

MIST352 Homework #4 (100 points)

Problem Statement:

The objective of this project is to convert Comma Separated Values files (csv) to Attribute-Relation File Format (ARFF). Your program should read the csv files given in HW4 folder of your project's directory and create local text arff files named [YourLastName]_[YourFirstName]_[File name].arff at the same directory of the csv files. The data folder has 7 csv files, your project should create text files with the extension ".arff" for each of those files. You will have to utilize one dimensional and two-dimensional arrays as stated in the instructions.

The arff files should have the correct format for each of the files as shown in Figure 1. Please explore the example of the Ant1.csv and how its Ant1.arff (given to you as example) should look like.

```
@relation Ant1

@attribute File Text
@attribute WMC numeric
@attribute DIT numeric
@attribute NOC numeric
@attribute CBO numeric
@attribute RFC numeric
@attribute LCOM numeric
@attribute CA numeric
@attribute CE numeric
@attribute NPM numeric
@attribute LCOM3 numeric
@attribute LOC numeric
@attribute DAM numeric
@attribute MOA numeric
@attribute MFA numeric
@attribute CAM numeric
@attribute IC numeric
@attribute CBM numeric
@attribute AMC numeric
@attribute CC numeric
@attribute MaX_CC numeric
@attribute SumOfLoC+ numeric
@attribute MaxOfLoC+ numeric
@attribute SumOfLoC- numeric
@attribute MaxOfLoC- numeric
@attribute Max_Set numeric
@attribute Avg_Set numeric
@attribute No_Rev numeric
@attribute Sum_Churn numeric
@attribute Max_Churn numeric
@attribute Age numeric
@attribute CountOfDeveloper numeric
@attribute Ref { TRUE , FALSE }
@attribute PRE_BUG { TRUE , FALSE }
@attribute BUG_FIX { TRUE , FALSE }

@data
src/main/org/apache/tools/ant/AntClassLoader.java,src/main/org/apache/tools/ant/BuildEvent.java,src/main/org/apache/tools/ant/BuildException.java,src/main/org/apache/t
11,2,0,7,15,13,5,3,11,0.75,97,1,3,0.2,0.220779221,0,0,7.272727273,1,1,1.00,1,1.00,1,288,288.00,1,2.00,2,40.14,1,FALSE,FALSE,FALSE
8,3,0,4,41,10,0,4,8,0.80952381,236,1,0,0.840909091,0.875,1,1,27.75,2.75,14,79.00,28,13.00,5,288,114.40,5,92.00,30,40.14,4,FALSE,FALSE,TRUE
5,1,0,4,20,0,0,4,4,0.33333333,144,1,1,0,0.9,0,0,27.2,2.4,5,176.00,155,35.00,31,280,73.25,4,211.00,155,45.28,2,FALSE,FALSE,FALSE
3,2,0,8,18,3,1,7,2,1,132,0,0,0.92,0.5,0,0,42,1,1,120.00,108,4.00,2,280,74.25,4,124.00,108,38.85,3,FALSE,FALSE,FALSE
2,2,0,8,17,1,1,7,2,2,97,1,0,0.95833333,1,0,0,47,1,1,120.00,113,17.00,11,280,66.40,5,137.00,113,38.85,3,FALSE,FALSE,FALSE
3,1,0,8,16,3,1,7,1,2,125,0,0,0.44444444,0,0,40.66666667,1,1,169.00,151,2.00,1,10,4.20,5,171.00,151,38.85,1,FALSE,FALSE,FALSE
```

FIGURE 1: FORMAT OF ARFF FILES

Arff files are ASCII text files that describe a list of instances sharing a set of attributes [1]. ARFF are normal text files where , unlike csv files, the names of the columns are listed on top of the file, followed by the actual data. Please explore and compare the Ant1.csv and Ant1.arff files in the repository.

