interface EmployeeManagement<T, U, V, W> {

void EmployeeDetails(T employeeId, U employeeName, V isFullTime, W salary);

}

interface Calculate<W> {

double calculateBonus(W salary);

double calculateTax(W salary);

void updateSalary(W newSalary);

}

class Employee implements EmployeeManagement<Integer, String, Boolean, Double>, Calculate<Double> {

private Integer employeeId;

private String employeeName;

private Boolean isFullTime;

private Double salary;

public void EmployeeDetails(Integer employeeId, String employeeName, Boolean isFullTime, Double salary) {

this.employeeId = employeeId;

this.employeeName = employeeName;

this.isFullTime = isFullTime;

this.salary = salary;

}

public void disp() {

System.out.println("Employee ID: " + employeeId);

System.out.println("Employee Name: " + employeeName);

System.out.println("Is Full Time: " + isFullTime);

System.out.println("Salary: " + salary);

System.out.println("Bonus: " + calculateBonus(salary));

System.out.println("Tax: " + calculateTax(salary));

}

public double calculateBonus(Double salary) {

return 0.1 \* salary;

}

public double calculateTax(Double salary) {

return 0.15 \* salary;

}

public void updateSalary(Double newSalary) {

this.salary = newSalary;

System.out.println("Salary updated to: " + newSalary);

}

}

public class Lab6 {

public static void main(String[] args) {

Employee employee = new Employee();

employee.EmployeeDetails(1, "John Doe", true, 50000.0);

employee.disp();

employee.updateSalary(60000.0);

employee.disp();

}

}

