versionsPerYearDistribution[j][1]++;
                    break;
                }
            }
        }
    }

```

```
128     public int getYearPerVersion(int i) {
129         return getVersions().get(i).yearOfDate();
130     }
131 
132     public DefaultCategoryDataset createVersionsPerYearDataset(int[][]
    versionsPerYearDistribution) {
133         DefaultCategoryDataset objDataset2 = new DefaultCategoryDataset();
134         for(int i=0; i<getProjectLifetimeInYears(); i++){
135             String xaxis = "'"+(versionsPerYearDistribution[i][0]-2000);
136             objDataset2.setValue(versionsPerYearDistribution[i][1],"Year Versions", xaxis);
137 
138         }
139         return objDataset2;
140     }
141 }
142 
```