```
/**
 * @author Harley Phung
 * Testing using junit to test the TradeProcess hierachy
 */
import org.junit.*;
import static org.junit.Assert.*;
import java.util.NoSuchElementException;

public class TradeProcessHierachyTester {
    Trader trader1 = new Trader("Harley");
    Trader trader2 = new Trader("Harold");
    Trader trader3 = new Trader("Harley");
    MarketMaker marketMaker1 = new MarketMaker("Jason", 1000, 100.0);
    MarketMaker marketMaker2 = new MarketMaker("Harold", 500, 20.0);

    BuyLimitOrder bLimitOrder = new BuyLimitOrder('A', 100, 50, true, true,
trader1);
    SellLimitOrder sLimitOrder = new SellLimitOrder('B', 400, 32, true, true,
trader2);
    MarketBuyOrder bMarketOrder = new MarketBuyOrder('C', 2000, 100.0,
marketMaker1);
    MarketSellOrder sMarketOrder = new MarketSellOrder('D', 400, 50, marketMaker2);
    Transaction successOrder = new Transaction('E', 100, 50, trader1, marketMaker1,
23);

    /**
     * Test the getStockSymbol() method
     */
    @Test
    public void testGetStockSymbol() {
        assertEquals('A', bLimitOrder.getStockSymbol()); //Test BuyLimitOrder
        assertEquals('B', sLimitOrder.getStockSymbol()); //Test SellLimitOrder
        assertEquals('C', bMarketOrder.getStockSymbol()); //Test MarketBuyOrder
        assertEquals('D', sMarketOrder.getStockSymbol()); //Test MarketSellOrder
        assertEquals('E', successOrder.getStockSymbol()); //Test Transaction
    }

    /**
     * Test the getNumberShares() method
     */
    @Test
    public void testGetNumberShares() {
        assertEquals(100, bLimitOrder.getNumberShares()); //Test BuyLimitOrder
        assertEquals(400, sLimitOrder.getNumberShares()); //Test SellLimitOrder
        assertEquals(2000, bMarketOrder.getNumberShares()); //Test MarketBuyOrder
        assertEquals(400, sMarketOrder.getNumberShares()); //Test MarketSellOrder
        assertEquals(100, successOrder.getNumberShares()); //Test Transaction
    }

    /**
     * Test the setNumberShares() method
     */
    @Test
    public void testSetNumberShares() {
        bLimitOrder.setNumberShares(150);
        assertEquals(150, bLimitOrder.getNumberShares()); // Test BuyLimitOrder
        sLimitOrder.setNumberShares(450);
        assertEquals(450, sLimitOrder.getNumberShares()); //Test SellLimitOrder
        bMarketOrder.setNumberShares(2500);
```

```
        assertEquals(2500, bMarketOrder.getNumberShares()); //Test MarketBuyOrder
        sMarketOrder.setNumberShares(250);
        assertEquals(250, sMarketOrder.getNumberShares()); //Test MarketSellOrder
    }

    /**
     * Test the getPrice() method
     */
    @Test
    public void testGetPrice() {
        assertEquals(50, 50, bLimitOrder.getPrice()); //Test BuyLimitOrder
        assertEquals(32, 32, sLimitOrder.getPrice()); //Test SellLimitOrder
        assertEquals(100, 100, bMarketOrder.getPrice()); //Test MarketBuyOrder
        assertEquals(50, 50, sMarketOrder.getPrice()); //Test MarketSellOrder
        assertEquals(50, 50, successOrder.getPrice()); //Test Transaction
    }

    /**
     * Test the isAllOrNone() method
     */
    @Test
    public void testIsAllOrNone() {
        assertTrue(bLimitOrder.isAllOrNone() == true); // Test true on
BuyLimitOrder
        assertFalse(bLimitOrder.isAllOrNone() == false); //Test false on
BuyLimitOrder
        assertTrue(sLimitOrder.isAllOrNone() == true); //Test true on
SellLimitOrder
        assertFalse(sLimitOrder.isAllOrNone() == false); //Test false on
SellLimitOrder
    }

    /**
     * Test the isDayOrder() method
     */
    @Test
    public void testIsDayOrder() {
        assertTrue(bLimitOrder.isDayOrder() == true); //Test true on BuyLimitOrder
        assertFalse(bLimitOrder.isDayOrder() == false); //Test false on
BuyLimitOrder
        assertTrue(sLimitOrder.isDayOrder() == true); //Test true on SellLimitOrder
        assertFalse(sLimitOrder.isDayOrder() == false); //Test false on
SellLimitOrder
    }

    /**
     * Test the getTrader() method
