Assignment

Overview

The "We Really Suck" Vacuum company is looking for a system to manage their employees. You have been hired as part of a team to create the system that holds information about the employees for the company. The main system that runs has been created for us, we need to create the class that holds all the Employee information.

Directions

To Get Started:

1. Create a project in your IntelliJ (or your preferred IDE) titled **YOURNAME-Assignment7**
2. Download and import into your project these files: **A7EmployeeChecker.java & EmployeeSystem.java**
3. Open up the A7EmployeeChecker.java file and review the code. This is the file that creates some base employees and does some quick checks of the methods on Employee. This will tell you if you coded the Employee file correct. Also take a look at the EmployeeSystem.java file and see how more of a presentation file could run.

Coding the file:

Create a new class called Employee.java. This will be our class that holds the information about each employee. Review the UML diagram below as it will help you with the layout for the file. I have also given notes for specifics on what some of the methods require.

Employee.java
-firstName : String -lastName : String -employeeNum : int -department : String -jobTitle : String -hoursWorked : double -payRate : double -currency : NumberFormat
<<constructor>>Employee(String fn, String ln, int en, String dept, String job, double pr) <<constructor>>Employee(String fn, String ln, int en) <<constructor>>Employee(Employee e) <<constructor>>Employee() +getFirstName() : String +setFirstName(String fn) : void +getLastName() : String +setLastName(String ln) : void +getEmployeeNumber() : int +setEmployeeNumber(int en) : void +getDepartment() : String +setDepartment(String dept) : void +getJobTitle() : String +setJobTitle(String job) : void +getHoursWorked() : double +addHours() : void +addHours(double h) : void +resetHours() : void +getPayRate() : double +setPayRate(double pr) : void +calculateWeeklyPay() : double +printEmployee() : void

Notes on Methods/Variables:

Static Variable:

See that currency is a NumberFormat static variable and should be setup with the getCurrencyInstance() at the variable declaration NOT in the constructors

Constructors

Each constructor should setup hoursWorked to 0.0

The second constructor should set variables to blank if they are not sent, ie department, jobTitle, etc.

The copy constructor takes on the employee and should use the get methods from the send employee€ and set up a new employee.

The default constructor should just set everything up to blank.

Getters & Setters

These should be basic enough that they are setting and returning values. Note that I grouped them together for set/get of each variable. You can code them in any order. There also isn't a setter for hoursWorked variable as we have addHours() and resetHours() that will do the job for us.

addHours() & addHours(double h)

These are overloaded methods. The addHours() should just add one (1) hour to the hoursWorked variable.

The addHours(double h) should add the variable h to hoursWorked variable. BUT we shouldn't add negative hours, so make sure you handle if the number is negative or not.

calculateWeeklyPay()

This method should take the payRate and multiply it by the hoursWorked. Then return that value.

resetHours()

This method should set the hoursWorked to 0.

printEmployee()

This should print the employee object out with headers and the variables. See the format below, where each variable is in square brackets

Name: [firstName] [lastName]
ID: [employeeNum]
Department: [department]
Title: [jobTitle]
Pay: [payRate]
Hours Worked: [hoursWorked]

Output of A7EmployeeChecker.java

```
*****  Assignment 7 Employee Checker *****
Name: Steve Rodgers
ID: 3781
Department: Sales
Title: Manager
Pay: $28.50
Hours Worked: 0.0

Name: Clint Barton
ID: 6847
Department: Sales
Title: Customer Representative
Pay: $15.34
Hours Worked: 0.0

Name: Tony Stark
ID: 5749
Department: Service
Title: Lead Service Manager
Pay: $32.85
Hours Worked: 0.0


Employee Pay:
Steve : $142.50
Clint : $184.08
Tony : $32.85


New Employee:
Name: Thor Odinson
ID: 8623
Department: Service
Title: Lead Service Manager
Pay: $24.36
Hours Worked: 18.0
```

```


Last employee print:
Name: Steve Rodgers
ID: 3781
Department: Sales
Title: Manager
Pay: $28.50
Hours Worked: 0.0

Name: Clint Barton
ID: 6847
Department: Sales
Title: Customer Representative
Pay: $15.34
Hours Worked: 0.0

Name: Tony Stark
ID: 5749
Department: Service
Title: Lead Service Manager
Pay: $32.85
Hours Worked: 0.0

Name: Thor Odinson
ID: 8623
Department: Service
Title: Lead Service Manager
Pay: $24.36
Hours Worked: 0.0
```

Submission

1. Compress the IntelliJ project folder and submit it to this assignment.

   to compress: On Windows, right click -> send to -> compressed .zip file

   on Mac, right-click -> Compress

Hints/Tips (Before Submitting):

- *Don't forget to have a header at the top of your file and include a Resource statement.*
- *Use comments as needed. Include comments for methods and what is going on in each method.*
- *Follow all Java Styling Guides as covered*
- *Your output SHOULD MATCH the A7EmployeeChecker output exactly.*
- *Ask me questions early and often as needed, this is a little more challenging of a program to make.*