```java
/**
 * @author Harley Phung
 * Create a Transaction class for successful trade.
 */
public class Transaction extends TradeProcess implements BuyOrder, SellOrder {
    /**A field that stored unique stock symbol for the transaction */
    private char stockSymbol = ' ';

    /**A field that stored number of shares traded in the transaction */
    private int numShares = 0;

    /**A field that get price per share traded in the transaction */
    private double price = 0.0;

    /**A field that call buyTrader */
    private Trader buyTrader = null;

    /** A field that call sellTrader */
    private Trader sellTrader = null;

    /**A field that stored the transaction number */
    private int transactionNumber = 0;

    /**
     * A constructor that have stock symbol as input
     * @param stockSymbol  stock symbol of the transaction
     * @param numShares  number of shares in this transaction
     * @param price  price per share in this transaction
     * @param buyTrader  show there's buy trader in the transaction
     * @param sellTrader  showd there's sell trader in the transaction
     */
    public Transaction (char stockSymbol,int numShares, double price, Trader
buyTrader, Trader sellTrader, int transactionNumber) {
        this.stockSymbol = stockSymbol;
        this.numShares = numShares;
        this.price = price;
        this.buyTrader = buyTrader;
        this.sellTrader = sellTrader;
        this.transactionNumber = transactionNumber;
    }

    /**
     * A method that returns the stock symbol
     * @return stockSymbol  the stock symbol of the transaction
     */
    public char getStockSymbol() {
        return this.stockSymbol;
    }

    /** A method that returns the number of shares traded in this transaction
      * @return numSharse  the number of shares in the transaction
      */
    public int getNumberShares() {
        return this.numShares;
    }

    /**
     * A method that return the price of the transaction
     * @return price  the price per share in this transaction
     */
    public double getPrice() {
        return this.price;
    }

    /**
     * A method that returns the Buyer in this transaction
     * @return buyTrader  the trader who buy in this transaction
     */
    public Trader getBuyer() {
        return this.buyTrader;
    }

    /**
     * A method that returns the Seller in this transaction
     * @return sellTrader  the trader who sell in this transaction
     */
    public Trader getSeller() {
        return this.sellTrader;
    }

    /**
     * A method that returns the unique number for this transaction
     * @return transactionNumber  the unique number for this transaction
     */
    public int getTransactionNumber() {
        return this.transactionNumber;
    }

    /**
     * toString method format the returned String
     */
    @Override
    public String toString() {
        return this.getStockSymbol() + ", " + this.getNumberShares()
            + ", " + this.getPrice() + ", " + this.getBuyer() + ", "
            + this.getSeller() + ", " + this.getTransactionNumber();
    }

    /**
     * An abstract equals method that compared the two trader's information.
     * @param t  compare the trader
     * @return true  if there's identical transaction
     * @return false  if there's no identical transaction
     */
    @Override
    public boolean equals(Object t){
        if(t instanceof Transaction) {
            Transaction newTransaction = (Transaction)t;
            if(this.getStockSymbol() == newTransaction.getStockSymbol()
            && this.getTransactionNumber() ==
newTransaction.getTransactionNumber()){
                return true;
            }
        }
        return false;
    }
}
```