/\*\*

\* This program will implement a GUI application for reading in a file containing employee information, and sorting them based on name, age, pay, etc...

\*

\* Name: Nabeel Hussain

\* Class: CMSC350

\* Professor: Didier Vergamini

\* Homework 1

\* Date: 10/25/2016

\*

\* **@author** Nabeel Hussain

\*/

My program for Homework 1 implements and displays a roster of employees of a company xyz. It uses an ArrayList to hold the employees that are added to the company, and then sorts them onto a GUI output text area by different key elements (first name, last name, position, pay, etc…).

My program has three classes. The first class is the data element class Employee, which holds the information required for each employee of the company. This includes an employee’s ID, first name, last name, job title, hourly pay, number of years worked, age, and gender.

The second class is the data manager class, which holds the methods that will be used by my GUI class to perform the different operations necessary by the application. This will include reading in a file and extracting employee information from it. It will also include writing the current list of employees of the company into a file. It will also include, a method to make random instances of the employee class.

The third class is used for the GUI, which will contain the buttons and a text area output to meet the functional requirements for this assignment.

Text files to be used to run my **program:**

Read in File: employeeReadFile.txt

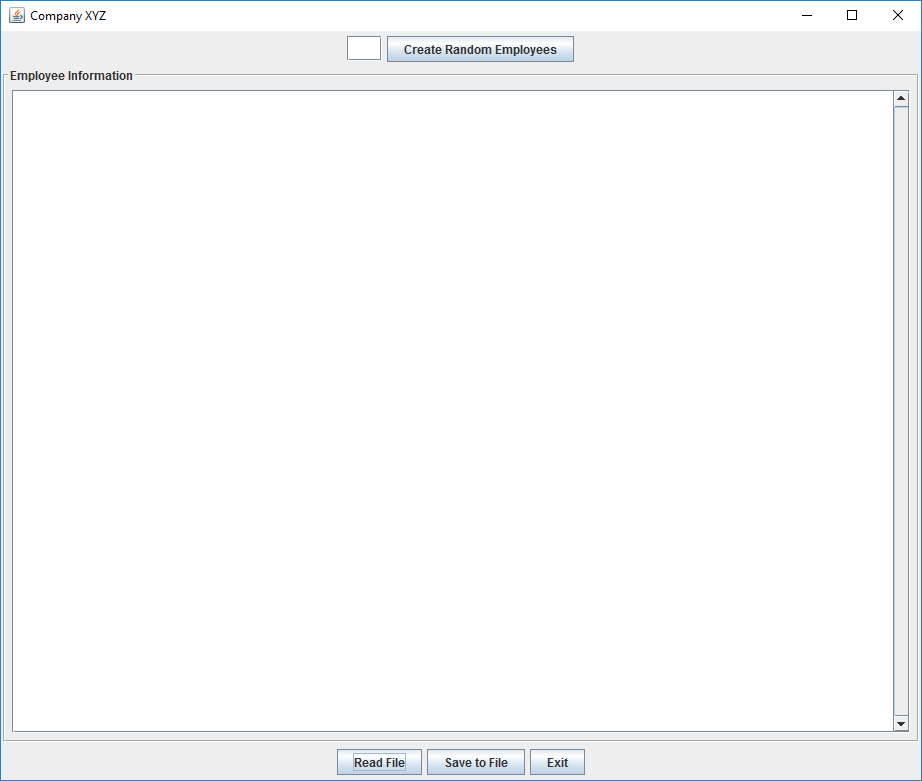
Save to File: employeeWriteFile.txt

Text files to be used with my **Junit tests:**

Read in File: TestReadFile.txt

Save to File: TestWriteFile.txt

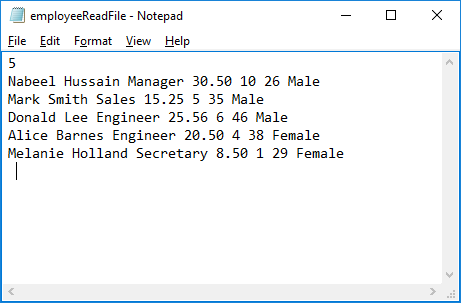
The application must always read in a file first, in order to create a pool of names and employee positions, for when the “Create Random Employees” button is selected.

