CT874

Assignment 7

H. Dip. Industry Stream

Hume, Tori (11486248)

Question 1

Code:

SUPERCLASS:

```java
/*Tori Hume
 * ID: 11486248
 * Assignment 7
 * Question 1
 * SalesEmployee SuperClass
 */

//import decimal method
import java.text.DecimalFormat;

//Create abstract class SalesEmployee
abstract public class SalesEmployee {
	//create new instance of decimal format
	DecimalFormat df = new DecimalFormat("€0.00");

	//declare variables
	private String firstName, lastName, ppsNumber;
	private  static int bikeEmployeeNumber=0;
	protected double sales, commission;
	private int employeeNumber;


	//create SalesEmployee constructor
	public SalesEmployee(){
		this.firstName= "Unknown";
		this.lastName= "Unknown";
		this.ppsNumber= "unknown";
		this.employeeNumber= ++bikeEmployeeNumber;//Increment employee number with each instance
	}

	//create new salesEmployee constructor with parameters
	public SalesEmployee(String firstName, String lastName, String ppsNumber) {
		this.firstName = firstName;
		this.lastName = lastName;
		this.ppsNumber = ppsNumber;
		this.employeeNumber= ++bikeEmployeeNumber;
		//Increment employee number with each instance
	}
	//getters and setters for each private variable,
	public void setFirstName( String fName){
		this.firstName= fName;
	}

	public void setLastName(String lName){
		this.lastName= lName;
	}

	public void setPPS(String PPSNo){
		this.ppsNumber= PPSNo;
	}

	public String getFistName(){
		return firstName;
	}

	public String getLastName(){
		return lastName;
```

```java
    }

    public String getPPS(){
        return ppsNumber;
    }

    public int getEmployeeNumber(){
        return employeeNumber;
    }


    //override toString method
    public String toString() {
    return "SalesEmployee [First Name= " + firstName + ", Last Name= " + lastName + ", PPS
    Number= " + ppsNumber + ", Sales= " + df.format(sales)+ ", Commission =" +
    df.format(commission )+ ", Employee Number= " + employeeNumber + "]";
    }

    //create abstract method calculateCommission
    abstract void calculateCommission();
}
```

