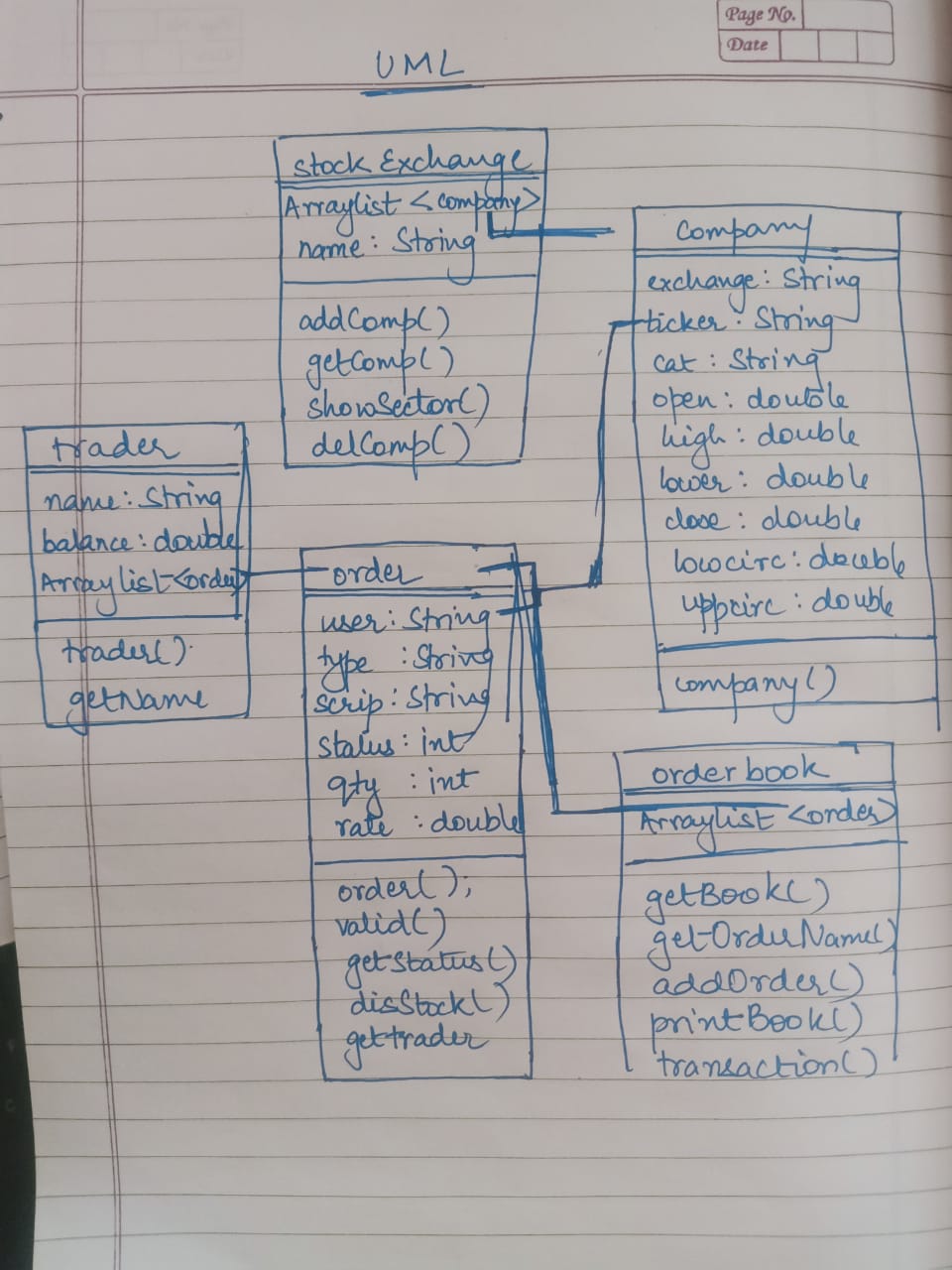
Name : Diya Sachdev

Roll Number : 1910110140

Submission : GL2 CSD203



CODE

//brokerage firm

package gl2;  
  
import javax.swing.\*;  
import java.io.\*;  
import java.util.\*;  
  
public class brokerageFirm {  
  
 public static void main(String[] args) {  
 stockExchange BSE = new stockExchange("BSE");  
 stockExchange NSE = new stockExchange("NSE");  
  
