2003 AP Computer Science A Question 2, Java

Periodically, a company processes the retirement of some of its employees. In this question, you will write methods to help the company determine whether an employee is eligible to retire and to process the retirement of all eligible employees.

The Java interface Employee is implemented by objects that represent employees. The interface is declared as follows.

 public interface Employee

 {

 public int getAge(); // returns age of employee

 public int getYearsOnJob(); // returns number of years on job

 public double getSalary(); // returns salary in dollars

 public int getID(); // returns unique employee ID number

 }

The Company class is declared as follows.

public class Company

{

 // minimum age, years on job, and salary needed to retire

 private final static int RETIRE\_AGE = 65;

 private final static int RETIRE\_YEARS = 30;

 private final static double RETIRE\_SALARY = 10000.0;

 private ArrayList<Employee> myEmployees; // list of employees

 private double myTotalSalary; // total salary of all employees

 // ... constructor and other methods not shown

 // postcondition: returns true if emp is eligible to retire;

 // otherwise, returns false

 private boolean employeeIsEligible(Employee emp)

 {

 // you will write this function

 }

 // postcondition: all retirement-eligible employees have been

 // removed from myEmployees; myEmployees remains

 // sorted by employee ID; myTotalSalary has been

 // updated to maintain invariant that it represents

 // total of all employee salaries

 public void processRetirements()

 {

 // you will write this function

 }

}

Two class invariants must be maintained by Company objects:

The instance variable myEmployees is sorted by employee ID.

The instance variable myTotalSalary is the total of all employee salaries.

The Company constructor will establish these invariants as true initially. All Company methods must ensure that the invariants remain true.

Part A

An employee is eligible for retirement if (s)he meets at least two of the following requirements.

The employee is at least RETIRE\_AGE years old.

The employee has worked for at least RETIRE\_YEARS years.

The employee's salary is at least RETIRE\_SALARY.

Write the private Company method employeeIsEligible, which is described as follows. Method employeeIsEligible returns a boolean value that indicates whether the employee represented by parameter emp is eligible for retirement, using the rules above.

Complete method employeeIsEligible below.

 // postcondition: returns true if emp is eligible to retire;

 // otherwise, returns false

 private boolean employeeIsEligible(Employee emp)

 {

 }

Part B

( Assume all import statements you need are made. You do not need to write import statements).

Write the Company method processRetirements, which is described as follows. Method processRetirements removes all retirement-eligible emloyees from ArrayList myEmployees and maintains the two class invariants described above: the ArrayList is maintained in order by employee ID and myTotalSalary is the total of all salaries of the remaining employees.

In writing processRetirements, you may call method employeeIsEligible, specified in part (a). Assume that employeeIsEligible works as specified, regardless of what you wrote in part (a).

Complete method processRetirements below.

 // postcondition: all retirement-eligible employees have been

 // removed from myEmployees; myEmployees remains

 // sorted by employee ID; myTotalSalary has been

 // updated to maintain invariant that it represents

 // total of all employee salaries

 public void processRetirements()

 {

 }