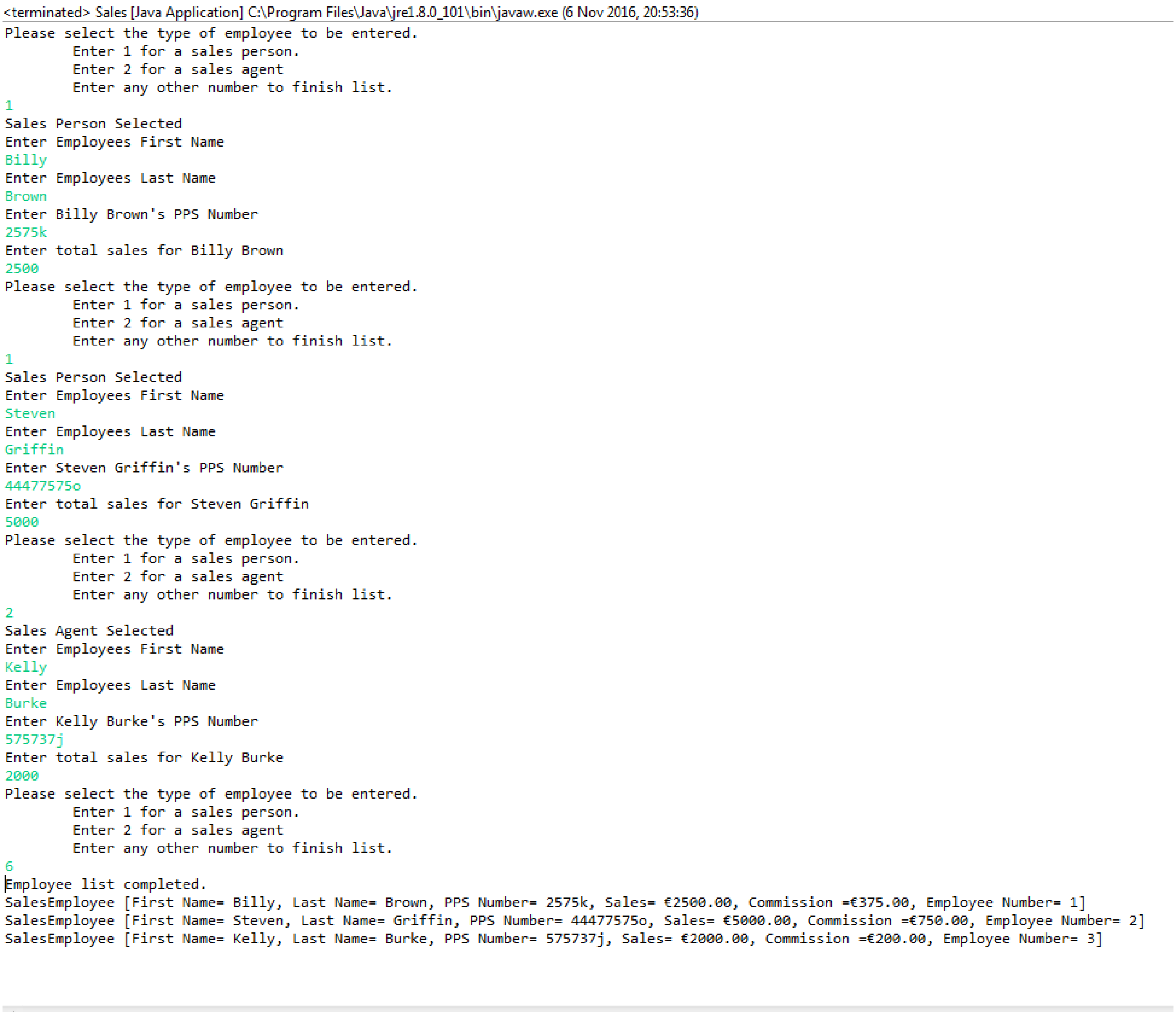
SUBCLASS':

```
/*Tori Hume
 * ID: 11486248
 * Assignment 7
 * SalesPerson Subclass of SalesEmployee
 */

//create subclass
public class SalesPerson extends SalesEmployee{
	//create constructor
	public SalesPerson() {
		super();
	}
	//Create constructor with parameters
	public SalesPerson(String firstName, String lastName, String ppsNumber){

		//pulls the constructor with parameters from the superclass SalesEmployee
		super(firstName, lastName, ppsNumber);
	}
	//overriding and implementing the calculateCommission method found in the Superclass
	public void calculateCommission(){
		commission= (sales)*0.15;
		//A Sales Person makes 15% commission on sales
	}
}


/*Tori Hume
 * ID: 11486248
 * Assignment 7
 * SalesAgent Subclass of SalesEmployee
 */

//create subclass
public class SalesAgent extends SalesEmployee{
	//create constructor
	public SalesAgent() {
		super();
	}
	//Create constructor with parameters
	public SalesAgent(String firstName, String lastName, String ppsNumber){
		//pulls the constructor with parameters from the superclass
		super();
	}

	//overriding and implementing the calculateCommission method found in the Superclass
	public void calculateCommission(){
		commission= (sales)*0.10;
		//Sales Agents make 10% commission on sales
	}
}
```

