Employee Payroll System

Overview

This project implements an employee payroll system using object-oriented programming principles in Java. It includes a base class Employee and three derived classes (HourlyEmployee, SalariedEmployee, and ExecutiveEmployee) representing different employee types.

Base Class: Employee

The Employee class serves as the base class with the following attributes:

- employeeId (int): unique identification number for the employee
- employeeName (String): full name of the employee
- designation (String): job title or role within the organization

Methods in Employee class:

1) Constructor (Employee(int, String, String))
   Initializes the attributes of the employee.

2) Accessor and Mutator Methods

- getEmployeeId(), setEmployeeId(int): Accessor and mutator for employeeId.
- getEmployeeName(), setEmployeeName(String): Accessor and mutator for employeeName.
- getDesignation(), setDesignation(String): Accessor and mutator for designation.

3) Abstract Methods

- calculateWeeklySalary(): Abstract method to be implemented by derived classes.
- displayEmployeeDetails(): Abstract method to display employee details.
- calculateBonus(): Abstract method to calculate bonus.

4) Concrete Method (calculateAnnualEarnings())

- Calculates the annual earnings by combining weekly salary and bonus.

Derived Class: HourlyEmployee

The HourlyEmployee class extends the Employee class, representing employees paid on an hourly basis.

Additional Attributes:

- hourlyRate (double): compensation per hour.

- hoursWorked (int): total hours worked in a week.

Methods in HourlyEmployee class:
1)Constructor (HourlyEmployee(int, String, String, double, int))
- Initializes attributes specific to hourly employees.

2)Accessor and Mutator Methods
- Accessor and mutator methods for hourlyRate and hoursWorked.

3)Overridden Methods

- calculateWeeklySalary(): Calculates the weekly salary based on hourly rate and hours worked.
- displayEmployeeDetails(): Displays detailed information about the hourly employee.
- calculateBonus(): Overrides the base class method to return 0 since hourly employees do not receive a bonus.

Derived Class: SalariedEmployee
The SalariedEmployee class extends the Employee class, representing employees with a fixed monthly salary.

Additional Attribute:
monthlySalary (double): fixed monthly compensation.

Methods in SalariedEmployee class:

1)Constructor (SalariedEmployee(int, String, String, double))
- Initializes attributes specific to salaried employees.

2)Accessor and Mutator Methods
- Accessor and mutator method for monthlySalary.

3)Overridden Methods
- calculateWeeklySalary(): Calculates the weekly salary based on the monthly salary.
- displayEmployeeDetails(): Displays detailed information about the salaried employee.
- calculateBonus(): Overrides the base class method to return 10% of the monthly salary as a bonus.

Derived Class: ExecutiveEmployee
The ExecutiveEmployee class extends the SalariedEmployee class, representing executive-level employees.

Additional Attribute:
bonusPercentage (double): the percentage of the annual salary allocated as a bonus.

Methods in ExecutiveEmployee class:
1)Constructor (ExecutiveEmployee(int, String, String, double, double))

- Initializes attributes specific to executive employees.

2)Accessor and Mutator Methods

- Accessor and mutator method for bonusPercentage.

3)Overridden Methods

- calculateBonus(): Overrides the method in the base class to calculate the bonus based on the bonus percentage.
- calculateWeeklySalary(): Overrides the method in the base class to adjust the weekly salary considering the bonus.

Main Class for Testing
The Main class contains the main method for testing the functionality of the employee payroll system.

Test Program

- Instantiates objects of each employee type.
- Sets attributes such as employeeId, employeeName, designation, hourlyRate, hoursWorked, monthlySalary, and bonusPercentage.
- Displays detailed information for each employee, including calculated weekly salary, bonus, annual earnings, and total payroll.
- Utilizes the super keyword to invoke methods from the base class.

Employee Payroll System - Test Program

Overview
The PayrollTest class serves as a comprehensive test program for the Employee Payroll System implemented in Java. This program instantiates objects of different employee types (HourlyEmployee, SalariedEmployee, and ExecutiveEmployee), sets their attributes, and showcases various functionalities such as calculating weekly salary, bonus, annual earnings, and total payroll.

Test Scenarios
1. Test HourlyEmployee
Instantiate an object of HourlyEmployee with the following attributes:

employeeId: 1
employeeName: "prabu"
designation: "Hourly Worker"
hourlyRate: 15.0
hoursWorked: 40

Output:

- Display detailed employee information.
- Print the calculated weekly salary, bonus, and annual earnings.
- Display a newline for separation.

2. Test SalariedEmployee
Instantiate an object of SalariedEmployee with the following attributes:

employeeId: 2
employeeName: "senthilnathan"
designation: "Manager"
monthlySalary: 6000.0

Output:

- Display detailed employee information.
- Print the calculated weekly salary, bonus, and annual earnings.
- Display a newline for separation.

3. Test ExecutiveEmployee
Instantiate an object of ExecutiveEmployee with the following attributes:

employeeId: 3
employeeName: "helen"
designation: "Executive"
monthlySalary: 10000.0
bonusPercentage: 0.15

Output:

- Display detailed employee information.
- Print the calculated weekly salary, bonus, and annual earnings.
- Display a newline for separation.

4. Display Total Payroll
Invoke the displayTotalPayroll method to calculate and display the total payroll for all employees.

Output:

- Print the total payroll for all employees.

Usage
Execute the main method in the PayrollTest class to run the test scenarios. Review the output to ensure that each employee type behaves as expected.

Observations

Inheritance: Observe the use of inheritance, where each employee type extends the Employee base class, inheriting common attributes and methods.

Method Invocation: Notice how methods from the base class are invoked using the super keyword, ensuring seamless access to shared functionalities.

Calculation Formulas: Understand the formulas used for calculating weekly salary, bonus, and annual earnings. Consider how these formulas vary based on the employee type.