

Answer on Question #59443, Programming & Computer Science, Java, JSP, JSF

Condition

1-public double getTotalCost();

A method that returns the total cost of items without tax. The number has to be rounded to the nearest two decimal points.... @return The total cost without tax

2-public double getTotalTax();

A method that returns the total tax. The number has to be rounded to the nearest two decimal points..@return The total taxes of items

3-public void addItem(Fruit fruit);

Add a fruit to the cashier for checkout...@param fruit The fruit object to add

4-public void clear();

Clear all fruits from this cashier

5-public int getNumberOfItems();

Get the number of items in the current cashier box .. @return the total number of items in this cashier box

Code

Shop.java

```
import java.util.ArrayList;

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

        Apple apple = new Apple(3);
        Banana banana = new Banana(3, 12.5);
        Lemon lemon = new Lemon(6, 1.5);
        OrangeLime orangeLime = new OrangeLime(4, 0.4, 5);

        Cashier cashier = new Cashier("Fruit shop", fruits);
        cashier.addItem(apple);
        cashier.addItem(banana);
        cashier.addItem(lemon);
        cashier.addItem(orangeLime);
        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;
    }

    public void addItem(Fruit fruit) {
        fruits.add(fruit);
    }

    public int getNumberOfItems() {
        return fruits.size();
    }

    public void clear() {
        fruits.clear();
    }

    @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(Constants.PRICE_FORMAT).format(fruits.get(1).getCost()) + "$" +
"\n" +

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

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

                "-----" + "\n" +

                "Items: " + getNumberOfItems() + "\n" +

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

                "Total tax: " + new
DecimalFormat(Constants.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 (Constants.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\$

Items: 4

Total cost: 67.10\$

Total tax: 13.42\$