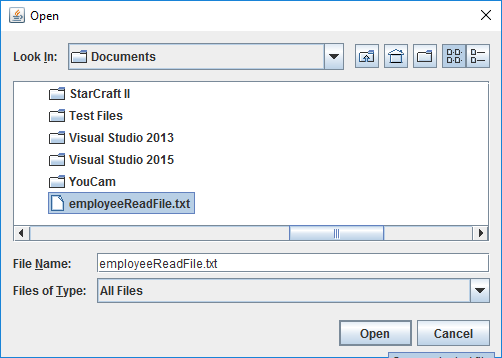
My GUI:

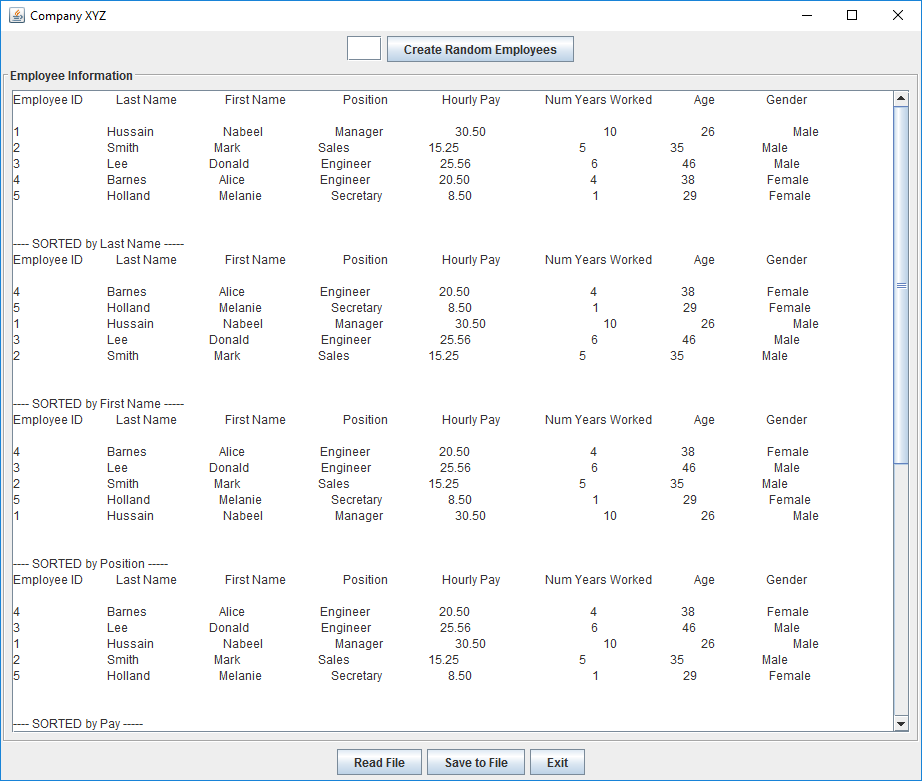
**Test Plan:**

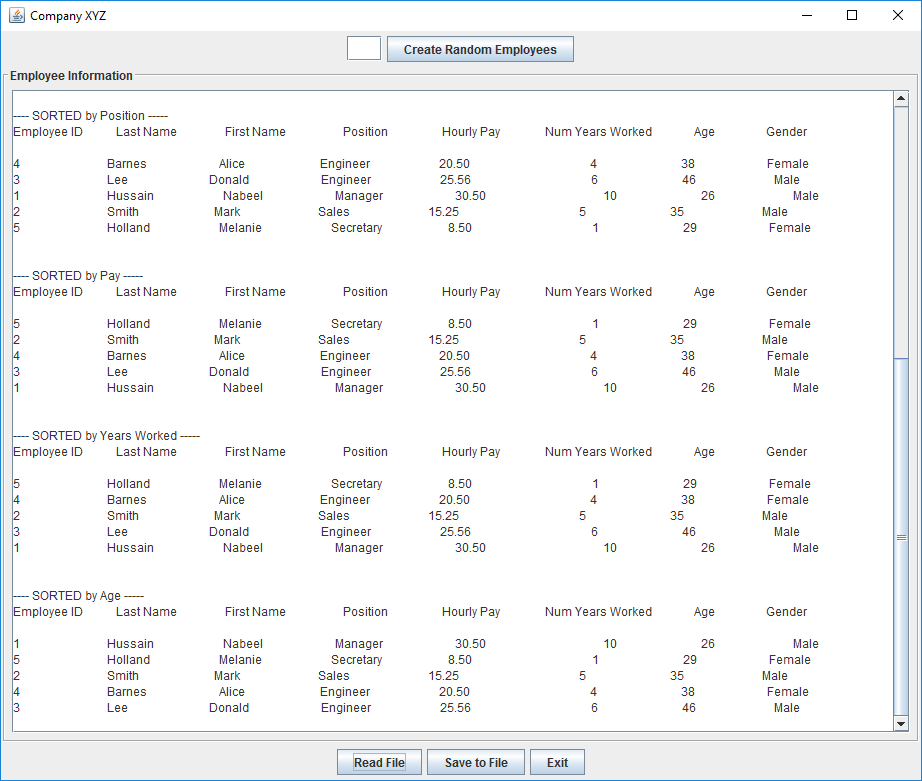
|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Input** | **Expected Output** | **Did Test Pass?** |
| 1 | Reads in the .txt file: employeeReadFile.txt | A string that has been formatted to display all the employee’s information in the GUI text output area. | Y |
| 2 | Reads in the .txt file: employeeReadFile.txt  Create Random Employees: 10 | 10 randomly created employees, that will be added to the current list of employees. All the employees will be display in the text area box of the GUI. | Y |
| 3 | Reads in the .txt file: employeeReadFile.txt  Create Random Employees: 10  Writes to the .txt file: employeeWriteFile.txt | The full list of employees written into the selected file. The first line should contain the number of employees in the list. The following lines will include the information of each employee per line.  