1 定义一个接口，接口内有个方法，判断某个员工是否合格，合格返回真。

public interface Qualifiable {

boolean isQualified();

}

2 有一个抽象员工类，有姓名属性，一个抽象方法统计员工薪水

public abstract class Employee {

protected String name;

public Employee(String name) {

this.name = name;

}

public abstract double calculateSalary();

}

3 有一个教师类，继承员工类与接口，教师类有每年科研积分，职务薪水与基本薪水，重写toString方法输出教师所有属性的值；重写员工薪水方法计算教师的总薪水（基本薪水加职务薪水)，同时科研积分大于100分表示该教师合格。

public class Teacher extends Employee implements Qualifiable {

private double researchPoints;

private double baseSalary;

private double positionSalary;

public Teacher(String name, double researchPoints, double baseSalary, double positionSalary) {

super(name);

this.researchPoints = researchPoints;

this.baseSalary = baseSalary;

this.positionSalary = positionSalary;

}

@Override

public boolean isQualified() {

return researchPoints > 100;

}

@Override

public double calculateSalary() {

return baseSalary + positionSalary;

}

@Override

public String toString() {

return "Teacher [Name: " + name + ", Research Points: " + researchPoints +

", Base Salary: " + baseSalary + ", Position Salary: " + positionSalary +

", Total Salary: " + calculateSalary() + "]";

}

}

4 定义一个销售员类，他继承了员工类，销售员有销售数量、单价，重写toString方法输出销售员所有属性的值；一个无参的构造函数和一个带有参数的构造函数,还有一个方法计算销售员的总薪水（销售数量乘单价)，同时销售数量乘单价大于5000表示合格。

public class Salesperson extends Employee {

private int salesQuantity;

private double unitPrice;

public Salesperson(String name, int salesQuantity, double unitPrice) {

super(name);

this.salesQuantity = salesQuantity;

this.unitPrice = unitPrice;

}

@Override

public double calculateSalary() {

return salesQuantity \* unitPrice;

}

@Override

public String toString() {

return "Salesperson [Name: " + name + ", Sales Quantity: " + salesQuantity +

", Unit Price: " + unitPrice + ", Total Salary: " + calculateSalary() + "]";

}

}

public class Main {

public static void main(String[] args) {

// 创建教师对象并验证

Teacher teacher = new Teacher("John Doe", 120, 50000, 20000);

System.out.println(teacher.toString());

System.out.println("Is Qualified: " + teacher.isQualified());

// 创建销售员对象并验证

Salesperson salesperson = new Salesperson("Jane Doe", 100, 50);

System.out.println(salesperson.toString());

System.out.println("Is Qualified: " + (salesperson.calculateSalary() > 5000));

}

}