Unit III Project

Keith Lopez
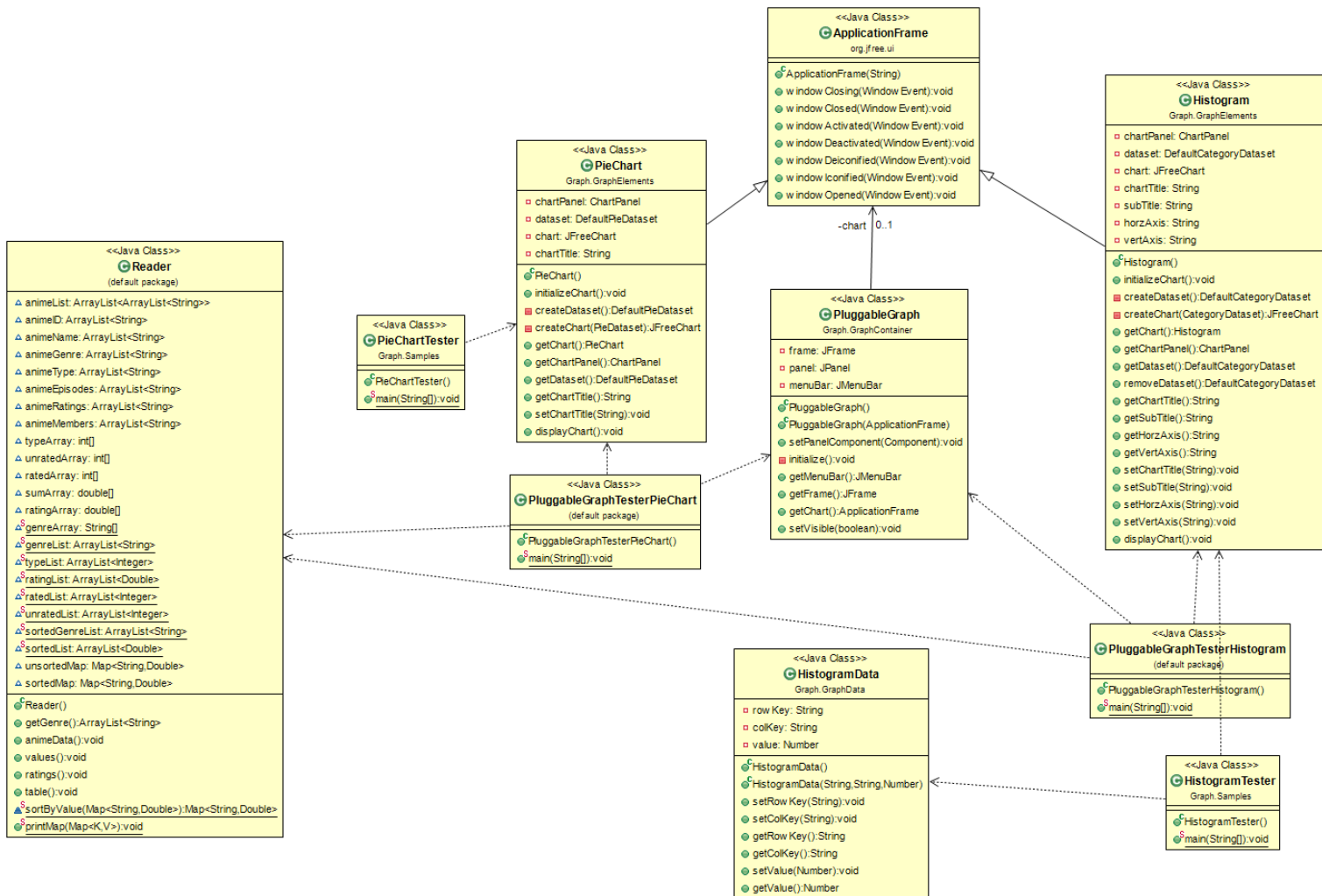
Psuedocode

```
for i: to array length:i++
      double ratenumber:=0
      If (ratings is empty AND genre(i) contains genre(j))
            ratenumber := 0
            unratedArray[j] := unratedarray +1
      else if( genre(i) contains genre(j))
            ratenumber := ratings(i);
            ratedArray[j] +:=1
```

For loop in Reader.java under ratings() method
      -The loops gets the total number of times each genre was
      rated and how many times it was not rated.