     */
    @Test
    public void testGetTrader() {
        //Trader
        assertEquals(trader1, bLimitOrder.getTrader()); //Test BuyLimitOrder
        assertEquals(trader2, sLimitOrder.getTrader()); //Test SellLimitOrder
        //Market maker
        assertEquals(marketMaker1, bMarketOrder.getTrader()); //Test MarketBuyOrder
        assertEquals(marketMaker2, sMarketOrder.getTrader());  //Test
MarketSellOrder
    }
```

```
    /**
     * Test the getBuyer() method in Transaction
     */
    @Test
    public void testGetBuyer() {
        Transaction successOrder2 = new Transaction('L', 235, 26, trader3,
marketMaker2, 27);
        Transaction successOrder3 = new Transaction('K', 457, 436,trader2,
marketMaker1, 45);
        assertEquals(trader1, successOrder.getBuyer());
        assertEquals(trader3, successOrder2.getBuyer());
        assertEquals(trader2, successOrder3.getBuyer());
    }

    /**
     * Test the getSeller() method in Transaction
     */
    @Test
    public void testGetSeller() {
        Transaction successOrder2 = new Transaction('L', 235, 26, trader3,
marketMaker2, 27);
        Transaction successOrder3 = new Transaction('K', 457, 436,trader2,
marketMaker1, 45);
        assertEquals(marketMaker1, successOrder.getSeller());
        assertEquals(marketMaker2, successOrder2.getSeller());
        assertEquals(marketMaker1, successOrder3.getSeller());
    }

    /**
     * Test the getTransactionNumber() method in Transaction
     */
    @Test
    public void testGetTransactionNumber() {
        Transaction successOrder2 = new Transaction('L', 235, 26, trader3,
marketMaker2, 27);
        Transaction successOrder3 = new Transaction('K', 457, 436,trader2,
marketMaker1, 45);
        assertEquals(23, successOrder.getTransactionNumber());
        assertEquals(27, successOrder2.getTransactionNumber());
        assertEquals(45, successOrder3.getTransactionNumber());
    }

    /**
     * Test the toString() method
     */
    @Test
    public void testToString() {
        assertEquals("A, 100, 50.0, true, true, the name of the trader: Harley",
bLimitOrder.toString());
        assertEquals("B, 400, 32.0, true, true, the name of the trader: Harold",
sLimitOrder.toString());
        assertEquals("C, 2000, 100.0, The market maker name is: Jason",
bMarketOrder.toString());
        assertEquals("D, 400, 50.0, The market maker name is: Harold",
sMarketOrder.toString());
        assertEquals("E, 100, 50.0, the name of the trader: Harley, The market
maker name is: Jason, 23", successOrder.toString());
    }
```

```
    /**
     * Test the equals() method
     */
    @Test
    public void testEquals() {
        BuyLimitOrder test1 = new BuyLimitOrder('H', 235, 50, true, true, trader2);
        SellLimitOrder test2 = new SellLimitOrder('B', 400, 32, true, true,
trader2);
        MarketBuyOrder test3 = new MarketBuyOrder('C', 2000, 100.0, marketMaker1);
        MarketSellOrder test4 = new MarketSellOrder('U', 546, 50, marketMaker1);
        Transaction test5 = new Transaction('E', 235, 26, trader1, marketMaker2,
23);
        Transaction test6 = new Transaction('E', 457, 436,trader2, marketMaker2,
45);
        //true
        assertTrue(test2.equals(sLimitOrder));
        assertTrue(test3.equals(bMarketOrder));
        assertTrue(test5.equals(successOrder));

        //false
        assertFalse(test1.equals(bLimitOrder));
        assertFalse(test1.equals(bMarketOrder));
        assertFalse(test1.equals(sMarketOrder));
        assertFalse(test2.equals(bLimitOrder));
        assertFalse(test2.equals(bMarketOrder));
        assertFalse(test2.equals(sMarketOrder));
        assertFalse(test3.equals(bLimitOrder));
        assertFalse(test3.equals(sLimitOrder));
        assertFalse(test3.equals(sMarketOrder));
        assertFalse(test4.equals(bLimitOrder));
        assertFalse(test4.equals(sLimitOrder));
        assertFalse(test4.equals(bMarketOrder));
        assertFalse(test6.equals(successOrder));
    }


}
```