Command Pattern

The **Command Pattern** is a **behavioral pattern** which allows us to pass requests as objects. Requests are passed to an **invoker** which simply executes the command. Since the request is being passed as an object, we can easily parameterize the data. This is very useful for cases where the command is complicated and has a lot of information that must be passed.

public class Stock {  
 public void buy() {  
 // code  
 }  
 public void sell() {  
 // code  
 }  
}  
  
public interface *Order* {  
 void execute();  
}  
  
public class BuyStock implements *Order* {  
 private Stock stock;  
  
 public BuyStock(Stock stock){  
 this.stock = stock;  
 }  
  
 public void execute() {  
 stock.buy();  
 }  
}  
  
public class SellStock implements *Order* {  
 private Stock stock;  
  
 public SellStock(Stock stock) {  
 this.stock = stock;  
 }

public void execute() {  
 stock.sell();  
 }  
}  
  
public class Broker {  
 private *List*<*Order*> orderList = new ArrayList<>();  
  
 public void takeOrder(*Order* order) {  
 orderList.add(order);  
 }  
  
 public void placeOrders() {  
  
 for (*Order* order : orderList) {  
 order.execute();  
 }  
 orderList.clear();  
 }  
}  
  
public class Demo {  
 public static void main(String[] args) {  
 Stock abcStock = new Stock();  
  
 BuyStock buyStockOrder = new BuyStock(abcStock);  
 SellStock sellStockOrder = new SellStock(abcStock);  
  
 Broker broker = new Broker();  
 broker.takeOrder(buyStockOrder);  
 broker.takeOrder(sellStockOrder);  
  
 broker.placeOrders();  
 }  
}

JAVA