UML

<<Java Class>>
ApplicationFrame
org.jfree.ui

- ApplicationFrame(String)
- window Closing(Window Event):void
- window Closed(Window Event):void
- window Activated(Window Event):void
- window Deactivated(Window Event):void
- window Deiconified(Window Event):void
- window Iconified(Window Event):void
- window Opened(Window Event):void

-chart 0..1

<<Java Class>>
PieChart
Graph.GraphElements

- chartPanel: ChartPanel
- dataset: DefaultPieDataset
- chart: JFreeChart
- chartTitle: String

- PieChart()
- initializeChart():void
- createDataset():DefaultPieDataset
- createChart(PieDataset):JFreeChart
- getChart():PieChart
- getChartPanel():ChartPanel
- getDataset():DefaultPieDataset
- getChartTitle():String
- setChartTitle(String):void
- displayChart():void

<<Java Class>>
PieChartTester
Graph.Samples

- PieChartTester()
- main(String[]):void

<<Java Class>>
PluggableGraph
Graph.GraphContainer

- frame: JFrame
- panel: JPanel
- menuBar: JMenuBar

- PluggableGraph()
- PluggableGraph(ApplicationFrame)
- setPanelComponent(Component):void
- initialize():void
- getMenuBar():JMenuBar
- getFrame():JFrame
- getChart():ApplicationFrame
- setVisible(boolean):void

<<Java Class>>
Histogram
Graph.GraphElements

- chartPanel: ChartPanel
- dataset: DefaultCategoryDataset
- chart: JFreeChart
- chartTitle: String
- subTitle: String
- horzAxis: String
- vertAxis: String

- Histogram()
- initializeChart():void
- createDataset():DefaultCategoryDataset
- createChart(CategoryDataset):JFreeChart
- getChart():Histogram
- getChartPanel():ChartPanel
- getDataset():DefaultCategoryDataset
- removeDataset():DefaultCategoryDataset
- getChartTitle():String
- getSubTitle():String
- getHorzAxis():String
- getVertAxis():String
- setChartTitle(String):void
- setSubTitle(String):void
- setHorzAxis(String):void
- setVertAxis(String):void
- displayChart():void

<<Java Class>>
Reader
(default package)

- animeList: ArrayList<ArrayList<String>>
- animeID: ArrayList<String>
- animeName: ArrayList<String>
- animeGenre: ArrayList<String>
- animeType: ArrayList<String>
- animeEpisodes: ArrayList<String>
- animeRatings: ArrayList<String>
- animeMembers: ArrayList<String>
- typeArray: int[]
- unratedArray: int[]
- ratedArray: int[]
- sumArray: double[]
- ratingArray: double[]
- genreArray: String[]
- genreList: ArrayList<String>
- typeList: ArrayList<Integer>
- ratingList: ArrayList<Double>
- ratedList: ArrayList<Integer>
- unratedList: ArrayList<Integer>
- sortedGenreList: ArrayList<String>
- sortedList: ArrayList<Double>
- unsortedMap: Map<String,Double>
- sortedMap: Map<String,Double>

- Reader()
- getGenre():ArrayList<String>
- animeData():void
- values():void
- ratings():void
- table():void
- sortByValue(Map<String,Double>):Map<String,Double>
- printMap(Map<K,V>):void

<<Java Class>>
PluggableGraphTesterPieChart
(default package)

- PluggableGraphTesterPieChart()
- main(String[]):void

<<Java Class>>
PluggableGraphTesterHistogram
(default package)

- PluggableGraphTesterHistogram()
- main(String[]):void

<<Java Class>>
HistogramData
Graph.GraphData

- row Key: String
- colKey: String
- value: Number

- HistogramData()
- HistogramData(String,String,Number)
- setRow Key(String):void
- setColKey(String):void
- getRow Key():String
- getColKey():String
- setValue(Number):void
- getValue():Number

<<Java Class>>
HistogramTester
Graph.Samples

- HistogramTester()
- main(String[]):void

Source code attached as "Graph".

