CouponShoppingCartCaseStudy

CouponDecorator

```
1. abstract class
CouponDecorator
extends Product{}
```

Item1(extends Product)

```
 Item1(String name, double
originalPrice,ProductType type){
      super(name,originalPrice,type);
   }

   @Override
   public double getPrice() {
      return originalPrice;
   }
}
```

Item2(extends Product)

```

   Item2(String name, double originalPrice,
ProductType type){
      super(name,originalPrice,type);
   }

   @Override
   public double getPrice() {
      return originalPrice;
   }
}
```

Main

```
 public static void main(String[] args) {

      Product item1 = new
Item1("FAN",1000,ProductType.ELECTRONIC_GOODS);

      Product item2 = new
Item2("SOFA",2000,ProductType.FURNITURE_GODOS);

      ShoppingCart shoppingCart = new ShoppingCart();

      shoppingCart.addToCart(item1);

      shoppingCart.addToCart(item2);

      System.out.println("Total price is =
"+shoppingCart.getTotalPrice());
   }
}
```

PercentageCouponDecorator

```

public class PercentageCouponDecorator extends CouponDecorator{

   Product product ;

   int discountPercentage;

   PercentageCouponDecorator(Product product, int discountPercentage){

      this.product=product;

      this.discountPercentage=discountPercentage;

   }

   @Override
   public double getPrice() {

      double price = product.getPrice();

      return price- (price*discountPercentage)
   }
}
```

Product

```
package CouponShoppingCartCaseStudy;

public abstract class Product {

   String name;

   double originalPrice;

   ProductType productType;

   Product(){}
   Product(String name, double originalPrice, ProductType productType){

      this.name=name;
      this.originalPrice=originalPrice;
      this.productType=productType;
   }

   public abstract double getPrice();

   public ProductType getProductType() {
      return productType;
   }
}
```

TypeCouponDecorator

```
public class TypeCouponDecorator extends CouponDecorator{

   Product product ;
   int discountPercentage;
   ProductType productType;
   static List<ProductType> eligibleTypes = new ArrayList<>();

   static {
      eligibleTypes.add(ProductType.FURNITURE_GODOS);
      eligibleTypes.add(ProductType.ELECTRONIC_GOODS);
   }

   TypeCouponDecorator(Product product, int percent, ProductType productType){
      this.product=product;
      this.discountPercentage=percent;
      this.productType=productType;
   }

   @Override
   public double getPrice() {
      double price = product.getPrice();
      if(eligibleTypes.contains(productType)){
         return price- (price*discountPercentage)/100;
      }

      return price;
   }
}
```

ProductType - ENUM

```
public enum ProductType {
   ELECTRONIC_GOODS,
   FURNITURE_GODOS
}
```

ShoppingCart

```
public class ShoppingCart {

   List<Product> productList;

   ShoppingCart(){

      productList = new ArrayList<>();
   }

   public void addToCart(Product product){

      Product productWithEligibleDiscount =
            new TypeCouponDecorator(
                  new PercentageCouponDecorator(product,10),3,
product.getProductType()
            );

      productList.add(productWithEligibleDiscount);
   }

   public int getTotalPrice(){

      int totalPrice=0;
      for(Product product : productList){

         totalPrice+=product.getPrice();
      }

      return totalPrice;
   }
}
```