TESTER CLASS:

```
/*Tori Hume
 * ID: 11486248
 * Assignment 7
 * Tester class
 */

import java.util.Scanner; //Import Scanner
import java.util.List; //Import List
import java.util.ArrayList;      //Import ArrayList

public class Sales {


    public static void main(String[] args){
        //create new instance of scanner
        Scanner input = new Scanner(System.in);
        //Create a new ArrayList of populated with instances of SalesEmployee
        List<SalesEmployee> SalesList= new ArrayList<SalesEmployee>();

        //Declare and initialise variables of type integer
        int employeeType=0, i=0;

        //get user to select what type the employee is.
        System.out.println("Please select the type of employee to be entered. "
                     + "\n\tEnter 1 for a sales person."
                     + " \n\tEnter 2 for a sales agent"
                     + "\n\tEnter any other number to finish list.");
        //Assign next input to employeeType
        employeeType=input.nextInt();

        input.nextLine();

        //a while loop is used to insure the arraylist is only
        //filled when option 1 or 2 is selected
        while (employeeType==1 || employeeType==2){

            // using an if and else statement to implement the decision of
            //Sales person or Sales Agent

            if(employeeType==1){
                SalesList.add( new SalesPerson());
                System.out.print("Sales Person Selected\n");
            }
            else {
                SalesList.add( new SalesAgent());
                System.out.print("Sales Agent Selected\n");
            }


            //Use input & getters & setters to assign values entered in the
            //command window to variables
            System.out.println("Enter Employees First Name");
            String fn=input.nextLine();
            SalesList.get(i).setFirstName(fn);

            System.out.println("Enter Employees Last Name");
            String ln=input.nextLine();
            SalesList.get(i).setLastName(ln);

            System.out.println("Enter "+fn+" "+ln+"'s PPS Number");
            String ppsid=input.nextLine();
            SalesList.get(i).setPPS(ppsid);

            System.out.println("Enter total sales for "+fn+" "+ln);
            SalesList.get(i).sales=input.nextDouble();
            //Implement the calculateCommision method
            SalesList.get(i).calculateCommission();
            //Increment i
            i++;
            //clear the buffer
            input.nextLine();

            //gets user to select if they want to add to the list, and what type of
            employee,
            //or if they would like to close the list.
            System.out.println("Please select the type of employee to be entered. "
                    + "\n\tEnter 1 for a sales person."
                    + " \n\tEnter 2 for a sales agent"
                    + "\n\tEnter any other number to finish list.");
            employeeType=input.nextInt();

            input.nextLine();
        }
        //prints to list when while loop exited.
        System.out.println("Employee list completed.");
        input.close();

        //An enhanced for loop is used to iterate through the arraylist and
        //the overridden toSting method is used to print information to screen.
        for(SalesEmployee s: SalesList){
            //Print toString method to screen
            System.out.println(s.toString());
        }
    }

}
```

Screen Shot:



```
<terminated> Sales [Java Application] C:\Program Files\Java\jre1.8.0_101\bin\javaw.exe (6 Nov 2016, 20:53:36)
Please select the type of employee to be entered.
        Enter 1 for a sales person.
        Enter 2 for a sales agent
        Enter any other number to finish list.
1
Sales Person Selected
Enter Employees First Name
Billy
Enter Employees Last Name
Brown
Enter Billy Brown's PPS Number
2575k
Enter total sales for Billy Brown
2500
Please select the type of employee to be entered.
        Enter 1 for a sales person.
        Enter 2 for a sales agent
        Enter any other number to finish list.
1
Sales Person Selected
Enter Employees First Name
Steven
Enter Employees Last Name
Griffin
Enter Steven Griffin's PPS Number
44477575o
Enter total sales for Steven Griffin
5000
Please select the type of employee to be entered.
        Enter 1 for a sales person.
        Enter 2 for a sales agent
        Enter any other number to finish list.
2
Sales Agent Selected
Enter Employees First Name
Kelly
Enter Employees Last Name
Burke
Enter Kelly Burke's PPS Number
575737j
Enter total sales for Kelly Burke
2000
Please select the type of employee to be entered.
        Enter 1 for a sales person.
        Enter 2 for a sales agent
        Enter any other number to finish list.
6
Employee list completed.
SalesEmployee [First Name= Billy, Last Name= Brown, PPS Number= 2575k, Sales= €2500.00, Commission =€375.00, Employee Number= 1]
SalesEmployee [First Name= Steven, Last Name= Griffin, PPS Number= 44477575o, Sales= €5000.00, Commission =€750.00, Employee Number= 2]
SalesEmployee [First Name= Kelly, Last Name= Burke, PPS Number= 575737j, Sales= €2000.00, Commission =€200.00, Employee Number= 3]
```