 System.*out*.println(" Reading from file1");  
  
 *readFile1*(BSE,NSE);  
  
 System.*out*.println("Reading from file 2");  
 *readFile2*();  
 }  
  
 public static void readFile1(stockExchange BSE, stockExchange NSE) {  
 JFileChooser op = new JFileChooser(" ");  
 op.setFileSelectionMode(JFileChooser.FILES\_AND\_DIRECTORIES);  
 if (op.showOpenDialog(null) == JFileChooser.APPROVE\_OPTION) {  
 File f1 = op.getSelectedFile();  
  
 try {  
  
 Scanner file = new Scanner(f1);  
 ArrayList<order> o2 = new ArrayList<>();  
 ArrayList<trader> tr = new ArrayList<>();  
 orderBook oBook = new orderBook();  
  
  
 try {  
 BufferedWriter out = new BufferedWriter(new FileWriter("outfilename.txt"));  
  
 int a =0;  
 while (file.hasNextLine()) {  
 String line = file.nextLine();  
 if (line.startsWith("Add scrip:")) {  
 String[] pos = line.split("[;:, ]", 0);  
 int na = (int) (Math.random() \* 2);  
 String className;  
 if (na == 0)  
 {  
 BSE.addComp(new company("BSE", pos[3], pos[7], Double.parseDouble(pos[10]), Double.parseDouble(pos[13]), Double.parseDouble(pos[16]), Double.parseDouble(pos[19])));  
 className = "BSE";  
 }  
 else  
 {  
 NSE.addComp(new company("NSE", pos[3], pos[7], Double.parseDouble(pos[10]), Double.parseDouble(pos[13]), Double.parseDouble(pos[16]), Double.parseDouble(pos[19])));  
 className = "NSE";  
 }  
  
 out.write("Added scrip: "+pos[3]+ " with a new instantiation of "+ className + " stockExchange");  
 out.write("\n");  
  
 } else if (line.startsWith("Add user:")) {  
 String[] pos = line.split("[;:, {}]", 0);  
 ArrayList<order> o1 = new ArrayList<>();  
 for (int i = 0; i < pos.length; i++) {  
 if (i >= 9) {  
 if (pos.length > 10) {  
 o1.add(new order(BSE, NSE, tr, oBook, pos[3], "none", pos[i + 1], Integer.parseInt(pos[i + 2]), 0));  
 }  
 i = i + 2;  
  
 }  
  
 }  
 tr.add(new trader(pos[3], Integer.parseInt(pos[6]), o1));  
 out.write("Added user: "+pos[3]+ " with a new instantiation of Trader");  
 out.write("\n");  
  
 } else if (line.startsWith("Place order,")) {  
 String[] pos = line.split("[;:, ]", 0);  
  
 if(a==0) {  
 out.write("\n");  
 out.write("Market opens");  
 out.write("\n");  
 }  
  
 order o1 = new order(BSE, NSE, tr, oBook, pos[5], pos[9], pos[13], Integer.parseInt(pos[16]), Double.parseDouble(pos[20]));  
 o2.add(o1);  
if(o1.getStatus() == 10)  
 out.write("Order rejected for user: "+pos[5]+ ", type: "+pos[9]+", scrip: "+pos[13]+", qty: "+pos[16]+", rate: "+pos[20]+" : Lower circuit violation");  
else if(o1.getStatus() == 11)  
 out.write("Order rejected for user: "+pos[5]+ ", type: "+pos[9]+", scrip: "+pos[13]+", qty: "+pos[16]+", rate: "+pos[20]+" : Insufficient funds");  
else if(o1.getStatus() == 12)  
 out.write("Order rejected for user: "+pos[5]+ ", type: "+pos[9]+", scrip: "+pos[13]+", qty: "+pos[16]+", rate: "+pos[20]+" : Upper circuit violation");  
else  
 out.write("Order placed for user: "+pos[5]+ ", type: "+pos[9]+", scrip: "+pos[13]+", qty: "+pos[16]+", rate: "+pos[20]);

