|  |
| --- |
| package io.github.picodotdev.blogbitix.kata01; |
|  |  |
|  | import java.math.BigDecimal; |
|  | import java.math.RoundingMode; |
|  |  |
|  | public class Main { |
|  |  |
|  | public interface Pricing { |
|  | BigDecimal calculate(BigDecimal quantity); |
|  | } |
|  |  |
|  | public interface Offer { |
|  | BigDecimal calculateWithOffer(BigDecimal quantity); |
|  |  |
|  | BigDecimal getNumberWithoutOffer(BigDecimal quantity); |
|  | BigDecimal getNumberWithOffer(BigDecimal quantity); |
|  | } |
|  |  |
|  | public static class Product { |
|  |  |
|  | Pricing pricing; |
|  |  |
|  | Product(Pricing pricing) { |
|  | this.pricing = pricing; |
|  | } |
|  |  |
|  | public BigDecimal calculate(BigDecimal quantity) { |
|  | return pricing.calculate(quantity); |
|  | } |
|  | } |
|  |  |
|  | public static class SimplePricing implements Pricing { |
|  |  |
|  | BigDecimal unitPrice; |
|  |  |
|  | SimplePricing(BigDecimal unitPrice) { |
|  | this.unitPrice = unitPrice; |
|  | } |
|  |  |
|  | @Override |
|  | public BigDecimal calculate(BigDecimal quantity) { |
|  | return unitPrice.multiply(quantity); |
|  | } |
|  | } |
|  |  |
|  | public static class WeightPricing implements Pricing { |
|  |  |
|  | BigDecimal weightPrice; |
|  |  |
|  | WeightPricing(BigDecimal weightPrice) { |
|  | this.weightPrice = weightPrice; |
|  | } |
|  |  |
|  | @Override |
|  | public BigDecimal calculate(BigDecimal weight) { |
|  | return weightPrice.multiply(weight); |
|  | } |
|  | } |
|  |  |
|  | public static class OfferPricing implements Pricing, Offer { |
|  |  |
|  | Pricing normalPricing; |
|  | BigDecimal offerQuantity; |
|  | BigDecimal offerPercent; |
|  |  |
|  | OfferPricing(Pricing normalPricing, BigDecimal offerQuantity, BigDecimal offerPercent) { |
|  | this.normalPricing = normalPricing; |
|  | this.offerQuantity = offerQuantity; |
|  | this.offerPercent = offerPercent; |
|  | } |
|  |  |
|  | public BigDecimal calculate(BigDecimal quantity) { |
|  | BigDecimal withoutOfferPrice = normalPricing.calculate(getNumberWithoutOffer(quantity)); |
|  | BigDecimal withOfferPrice = calculateWithOffer(getNumberWithOffer(quantity)); |
|  |  |
|  | return withoutOfferPrice.add(withOfferPrice); |
|  | } |
|  |  |
|  | public BigDecimal calculateWithOffer(BigDecimal quantity) { |
|  | return normalPricing.calculate(new BigDecimal("1")).multiply(quantity).multiply(getOfferPercent()); |
|  | } |
|  |  |
|  | public BigDecimal getNumberWithoutOffer(BigDecimal quantity) { |
|  | BigDecimal groups = quantity.divide(offerQuantity, 0, RoundingMode.DOWN); |
|  | return quantity.subtract(groups); |
|  | } |
|  |  |
|  | public BigDecimal getNumberWithOffer(BigDecimal quantity) { |
|  | return quantity.subtract(getNumberWithoutOffer(quantity)); |
|  | } |
|  |  |
|  | private BigDecimal getOfferPercent() { |
|  | return BigDecimal.valueOf(100).subtract(offerPercent).divide(new BigDecimal("100")); |
|  | } |
|  | } |
|  |  |
|  | public static void main(String[] args) { |
|  | Product p1 = new Product(new SimplePricing(new BigDecimal("2"))); |
|  | System.out.println(p1.calculate(new BigDecimal("3"))); |
|  |  |
|  | Product p2 = new Product(new WeightPricing(new BigDecimal("1.35"))); |
|  | System.out.println(p2.calculate(new BigDecimal("3"))); |
|  |  |
|  | Product p3 = new Product(new OfferPricing(new SimplePricing(new BigDecimal("1")), new BigDecimal("3"), new BigDecimal("50"))); |
|  | System.out.println(p3.calculate(new BigDecimal("6"))); |
|  | } |
|  | }   |  | | --- | | package io.github.picodotdev.blogbitix.kata01; | |  |  | |  | import io.github.picodotdev.blogbitix.kata01.Main.OfferPricing; | |  | import io.github.picodotdev.blogbitix.kata01.Main.Product; | |  | import io.github.picodotdev.blogbitix.kata01.Main.SimplePricing; | |  | import io.github.picodotdev.blogbitix.kata01.Main.WeightPricing; | |  |  | |  | import java.math.BigDecimal; | |  |  | |  | import org.junit.Assert; | |  | import org.junit.Test; | |  |  | |  | public class MainTest { | |  |  | |  | @Test | |  | public void simplePricing() { | |  | Product p1 = new Product(new SimplePricing(new BigDecimal("2"))); | |  | Assert.assertEquals(new BigDecimal("6"), p1.calculate(new BigDecimal("3"))); | |  | } | |  |  | |  | @Test | |  | public void weightPricing() { | |  | Product p2 = new Product(new WeightPricing(new BigDecimal("1.35"))); | |  | Assert.assertEquals(new BigDecimal("4.05"), p2.calculate(new BigDecimal("3"))); | |  | } | |  |  | |  | @Test | |  | public void offerPricing() { | |  | Product p3 = new Product(new OfferPricing(new SimplePricing(new BigDecimal("1")), new BigDecimal("3"), new BigDecimal("50"))); | |  | Assert.assertEquals(new BigDecimal("4.5"), p3.calculate(new BigDecimal("5"))); | |  | } | |  | } | |