

Answer on Question #59374, Programming & Computer Science, Java, JSP, JSF

Condition

The cashier class must have an ArrayList of fruits, a method called getTotalCost() will return the cost of all the fruit objects in the list, and getTotalTax() will return only the total taxes of the fruits in the list. You must use the constant TAX_RATE in the Consts.java file to get that rate and calculate the taxes properly. The constructor of this class must take the shop name. This shop name will be printed in the receipt eventually. Round Prices to nearest 2 digit decimal points in both of these methods. A getTotalTax() that results in 3.432 will not be accepted. Here, the number should be 3.43. You must also override toString() method and return a user-friendly receipt of all the items in the basket in this cashier.

Code

Shop.java

```
import java.util.ArrayList;

public class Shop {
    public static void main(String[] args) {
        ArrayList<Fruit> fruits = new ArrayList<Fruit>();

        Apple apple = new Apple(3);
        fruits.add(apple);

        Banana banana = new Banana(3, 12.5);
        fruits.add(banana);

        Lemon lemon = new Lemon(6, 1.5);
        fruits.add(lemon);

        OrangeLime orangeLime = new OrangeLime(4, 0.4, 5);
        fruits.add(orangeLime);

        Cashier cashier = new Cashier("Fruit shop", fruits);
        System.out.println(cashier);
    }
}
```

Fruit.java

```
public abstract class Fruit {
    public abstract double getCost();
}
```

Consts.java

```
public class Consts {
    public static final String PRICE_FORMAT = "#0.00";
    public static final double APPLE_PRICE = 24.0; // per dozen
    public static final int LEMONS_IN_A_JUICE = 6;
    public static final int TAX_RATE = 20;
}
```

Cashier.java

```
import java.text.DecimalFormat;
import java.util.ArrayList;

public class Cashier {
    private String shopName;
    private ArrayList<Fruit> fruits;

    public Cashier(String shop, ArrayList<Fruit> fruits) {
        this.shopName = shop;
        this.fruits = fruits;
    }

    public double getTotalCost() {
        double cost = 0;
        for (Fruit fruit : fruits) {
            cost = cost + fruit.getCost();
        }
        return cost;
    }

    public double getTotalTax() {
        return getTotalCost() * Consts.TAX_RATE / 100;
    }

    @Override
    public String toString() {
        return "Shop: " + shopName + "\n" +
            "1. " + fruits.get(0).getClass().getName() + ": " +
                new
DecimalFormat(Consts.PRICE_FORMAT).format(fruits.get(0).getCost()) + "$" +
"\n" +

            "2. " + fruits.get(1).getClass().getName() + ": " +
                new
DecimalFormat(Consts.PRICE_FORMAT).format(fruits.get(1).getCost()) + "$" +
"\n" +

            "3. " + fruits.get(2).getClass().getName() + ": " +
                new
DecimalFormat(Consts.PRICE_FORMAT).format(fruits.get(2).getCost()) + "$" +
"\n" +

            "4. " + fruits.get(3).getClass().getName() + ": " +
                new
DecimalFormat(Consts.PRICE_FORMAT).format(fruits.get(3).getCost()) + "$" +
"\n" +

            "Total cost: " + new
DecimalFormat(Consts.PRICE_FORMAT).format(getTotalCost()) + "$" + "\n" +
```

```

        "Total tax: " + new
DecimalFormat(Consts.PRICE_FORMAT).format(getTotalTax()) + "$";
    }
}

```

Apple.java

```

public class Apple extends Fruit {
    private int applesNumber;

    public Apple(int number) {
        this.applesNumber = number;
    }

    @Override
    public double getCost() {
        return (Consts.APPLE_PRICE / 12.0) * applesNumber;
    }
}

```

Banana.java

```

public class Banana extends Fruit {
    private double bananasWeight;
    private double bananaPrice;

    public Banana(double weight, double price) {
        this.bananasWeight = weight;
        this.bananaPrice = price;
    }

    @Override
    public double getCost() {
        return bananasWeight * bananaPrice;
    }
}

```

Lemon.java

```

public class Lemon extends Fruit {
    private int lemonsNumber;
    private double lemonPrice;

    public Lemon(int number, double price) {
        this.lemonsNumber = number;
        this.lemonPrice = price;
    }

    @Override
    public double getCost() {
        return lemonPrice * lemonsNumber;
    }
}

```

OrangeLime.java

```
public class OrangeLime extends Lemon {  
    private int numberOfJuices;  
    private double pricePerLemon;  
    private double laborCost;  
  
    public OrangeLime(int number, double price, double labor) {  
        super(number, price);  
        this.numberOfJuices = number;  
        this.pricePerLemon = price;  
        this.laborCost = labor;  
    }  
  
    @Override  
    public double getCost() {  
        return (numberOfJuices * Consts.LEMONS_IN_A_JUICE * pricePerLemon) +  
        laborCost;  
    }  
}
```

Output

Shop: Fruit shop

1. Apple: 6.00\$

2. Banana: 37.50\$

3. Lemon: 9.00\$

4. OrangeLime: 14.60\$

Total cost: 67.10

\$ Total tax: 13.42

\$