- ∨ 📂 Graph
  - ∨ 📦 src
    - ∨ ⊞ (default package)
      - › 🗾 PluggableGraphTesterHistogram.java
      - › 🗾 PluggableGraphTesterPieChart.java
      - › 🗾 Reader.java
    - ∨ ⊞ Graph.GraphContainer
      - › 🗾 PluggableGraph.java
    - ∨ ⊞ Graph.GraphData
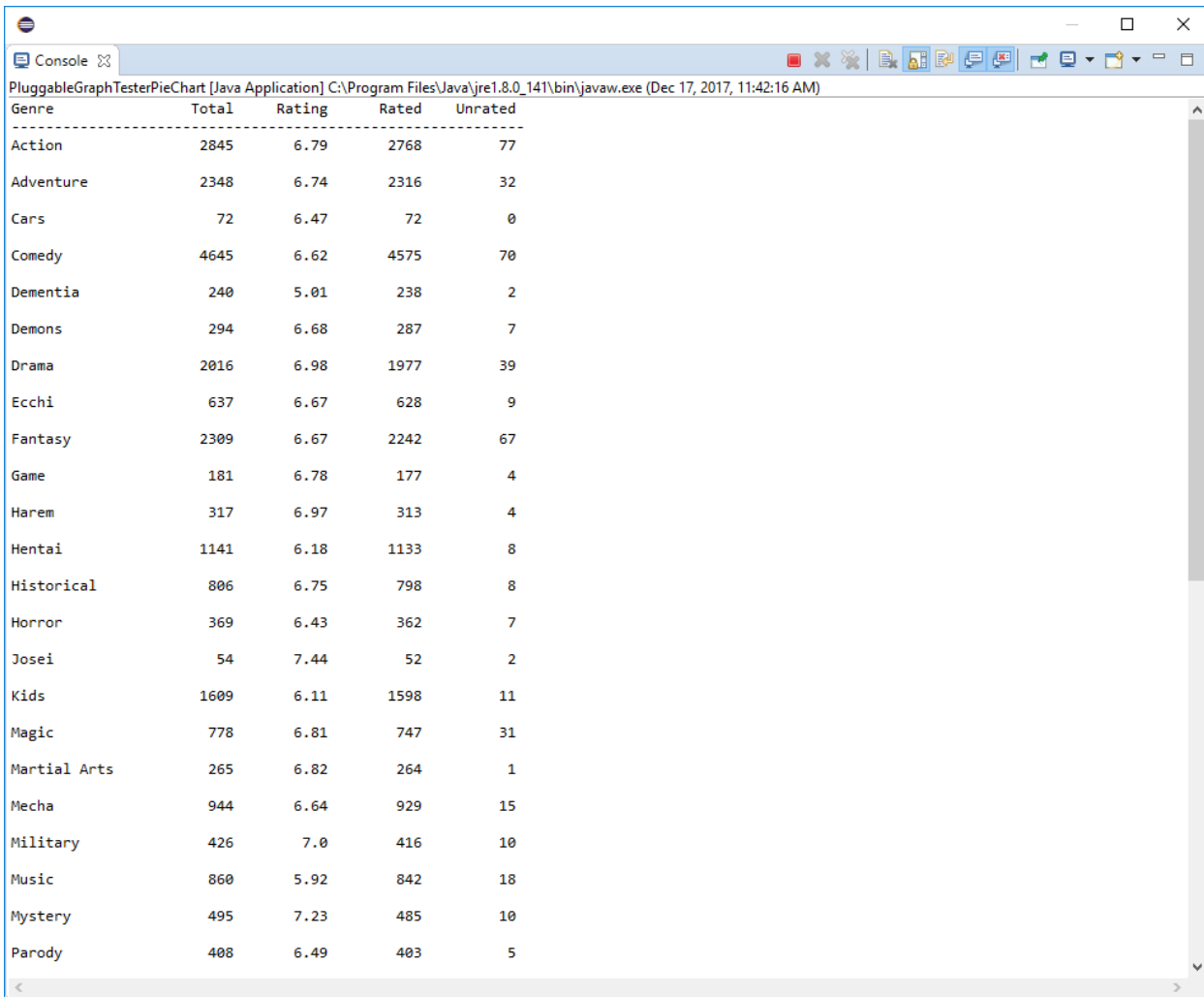   - › 🗾 FunctionGrapherData.java
      - › 🗾 HistogramData.java
    - ∨ ⊞ Graph.GraphElements
      - › 🗾 FunctionGrapher.java
      - › 🗾 Histogram.java
      - › 🗾 PieChart.java
    - ∨ ⊞ Graph.Samples
      - › 🗾 FunctionGrapherTester.java
      - › 🗾 HistogramTester.java
      - › 🗾 PieChartTester.java
  - › 📚 JRE System Library [JavaSE-1.8]
  - › 📚 Referenced Libraries
  - › 📂 lib
  - 📄 anime.txt

Program Output:

```
PluggableGraphTesterPieChart [Java Application] C:\Program Files\Java\jre1.8.0_141\bin\javaw.exe (Dec 17, 2017, 11:42:16 AM)
Genre              Total    Rating    Rated   Unrated
-----------------------------------------------------
Action             2845      6.79     2768       77

Adventure          2348      6.74     2316       32

Cars                 72      6.47       72        0

Comedy             4645      6.62     4575       70

Dementia            240      5.01      238        2

Demons              294      6.68      287        7

Drama              2016      6.98     1977       39

Ecchi               637      6.67      628        9

Fantasy            2309      6.67     2242       67

Game                181      6.78      177        4

Harem               317      6.97      313        4

Hentai             1141      6.18     1133        8

Historical          806      6.75      798        8

Horror              369      6.43      362        7

Josei                54      7.44       52        2

Kids               1609      6.11     1598       11

Magic               778      6.81      747       31

Martial Arts        265      6.82      264        1

Mecha               944      6.64      929       15

Military            426       7.0      416       10

Music               860      5.92      842       18

Mystery             495      7.23      485       10

Parody              408      6.49      403        5
```