Example of how the file should look like:  15  Nabeel Hussain Manager 30.50 10 26 Male  Mark Smith Sales 15.25 5 35 Male  Donald Lee Engineer 25.56 6 46 Male  Alice Barnes Engineer 20.50 4 38 Female  Melanie Holland Secretary 8.50 1 29 Female  Etc… | Y |
| 4 | No File initially read in.  Create Random Employees: 5 | JOption Pane Show Message Dialog:  “You must read in a file of employees first, in order to create new random ones.” | Y |
| 5 | Press “Exit” Button | Close GUI Application | Y |

Screen shots of successful compilation and running for all test cases



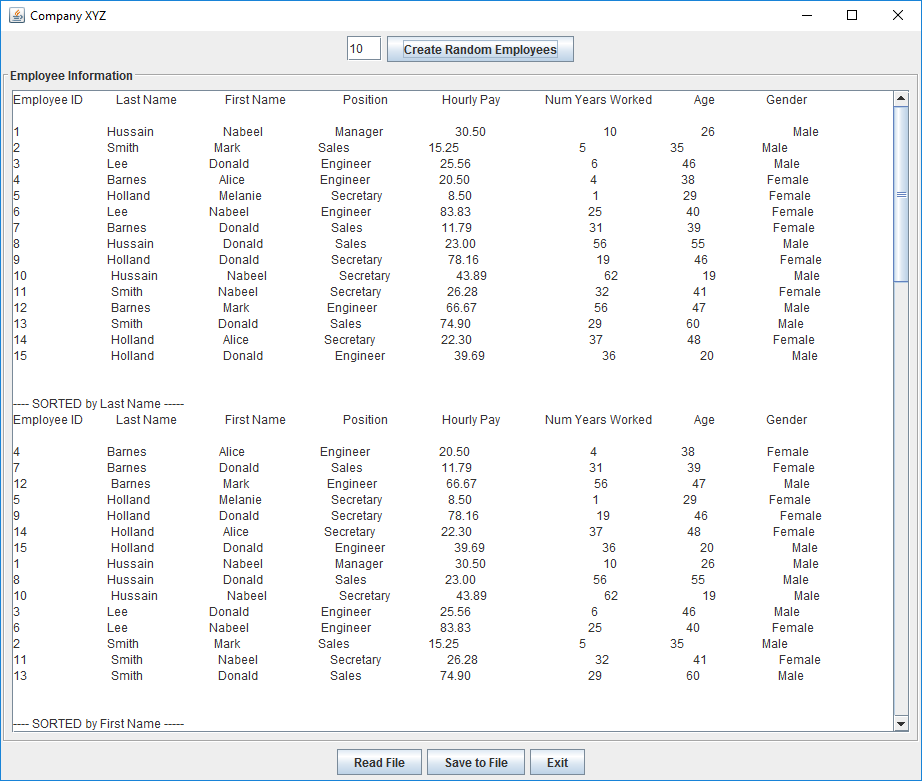


**Displays employee info from the file onto the text Area of the GUI, in various sorted orders.**



**Create 10 random employees, which will be added to the current list of employees:**

**The output will show a total of 15 employees now, which includes the 5 from the file read in, and the 10 random ones created.**



**Save the current list of employees by writing them into a selected file.**

