Answer on Question #59372, Programming & Computer Science, Java, JSP, JSF

Condition

4-OrangeLime: This class inherits from Lemon class that you just created above. This “fruit” is really a special addition of the fruit shop where they use some lemons to create special juice mixes. The constructor of this class will take the price per lemon (same as above for the Lemon class), how many juices, and the labor cost. Each juice will have a predefined number of lemons in it. Use the LEMONS_IN_A_JUICE constant in the Consts.java file to get that number. The getCost() method will return the cost of all lemons in the juices order plus a labor cost that was given in the constructor. The labor cost is for all the juices and not for each individual juice. So if there is an order of 4 juices, each juice takes 6 lemons, and each lemon costs 0.4 QR. The labor cost is 5 QR, then the cost is (4 * 6 * 0.4) + 5 QR which must be returned by the getCost() function.

Code

Task.java

```
public class Task {
   public static void main(String[] args) {
        Apple apple = new Apple(3);
        System.out.println("Apple: " + apple.getCost());

        Banana banana = new Banana(3, 12.5);
        System.out.println("Banana: " + banana.getCost());

        Lemon lemon = new Lemon(6, 1.5);
        System.out.println("Lemon: " + lemon.getCost());

        OrangeLime orangeLime = new OrangeLime(4, 0.4, 5);
        System.out.println("Orange lime: " + orangeLime.getCost());
   }
}
```

Fruit.java

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

Consts.java

```
public class Consts {
   public static final double APPLE_PRICE = 24.0; // per dozen
   public static final int LEMONS_IN_A_JUICE = 6;
}
```

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

Apple: 6.0

Banana: 37.5

Lemon: 9.0

Orange lime: 14.6