

```

/**
 * @author Harley Phung
 * Create a BuyLimitOrder for trader to trade with market maker
 */

public class SellLimitOrder extends LimitOrder implements SellOrder{

    /** The stock symbol of the limit order */
    private char stockSymbol = ' ';

    /** The number of shares the investor wants to buy in the limit order */
    private int numShares = 0;

    /** The price per share the investor wants to pay in the limit order */
    private double numPrice = 0.0;

    /** @param allIn determine if the investor have to trade all in the limit
order */
    private boolean allIn = true;

    /** @param dayOrder determine if the trade is happening when market is open */
    private boolean dayOrder = true;

    /** The instance of Trader class */
    private Trader sellTrader = null;

    /**
     * Constructor that take investor's name, Stock Symbol, num ber of shares,
price per share, all or none,
     * dayOrder, and instance of the sell trader as input
     * @param stockSymbol the symbol of the sell limit order
     * @param numShares the number of shares that the trader requested in sell
limit order
     * @param numPrice the price per share that the trader requested in sell limit
order
     * @param allIn determine if the trader have to trade all stock or not
     * @param dayOrder determine if the order is processed in day or not
     * @param sellTrader the instance of the trader who requested sell order
     */
    public SellLimitOrder (char stockSymbol, int numShares, double numPrice,
boolean allIn, boolean dayOrder, Trader sellTrader) {
        this.stockSymbol = stockSymbol;
        this.numShares = numShares;
        this.numPrice = numPrice;
        this.allIn = allIn;
        this.dayOrder = dayOrder;
        this.sellTrader = sellTrader;
    }

    /**
     * Returns the stock symbol of the sell limit order
     * @return stockSymbol the symbol of the sell limit order
     */
    @Override
    public char getStockSymbol() {
        return this.stockSymbol;
    }
}

```

```

/**
 * Returns the number of shares in the sell limit order
 * @return numShares the number of shares is requested in sell limit order
 */
@Override
public int getNumberShares() {
    return this.numShares;
}

/**
 * Changes the number of shares requested in the sell limit order
 * @param numShares new number of shares is requested in sell limit order
 */
@Override
public void setNumberShares(int numShares) {
    this.numShares = numShares;
}

/**
 * Returns the price per share requested in the sell limit order
 * @return numPrice the number of price is requested in sell limit order
 */
@Override
public double getPrice() {
    return this.numPrice;
}

/**
 * Determine whether trade all the shares of the order or not
 * @return allIn determine if the trader has to trade all or not
 */
@Override
public boolean isAllOrNone() {
    return this.allIn;
}

/**
 * Determines if the transaction is in day order
 * @return dayOrder to check if the order is processed in day or not
 */
@Override
public boolean isDayOrder() {
    return this.dayOrder;
}

/**
 * Returns information of the investor
 * @return sellTrader the instance of the trader who makes this order
 */
@Override
public Trader getTrader() {
    if(this.sellTrader instanceof Trader) {
        return this.sellTrader;
    }
    else {
        return null;
    }
}

```

```

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

/**
 * An abstract equals method that compared the two trader's information.
 * @param p compare the trader
 * @return true if there's identical sell limit order
 * @return false if there's no identical sell limit order
 */
@Override
public boolean equals(Object p){
    if(p instanceof SellLimitOrder) {
        SellLimitOrder newSellOrder = (SellLimitOrder)p;
        if(this.getStockSymbol() == newSellOrder.getStockSymbol()
            && this.getNumberShares() == newSellOrder.getNumberShares()
            && this.getPrice() == newSellOrder.getPrice()
            && this.getTrader() == newSellOrder.getTrader()){
            return true;
        }
    }
    return false;
}
}

```