**Assignment 1 Java2**

Student name: Thai Nguyen

Student ID: 991720991

\*Class Employee

package ngquocth;  
  
public class Employee {  
 private String id = "100";  
 private String name = "New Employee";  
 private int NumOfYears = 1;  
 private double MonthlySalary = 2000;  
 private boolean Manager = true;  
  
 //Constructors  
 public Employee(){  
  
 }  
 public Employee(String id, String name, int NumOfYears){  
 this.id = id;  
 this.name = name;  
 this.NumOfYears = NumOfYears;  
 }  
  
 public Employee(String id, String name, int NumOfYears, double MonthlySalary, boolean Manager){  
 this.id = id;  
 this.name = name;  
 this.NumOfYears = NumOfYears;  
 this.MonthlySalary = MonthlySalary;  
 this.Manager = Manager;  
 }  
  
  
 //set+get methods  
 public String getId(){  
 return this.id;  
 }  
 public void setId(String id){  
 if (Integer.*parseInt*(id) > 100)  
 this.id = id;  
 else {  
 throw new IllegalArgumentException("Invalid ID, try again!");  
 }  
 }  
 public String getName(){  
 return this.name;  
 }  
 public void setName(String name){  
 this.name = name;  
 }  
 public int getNumOfYears(){  
 return this.NumOfYears;  
 }  
 public void setNumOfYears(int NumOfYears){  
 if (NumOfYears >= 0) {  
 this.NumOfYears = NumOfYears;  
 } else {  
 throw new IllegalArgumentException("Number of years cannot be negative.");  
 }  
 }  
 public double getMonthlySalary(){  
 return this.MonthlySalary;  
 }  
 public void setMonthlySalary(double MonthlySalary){  
 if (MonthlySalary >= 2000) {  
 this.MonthlySalary = MonthlySalary;  
 } else {  
 throw new IllegalArgumentException("INvalid Monthly salary.");  
 }  
 }  
 public boolean getManager(){  
 return this.Manager;  
 }  
 public void setManager(boolean Manager){  
 this.Manager = Manager;  
 }  
  
 //Other methods  
 public double calTotal(int NumOfYears, double MonthlySalary){  
 double total\_salary = NumOfYears \* MonthlySalary;  
 return total\_salary;  
 }  
  
 public double calBonus(boolean Manager, double MonthlySalary){  
 if (Manager){  
 return MonthlySalary\*5;  
 }  
 else  
 return 0;  
 }  
  
 public String toString(){  
 return "Employee " +getName()+" with ID "+getId()+" has: \n" +  
 "Monthly Salary: " + getMonthlySalary()+ "\n" +  
 "Num of Years: " + getNumOfYears() +"\n"+  
 "Total Salary: " + calTotal(getNumOfYears(), getMonthlySalary()) + "\n"+  
 "Bonus: " + calBonus(getManager(), getMonthlySalary()) + "\n"+  
 "Manager: " + getManager();  
  
 }  
  
  
  
}

\*Main

package ngquocth;  
  
import java.util.Scanner;  
  
public class Main {  
 public static void main (String [] args){  
 Scanner input = new Scanner(System.*in*);  
  
 //Prompt user for input data  
 System.*out*.println("Enter an employee name: ");  
 String name = input.nextLine();  
 System.*out*.println("Enter employee ID: ");  
 String id = input.nextLine();  
 System.*out*.println("Enter number of years: ");  
 int NumOfYears = input.nextInt();  
 System.*out*.println("Enter monthly salary $: ");  
 double MonthlySalary = input.nextDouble();  
 input.nextLine();  
  
  
 //prompt boolean for manager  
 System.*out*.println("Is this employee is a Manager? true/false");  
 boolean manager;  
 while (true){  
 String checkManager = input.nextLine();  
 if (checkManager.equalsIgnoreCase("true")){  
 manager = true;  
 break;  
 } else if (checkManager.equalsIgnoreCase("false")) {  
 manager = false;  
 break;  
 }  
 else  
 System.*out*.println("Invalid input. Please enter \"true\" or \"false\" ");  
 }  
  
 //first employee  
 Employee e = new Employee();  
 e.setName(name);  
 e.setId(id);  
 e.setNumOfYears(NumOfYears);  
 e.setMonthlySalary(MonthlySalary);  
 e.setManager(manager);  
  
 //2 left employees with another setup way  
 Employee e2 = new Employee("102", "Jack", 2, 3000, false);  
 Employee e3 = new Employee("205", "Diana", 6, 7000, true);  
  
 System.*out*.println(e);  
 System.*out*.println();  
 System.*out*.println(e2);  
 System.*out*.println();  
 System.*out*.println(e3);  
  
  
  
  
 }  
}

\*Output

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

\*Hierarchy

A screenshot of a computer

Description automatically generated with medium confidence