Object-Oriented Programming

Assignment 2

You are tasked with creating a Java program to manage the employees of a company.

The company has three types of employees:

- Full-time employees work 40 hours per week and receive a base salary of $50,000 per year. They also receive a bonus based on their performance rating, which is a number between 1 and 10. The bonus is calculated as 5% of their base salary multiplied by their performance rating, with a minimum bonus of $500 and a maximum bonus of $10,000.
- Part-time employees work 20 hours per week and receive a base salary of $25,000 per year. They also receive a bonus based on their performance rating, which is a number between 1 and 10. The bonus is calculated as 5% of their base salary multiplied by their performance rating, with a minimum bonus of $250 and a maximum bonus of $2,500.
- Contract workers receive a fixed rate of $50 per hour worked. They do not receive any performance-based bonuses.

You should create the following classes and interfaces:

1. **Employee** with the following properties and methods:
   - A private field named name (String)
   - A private field named id (int)
   - A private field named salary (double)
   - A private field named performanceRating (int)
   - A public method named double calculateSalary() - calculates the salary of an employee, including any bonuses they may receive.
   - A public method named int getID() - returns the ID number of the employee.
   - A public method named String getName() - returns the name of the employee.
   - A public method named int getPerformanceRating() - returns the performance rating of the employee.
   - A public method named void setPerformanceRating(int rating) - sets the performance rating of the employee.
   - Employee class must implement PerformanceManipulator interface, you should recalculate the employee's salary based on the new performance rating

2. **FullTimeEmployee** with the following properties and methods:
   - A private field named baseSalary (double)
   - A public method named double calculateSalary() - returns the base salary plus bonus based on performance rating.

3. **PartTimeEmployee** with the following properties and methods:
   - A private field named baseSalary (double)
   - A public method named double calculateSalary() - returns the base salary plus bonus based on performance rating.

4. **ContractWorker** with the following properties and methods:
   - A private field named hourlyRate (double)
   - A private field named hoursWorked (double)
   - A public method named double calculateSalary() - returns the hourly rate multiplied by the hours worked.
   - A public method named PerformanceManipulator Interface

5. **The PerformanceManipulator interface should have the following method:**
   - void increasePerformanceRating() - increases employee performance rating by one.

6. **Company class**

Create a Company class that manages the employees of the company, with the following properties and methods:

- A private field named employees (ArrayList<Employee>) - stores the employees of the company.
- A public method named void addEmployee(Employee employee) - adds an employee to the employees list.
- A public method named void removeEmployee(int id) - removes an employee from the employees list given their ID number.
- A public method named Employee findEmployeeByID(int id) - returns the employee with the given ID number or null if not found.
- A public method named ArrayList<Employee> getEmployeesSortedByName() - returns a list of all employees sorted by name.
- A public method named double calculateTotalSalary() - calculates the total salary of all employees in the company.
- A public method named double calculateAverageSalary(String type) - calculates the average salary of the specified employee type (full-time, part-time, or contract).

Implementation Notes

- You should use an ArrayList to store the employees in the Company class.
- Validate input to ensure that it is of the correct data type and within the required range.

Example Output

Here is an example of what the output of the program might look like:

```
Welcome to the Employee Management System
1. Add an employee
2. Remove an employee
3. Find an employee by ID
4. Print a list of all employees sorted by name
5. Calculate the total salary of all employees in the company
6. Calculate the average salary of full-time employees
7. Calculate the average salary of part-time employees
8. Calculate the total cost of contract workers
9. Increase employee performance rating
0. Quit
Enter your choice: 1
Select an employee type:
1. Full-time
2. Part-time
3. Contract worker
Enter your choice: 1
Enter employee name: John Smith
Enter employee ID: 1001
Enter employee performance rating: 8
Employee added successfully.
------------------------------------------
1. Add an employee
2. Remove an employee
3. Find an employee by ID
```

```
4. Print a list of all employees sorted by name
5. Calculate the total salary of all employees in the company
6. Calculate the average salary of full-time employees
7. Calculate the average salary of part-time employees
8. Calculate the total cost of contract workers
9. Increase employee performance rating
0. Quit
Enter your choice: 9
Enter employee ID: 1001
Employee performance rating increased by 1.
New employee information:
- John Smith (ID: 1001, Salary: $59,400.00, Performance Rating: 9)
------------------------------------------
1. Add an employee
2. Remove an employee
3. Find an employee by ID
4. Print a list of all employees sorted by name
5. Calculate the total salary of all employees in the company
6. Calculate the average salary of full-time employees
7. Calculate the average salary of part-time employees
8. Calculate the total cost of contract workers
9. Increase employee performance rating
0. Quit
Enter your choice: 0
Goodbye!
```