5
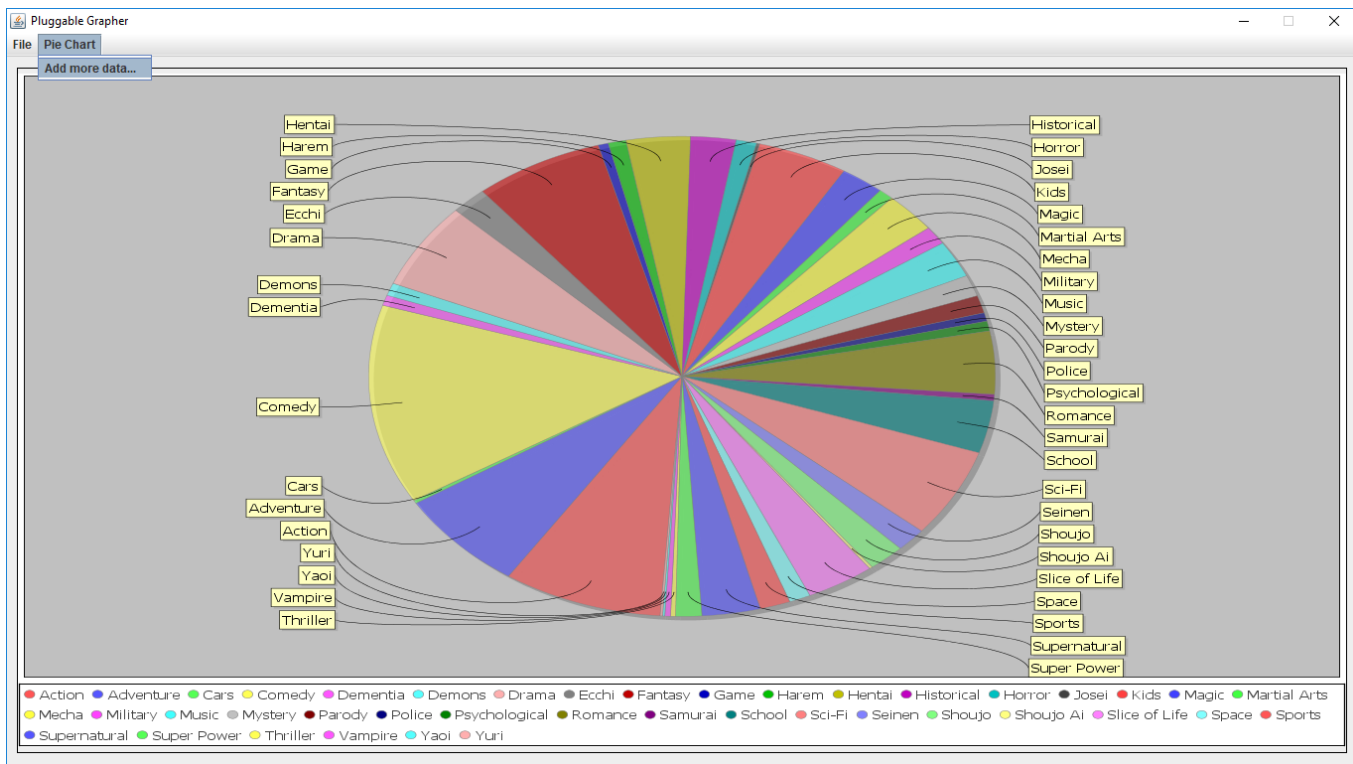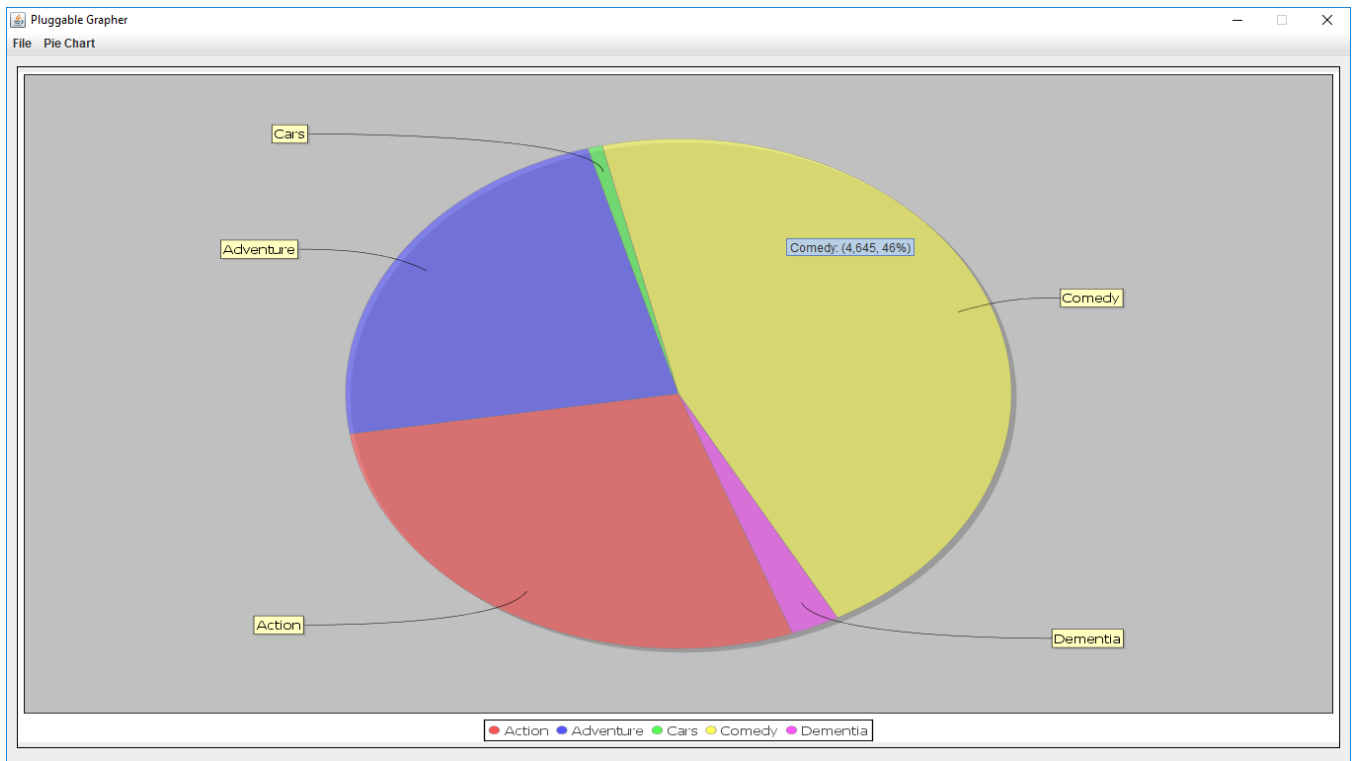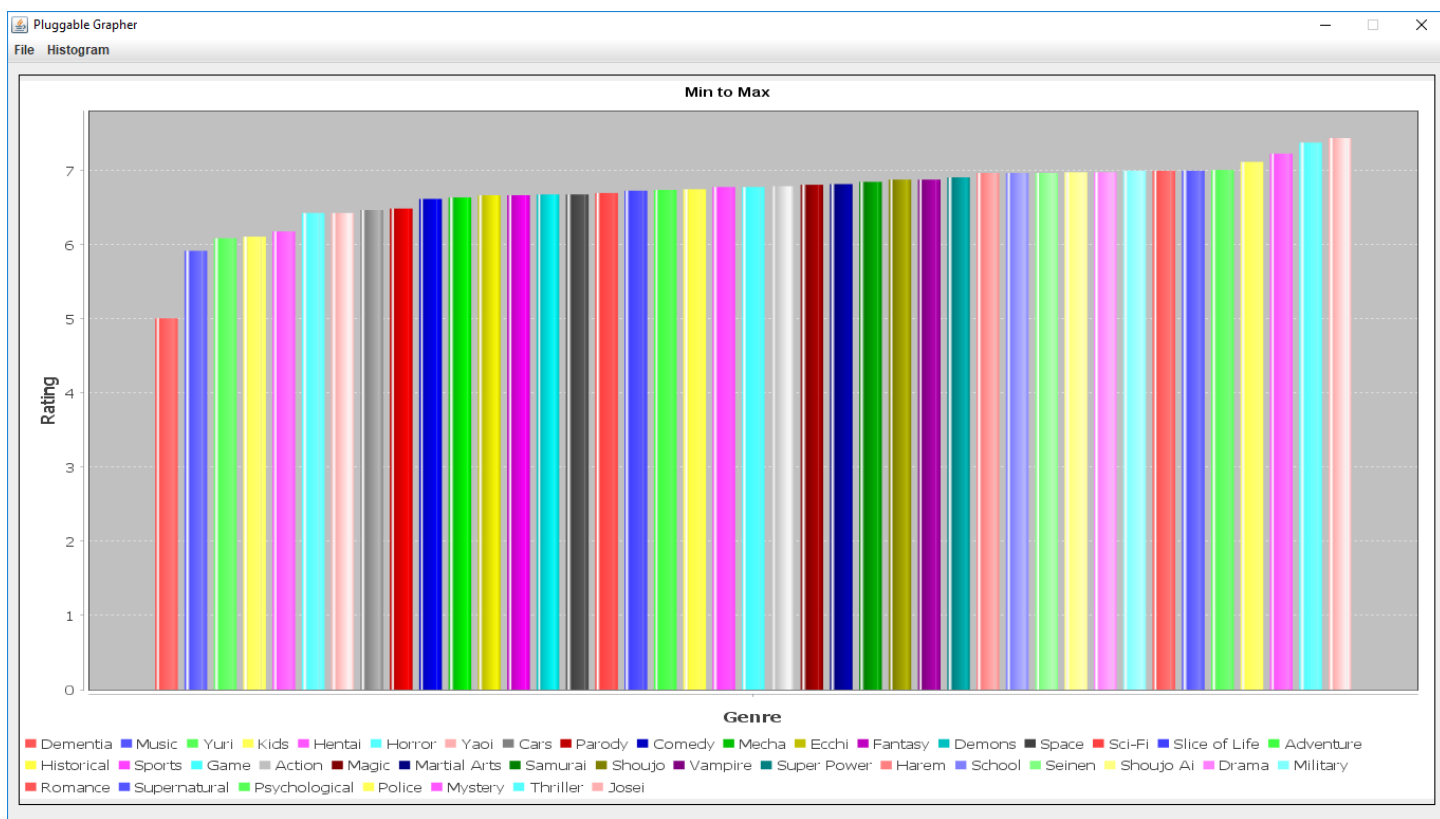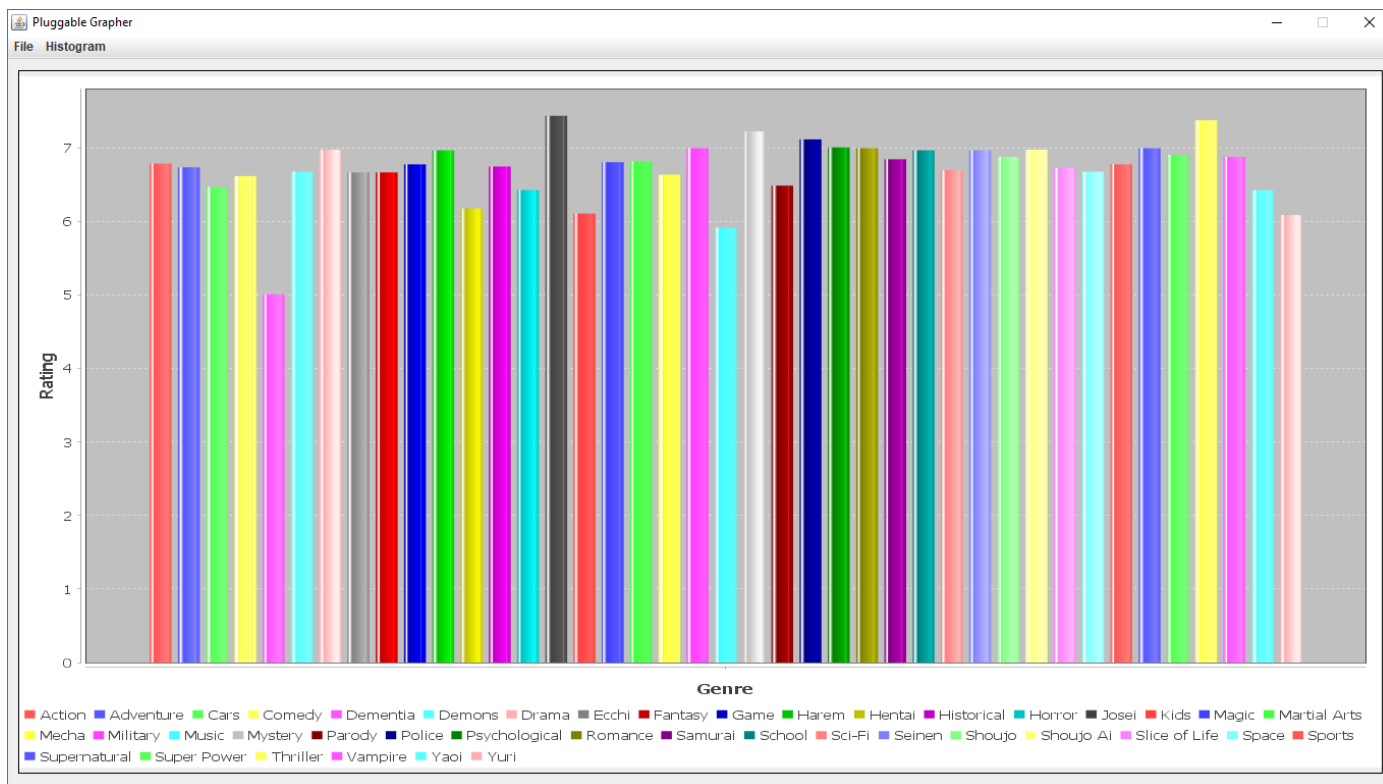
Package Explorer
- AdvancedPortfolioManager
- Emilia
- Exam-Practical
- FileException
- FinalExamPractical
- FMML
- FMML2
- Graph
  - src
    - (default package)
      - PluggableGraphTesterFunctionGrapher.java
      - PluggableGraphTesterHistogram.java
      - PluggableGraphTesterPieChart.java
    - Graph.GraphContainer
      - PluggableGraph.java
    - Graph.GraphData
      - FunctionGrapherData.java
      - HistogramData.java
    - Graph.GraphElements
      - FunctionGrapher.java
      - Histogram.java
      - PieChart.java
    - Graph.Samples
      - FunctionGrapherTester.java
      - HistogramTester.java
      - PieChartTester.java
    - Other
      - Reader.java
  - JRE System Library [JavaSE-1.8]
  - Referenced Libraries
  - lib
    - anime.txt
- InputOutput
  - src
    - (default package)
      - InputOutput.java
  - JRE System Library [JavaSE-1.8]
  - milkshake.txt
- reviewfinal
- Test
- Unit 2 Practical
- Unit II Project

