Dylan Pina

Lab #10

PSUEDO

4/20/2021

Employee Demo.java

```
// Creates 3 objects of the Employee class, these will act as
the employees //
Employee emp1 = new Employee(Name, IDnumber, Department, Job
title);
Employee emp2 = new Employee(Name, IDnumber, Department, Job
title);
Employee emp3 = new Employee(name, IDnumber, Department, Job
title);

// Initialize an arraylist to hold all of the Employee objects
//
ArrayList<Employee> empList = new ArrayList<Employee>();
// Add each employee object to the arraylist //
empList.add(emp1);
empList.add(emp2);
empList.add(emp3);

// Loop through the list of employees using a for each loop //
for(Employee e: empList)
```

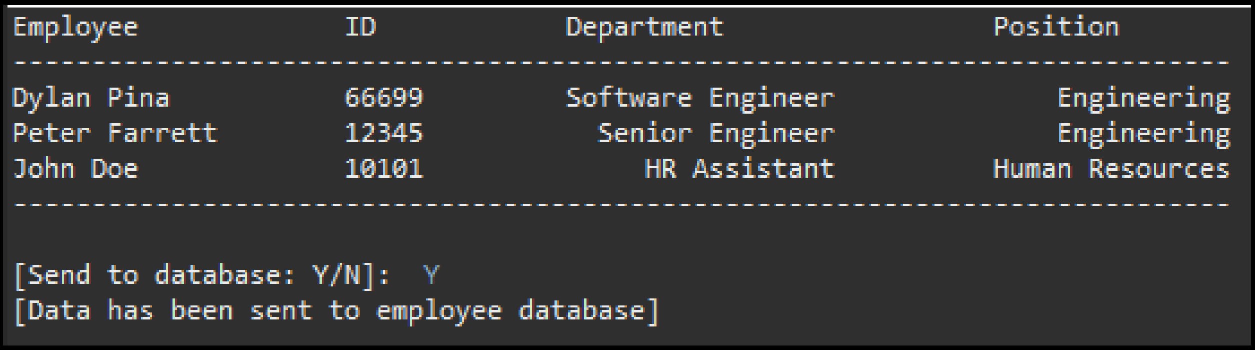
```
// Invoke the display info method on each Employee object which
displays every employee's information to the terminal //
e.displayInfo();

// Ask the user whether or not he/she wishes to send the
employee’s information to the employee database. Y will insist
yes, N will insist no //
Print([Send to database: Y/N])

Scanner input; // Create a scanner to hold the user’s input //

Char confirm = Scanner input; // Assign confirm to user input

If ( confirm == ‘Y’ ) // If the user has inputted ‘Y’ //
// Send all the information of each employee to the database //
sendToDataBase(empList)

Print([Information has been sent to employee database]);

 // If the user inputs any other character other than Y //
Print([Information has not been sent to employee database]);

input.close() // Close input scanner //
```

```
/** Sends employee’s information to the employee database */
public static  void sendToDataBase(ArrayList<Employee>
employees) throws IOException {
PrintWriter pw = null; // Initializes a PrintWriter //
FileOutputStream fo = null; // Initialize a FileOutPutStream //
File file = null; // Initialize a File //
// Create a new .txt file with the name “EmployeeDataBase //
file = new File("EmployeeDataBase.txt");
// Pass the file into the PrintWriter //
pw = new PrintWriter(new FileOutputStream(file));
// Pass the file into the FileOutputStream //
fo = new FileOutputStream(file);
// Formatting for table //
Print(“Employee”, “ID”, “Department”, “Position”);
Print(newline, “----...”);
// Assign list to the size of the arraylist of employees //
int list = employees.size();
// Loop through the arraylist of Employee objects //
for (int i = 0; i < list; i++)
// Write each line to the file, followed by a space in between
each line //
pw.write(employees.get(i).toString() + "\n");
// After the try scope has been executed //

// Write to the file(EmployeeDataBase.txt) //

pw.write("------------------...”);

// Close PrintWriter //

pw.close();

// Close the FileOutputStream //

fo.close();

}
```

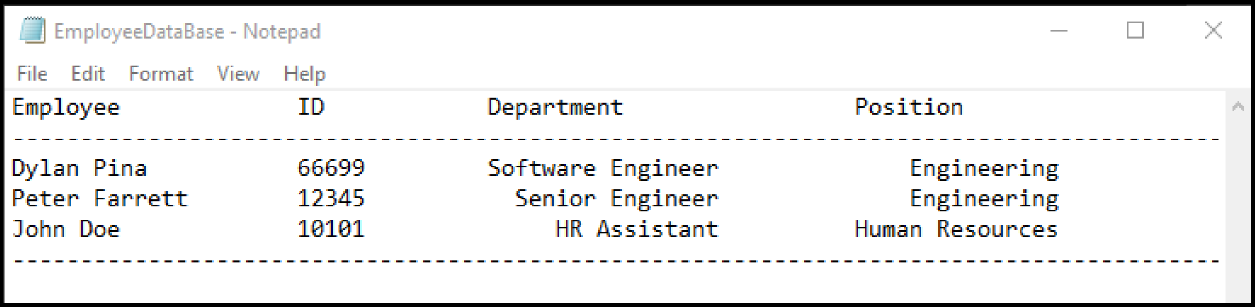
Employee.java

```
// Initialize a variable to store the Employee's name //

String name;

// Initialize a variable to store the Employee's ID number //

int idNumber;

// Initialize a variable to store the Employee's department //

String department;

// Initialize a variable to store the Employee's posistion/job
title //

String position;


/** This constructor allows all the information for the employee
to be set by passing in the employee's information as arguments
when a new instance object is created. */

Employee(String name, int idNumber, String department, String
position){

// Sets the employee's name //

this.name = name;

// Sets the employee's idNumber //

this.idNumber = idNumber;
// Sets the employee's department //
this.department = department;
// Sets the employee's position //
this.position = position;
}

/** @Overrides toString to format the employee's name, ID
number, posistion and department in a table format */
public String toString() {
return String.format("%s \t %9s \t %19s \t %20s",this.name,
this.idNumber, this.position, this.department);
}

/** Displays the employee's information in a table format*/
public void displayInfo() {System.out.println(String.format("%s
\t %9s \t %19s \t %20s", this.name, this.idNumber,
this.position, this.department));
}
```

Sample Output



```
Employee             ID            Department                 Position
-----------------------------------------------------------------------------
Dylan Pina           66699         Software Engineer              Engineering
Peter Farrett        12345           Senior Engineer              Engineering
John Doe             10101              HR Assistant          Human Resources
-----------------------------------------------------------------------------

[Send to database: Y/N]:  Y
[Data has been sent to employee database]
```



```
Employee             ID            Department                 Position
-----------------------------------------------------------------------------------------
Dylan Pina           66699         Software Engineer              Engineering
Peter Farrett        12345           Senior Engineer              Engineering
John Doe             10101              HR Assistant          Human Resources
-----------------------------------------------------------------------------------------
```