Instructions:

1. Pull the MIST352 repository and explore HW4 folder. **[10 points]**
 - a. Copy the entire HW4 project folder to the Homework folder in your Git repository. You will submit this, and this will become your HW4.
 - b. Do not modify any methods where the documentation states (* Keep as is).
 - c. Only code the methods where the documentation states (as * You need to code this).
 - d. **The program should work with these methods, you may add your own methods if you want for testing however, the program should only use ALL the methods given in the original project and should not use any of your extra methods.**
 - e. Provide documentation for everything you do. **[5 points]**
 - f. You may only use ChatGPT for coding the main method only. You are not allowed to use it when coding the CsV2Arff.java file.
2. The HW4.java class contains the main method:
 - a. Code the method PrintArray as given. (Use this method anytime you want to printout the contents of any 1-D array. **[10 points]**
 - b. Inside the main method: follow the comments given in the main:
 - i. Call the methos PrintsArray() to print out the contents of the array defined in line 26. The output should be similar to the output in Figure 2. **[5 points]**

```
Ant1.csv  
Ant2.csv  
Ant3.csv  
Ant4.csv  
Ant5.csv  
Ant6.csv  
Ant7.csv
```

FIGURE 2: OUTPUT WHEN PRINTING OUT THE ARRAY IN THE MAIN METHOD

- ii. Write code to access the csvFilesNames array and modify the content of each element by adding the directory to each name of the csv files in the array. For example, the first element in the array has **Ant1.csv**, your code should change that to **src/Data/Ant1.csv** and so on. The output should be like the output give in Figure 3. **[5 points]**

```
src/Data/Ant1.csv
src/Data/Ant2.csv
src/Data/Ant3.csv
src/Data/Ant4.csv
src/Data/Ant5.csv
src/Data/Ant6.csv
src/Data/Ant7.csv
```

FIGURE 3: FIGURE 2: OUTPUT WHEN PRINTING OUT THE ARRAY AFTER ADDING DIRECTORY TO EACH ELEMENT IN THE MAIN METHOD

- c. Inside the main method, write code that iterates through the names of the csv files in the array csvFilNames, each time you read a new element in the array, create a new Csv2Arff object using the data in that element.
 - i. For each Csv2Arff object, call the method `Convert2Arff` to convert that file to arff file. Make sure your arff files are named [Your First Name]_[Your Last Name]_[FileName].arff **[10 points]**
- d. Inside the main method, ask user for a name of a file name, a column number, and a row number. Call the method `Retrieve Cell` and display that value from that file. For example, if user provides the file name Ant1, and provided 4 for the column and 8 for the row, your program should display the value in the 4th column and 8th row in that file and so on. **[10 points]**
3. Given the class named Csv2Arff.java, this class should have one private String parameter named location.
 - a. Do not set accessor or mutators to this class. **[10 points]**
 - b. Create the two methods as given in the Csv2Arff.java. **[30 points]**

Deliverables:

1. Submit your HW on your GitHub repository.
2. Make a submission on eCampus by simply typing "Submitted on GitHub". **[5 points]**
3. The seven arff files named correctly.

Notes:

- You will lose 50% if your program has syntax errors.
- Group collaboration is not permitted.

Bonus (5 points added to your final course GPA): In the `Csv2Arff` class, create a method named `PrintMinMaxInfo(String strFile)` that prints out the name of the file in the first column of the CSV file that has the minimum non-zero value, and the column name where that minimum value appeared, as well as the maximum non-zero value and the column name where that maximum value appeared, for the file name given in `strFile`. The name of the file should be accepted in the main method and passed to the `Csv2Arff` object. For example, if the user provides the name `Ant1`, calling the `PrintMinMaxInfo`, for that file should print out:

Name: `Ant1.csv`

Maximum non-zero value:

File name: `src/main/org/apache/tools/ant/Project.java`

Maximum Value: 2247 found in column: LCOM

Minimum non-zero value:

File name: `src/main/org/apache/tools/ant/Project.java`

Minimum Value: 0.12773192 found in column: CAM

Bibliography

[1] WEKA, "ARFF," [Online]. Available: <https://www.cs.waikato.ac.nz/ml/weka/arff.html>.