Tabs: PluggableGr... PluggableGr... FunctionGra... HistogramDa... FunctionGra... InputOutput... Reader.java

```
17    // Create the Scanner object and scope to the entire main method
18    Scanner in = null;
19
20    try {
21        // Try to open the file
22        in = new Scanner(file);
23
24        try {
25            // Read in the data one line at a time
26                //System.out.println(in.nextLine());
27
28            while(in.hasNextLine()) {
29                // Do something with this String
30                //in.nextLine();
31                String input = in.nextLine();
32                String[] wordsArray = input.split("\t");
33                String animeID = wordsArray[0];
34                String animeName = wordsArray[1];
35                String animeGenre = wordsArray[2];
36                String animeType = wordsArray[3];
37                String animeEpisodes = wordsArray[4];
38                String animeRating = wordsArray[5];
39                String animeMembers = wordsArray[6];
40                System.out.println("Anime ID: " + animeID);
41                System.out.println("Name: " + animeName);
42                System.out.println("Genre: " + animeGenre);
43                System.out.println("Type: " + animeType);
44                System.out.println("Episodes: " + animeEpisodes);
45                System.out.println("Rating: " + animeRating);
46                System.out.println("Members: " + animeMembers);
47                System.out.println();
48            }
49
50
51            // Fix anything that broke and clean up
52        } catch (NoSuchElementException e) {
53            System.err.println("Record Error: " + e.getMessage());
54        } catch (IndexOutOfBoundsException e) {
55            System.err.println("Parse Error: " + e.getMessage());
56        } catch (NumberFormatException e) {
57            System.err.println("Data Error: " + e.getMessage());
58        } finally {
59            in.close();
```

