JAVA網路應用Homework #3

工資103 H34996158 侯瑋明

//Q1 Exercise 10.8

//Fig. 10.6 HourlyEmployee.java

public class HourlyEmployee extends Employee implements Payable

{

private double wage;

private double hours;

public HourlyEmployee(String first, String last, String ssn, double hourlyWage, double hoursWorked)

{

super(first, last, ssn);

setWage(hourlyWage);

setHours(hoursWorked);

}

public void setWage(double hourlyWage)

{

wage = (hourlyWage < 0.0) ? 0.0 : hourlyWage;

}

public double getWage()

{

return wage;

}

public void setHours(double hoursWorked)

{

hours = ((hoursWorked >= 0.0) && (hoursWorked <= 168.0))? hoursWorked : 0.0;

}

public double getHours()

{

return hours;

}

@Override

public double getPaymentAmount() //修改earnings-> getPaymentAmount

{

if(getHours() <= 40)

return getWage() \* getHours();

else

return 40 \* getWage() + (getHours() -40) \* getWage() \* 1.5;

}

@Override

public String toString()

{

return String.format("hourly employee: %s\n%s: $%,.2f; %s: %,.2f", super.toString(), "hourly wage", getWage(), "hours worked", getHours());

}

}

//Fig. 10.7 CommissionEmployee.java

public class CommissionEmployee extends Employee implements Payable

{

private double grossSales;

private double commissionRate;

public CommissionEmployee(String first, String last, String ssn, double sales, double rate)

{

super(first, last, ssn);

setGrossSales(sales);

setCommissionRate(rate);

}

public void setCommissionRate(double rate)

{

commissionRate = (rate >0.0 && rate <1.0)? rate : 0.0;

}

public double getCommissionRate()

{

return commissionRate;

}

public void setGrossSales(double sales)

{

grossSales = (sales <0.0)? 0.0 : sales;

}

public double getGrossSales()

{

return grossSales;

}

@Override

public double getPaymentAmount() //修改earnings-> getPaymentAmount

{

return getCommissionRate() \* getGrossSales();

}

@Override

public String toString()

{

return String.format("%s: %s\n%s: $%,.2f; %s: %.2f",

"commission employee", super.toString(),

"gross sales", getGrossSales(),

"commission rate", getCommissionRate());

}

}

//Fig. 10.8 BasePlusCommissionEmployee.java

public class BasePlusCommissionEmployee extends CommissionEmployee implements Payable

{

private double baseSalary;

public BasePlusCommissionEmployee(String first, String last, String ssn, double sales, double rate, double salary)

{

super(first, last, ssn, sales, rate);

setBaseSalary(salary);

}

public void setBaseSalary(double salary)

{

baseSalary = (salary < 0.0)? 0.0 : salary;

}

public double getBaseSalary()

{

return baseSalary;

}

@Override

public double getPaymentAmount() //修改earnings-> getPaymentAmount

{

return getBaseSalary() + super.getPaymentAmount();

}

@Override

public String toString()

{

return String.format("%s %s; %s: $%,.2f",

"base-salaried", super.toString(),

"base salary", getBaseSalary());

}

}

//Fig. 10.15 PayableInterfaceTest.java

public class PayableInterfaceTest

{

public static void main(String[] args)

{

Payable[] payableObjects = new Payable[6];

payableObjects[0] = new Invoice("01234", "seat", 2, 375.00);

payableObjects[1] = new Invoice("56789", "tire", 4, 79.95);

payableObjects[2] = new SalariedEmployee("Jason", "Terry", "111-11-1111", 800.00);

payableObjects[3] = new HourlyEmployee("Chris", "Kaman", "222-22-2222", 103.00, 99.00);

payableObjects[4] = new CommissionEmployee("O.J.", "Mayo", "333-33-3333", 78.00, 0.8);

payableObjects[5] = new BasePlusCommissionEmployee("David", "Lee", "444-44-4444", 97.00, 0.92, 800.00\*1.1);

System.out.println("Invoices and Employees processed polymorphically:\n");

for(Payable currentPayable : payableObjects)

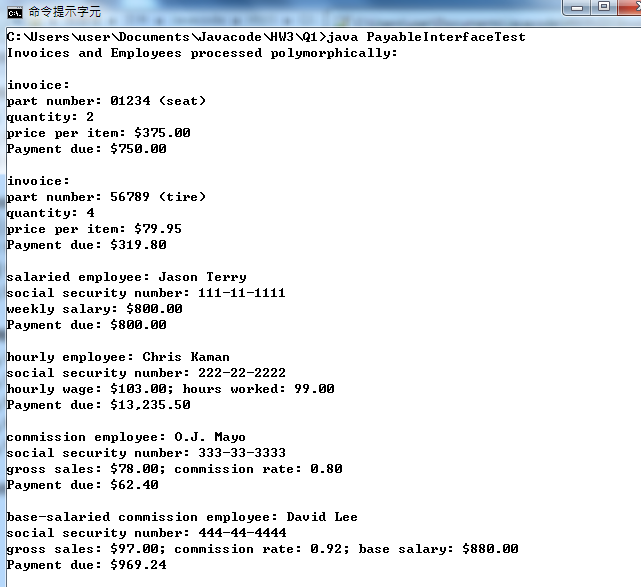
{

System.out.printf("%s \n%s: $%,.2f\n\n",

currentPayable.toString(),

"Payment due", currentPayable.getPaymentAmount());

}

 }

}

//Q2

// Message.java

public class Message extends Throwable

{

public Message(){}

public String toString()

{

return "體重不正常";

}

}

// BMIDetector.java

public class BMIDetector

{

double height = 1.7;

double weight = 100.0;

double bmi = 0.0;

public BMIDetector()

{

bmi = weight / (height \* height);

System.out.println("BMI: "+ bmi);

try{

MeasureHeightAndWeight();

} catch(Message m){

if(bmi>24 || bmi<18.5) //例外事件printout

{

System.out.println("Warning: "+ m);

}

}

}

public void MeasureHeightAndWeight() throws Message

{

throw new Message();

}

public static void main(String[] args)

{

new BMIDetector();

}

}