Task List

Console

```
<terminated> Reader [Java Application] C:\Program Files\Java\jre1.8.0_141\bin\javaw.exe (Dec 16, 2017, 7:55:03 PM)
Anime ID: 5543
Name: Under World
Genre: Hentai
Type: OVA
Episodes: 1
Rating: 4.28
Members: 183

Anime ID: 5621
Name: Violence Gekiga David no Hoshi
Genre: Hentai
Type: OVA
Episodes: 4
Rating: 4.88
Members: 219

Anime ID: 6133
Name: Violence Gekiga Shin David no Hoshi: Inma Densetsu
Genre: Hentai
Type: OVA
Episodes: 1
Rating: 4.98
Members: 175

Anime ID: 26081
Name: Yasuji no Pornorama: Yacchimae!!
Genre: Hentai
Type: Movie
Episodes: 1
Rating: 5.46
Members: 142
```

Problems | Javadoc | Declaration | Coverage | Debug

0 items

Description	Resource	Path	Location	Type

Updates Available

Activate Windows

Console ⌸

PluggableGraphTesterHistogram [Java Application] C:\Program Files\Java\jre1.8.0_141\bin\javaw.exe (Dec 17, 2017, 2:45:14 PM)

```
Super Power            465       6.91      451        14

Thriller                87       7.38       86         1

Vampire                102       6.88      100         2

Yaoi                    39       6.43       38         1

Yuri                    42       6.09       41         1

--------------------------------------------------
Total:               34557       7.0     33897       660
```

Sorted List By Rating
Key : Dementia Value : 5.01
Key : Music Value : 5.92
Key : Yuri Value : 6.09
Key : Kids Value : 6.11
Key : Hentai Value : 6.18
Key : Horror Value : 6.43
Key : Yaoi Value : 6.43
Key : Cars Value : 6.47
Key : Parody Value : 6.49
Key : Comedy Value : 6.62
Key : Mecha Value : 6.64
Key : Ecchi Value : 6.67
Key : Fantasy Value : 6.67
Key : Demons Value : 6.68
Key : Space Value : 6.68
Key : Sci-Fi Value : 6.7
Key : Slice of Life Value : 6.73
Key : Adventure Value : 6.74
Key : Historical Value : 6.75
Key : Sports Value : 6.78
Key : Game Value : 6.78
Key : Action Value : 6.79
Key : Magic Value : 6.81
Key : Martial Arts Value : 6.82
Key : Samurai Value : 6.85
Key : Shoujo Value : 6.88
Key : Vampire Value : 6.88
Key : Super Power Value : 6.91
Key : Harem Value : 6.97
Key : School Value : 6.97
Key : Seinen Value : 6.97
Key : Shoujo Ai Value : 6.98
Key : Drama Value : 6.98

Hypothesis

What is the most popular genre in anime?

The anime.txt dataset contains data about:

- Anime ID Number
- Anime Title
- Anime Genre
- Anime Type
- Anime Episodes
- Anime Rating
- Anime Members

I took all of this data and separated each anime by Genre. The following information resulted.
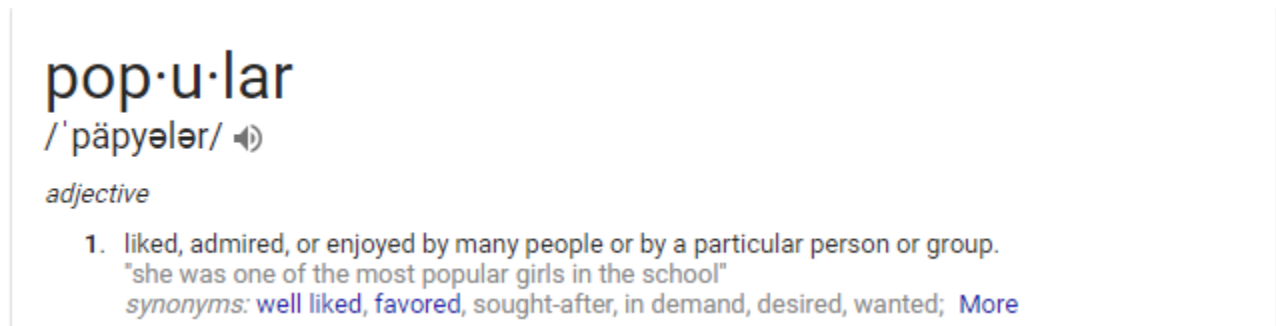
- **Total Anime:** 34557
  - **Total number of genres:** 40
  - **Genre containing most Anime:**
    - Comedy (4645)
      - **Rating**: 6.62
      - Rated 4575 times.
      - Unrated 70 times.
  - **Genre containing least Anime:**
    - Yaoi (39)
      - **Rating:** 6.43
      - Rated 38 times.
      - Unrated 1 times.
  - **Highest rated genre:**
    - Josei (54)
      - **Rating:** 7.44
      - Rated 52 times.
      - Unrated 2 times.
  - **Lowest rated genre:**
    - Dementia (240)
      - **Rating:** 5.01
      - Rated 238 times.
      - Unrated 2 times.

With this information we can see that the **most popular** genre in anime is **Comedy** according to *absolute* data. It is the genre that contains the most Anime along as the most ratings.

 Yaoi has the **least** Anime.

However, if we consider *relative* data, we can see that the genre **Josei** has the **highest** rating ranking out of all the other genres.

Dementai has the **lowest** rating.

pop·u·lar
/ˈpäpyələr/ 🔊
adjective
1. liked, admired, or enjoyed by many people or by a particular person or group.
   "she was one of the most popular girls in the school"
   synonyms: well liked, favored, sought-after, in demand, desired, wanted; More

Conclusion: Comedy is the most popular genre in anime since it is the genre with that has the most amount Anime and a relatively high rating of 6.62 among 4575 ratings.