out.write("\n");  
 a++;  
 }  
  
 else if (line.startsWith("Show Orderbook"))  
 {  
 out.write("\n");  
 out.write("Orderbook: \n");  
 oBook.printBook(out);  
 }  
  
 else if (line.startsWith("Execute"))  
 {  
 out.write("\n");  
 out.write("Executed transactions:\n");  
 oBook.transaction(tr,out);  
 }  
  
 else if (line.startsWith("Show sector:"))  
 {  
 out.write("\n");  
 String[] pos = line.split("[;:, ]", 0);  
 out.write("Scrips listed in sector: "+pos[3]+"\n");  
 // System.out.println(pos[3]);  
 BSE.showSector(pos[3],out);  
 NSE.showSector(pos[3],out);  
 }  
  
 else if (line.startsWith("Delete scrip:")) {  
 out.write("\n");  
 String[] pos = line.split("[;:, ]", 0);  
 // System.out.println(pos[3]);  
 for (int i = 0; i < BSE.getComp().size(); i++) {  
 if ((BSE.getComp()).get(i).ticker.equals(pos[3])) {  
 out.write("Deleted scrip: "+(BSE.getComp()).get(i).ticker+"\n");  
  
 BSE.delComp((BSE.getComp()).get(i));  
 }  
 }  
 for (int i = 0; i < NSE.getComp().size(); i++) {  
 if ((NSE.getComp()).get(i).ticker.equals(pos[3])) {  
 out.write("Deleted scrip: "+(NSE.getComp()).get(i).ticker+"\n");  
  
 NSE.delComp((NSE.getComp()).get(i));  
 }  
 }  
 }  
 else if (line.startsWith("Delete User:")) {  
 out.write("\n");  
 String[] pos = line.split("[;:, ]", 0);  
 for (int i = 0; i < tr.size(); i++) {  
 if (tr.get(i).name.equals(pos[3])) {  
 out.write("Deleted user: "+tr.get(i).name+"\n");  
  
 tr.remove(tr.get(i));  
 break;  
 }  
 }  
 }  
 else if (line.startsWith("Show Scrips")) {  
 out.write("\n");  
 printComp(BSE,out);  
 printComp(NSE,out);  
 }  
 else if (line.startsWith("Show Users")) {  
 out.write("\n");  
 printTrader(tr,out);  
 }  
 else if (line.startsWith("Exit")) {  
 out.write("\nMarket closes");  
 break;  
 }  
 }  
 out.close();  
 }  
 catch (IOException e) {  
 JOptionPane.showMessageDialog(null, "File not Found");  
 }  
  
 } catch (IOException e) {  
 JOptionPane.showMessageDialog(null, "File not Found");  
 }  
 }  
 }  
  
 public static void printComp(stockExchange SE,BufferedWriter out) throws IOException  
 {  
 out.write("Scrips: \n");  
 for (int i = 0; i < SE.comp.size(); i++) {  
 out.write(SE.comp.get(i).ticker + ", sector: "+SE.comp.get(i).cat+", O:"+SE.comp.get(i).open +", H:"+SE.comp.get(i).high+", L:"+SE.comp.get(i).lower +", C:"+SE.comp.get(i).close);  
 out.write("\n");  
  
 }  
 }  
  
 public static void printTrader(ArrayList<trader> tr,BufferedWriter out) throws IOException{  
 int j=0;  
 out.write("Users: \n");  
 String list=" ";  
  
 for (int i = 0; i < tr.size(); i++)  
 {  
 while(j<((tr.get(i)).stock.size()))  
 {  
 list += ((tr.get(i).stock).get(j++).disStock());  
 list += ", ";  
 }  
  
 out.write("user: "+tr.get(i).name+", funds: "+tr.get(i).balance+", holding: { "+list + " }");  
 out.write("\n");  
 j=0;  
 list = " ";  
 // System.out.println("user: "+tr.get(i).name+", funds: "+tr.get(i).balance+", holding: { "+(tr.get(i).stock).get(j).disStock());  
 }  
 }  
  
 public static void readFile2() {  
 JFileChooser op = new JFileChooser(" ");  
 op.setFileSelectionMode(JFileChooser.FILES\_AND\_DIRECTORIES);  
 if (op.showOpenDialog(null) == JFileChooser.APPROVE\_OPTION) {  
 File f1 = op.getSelectedFile();  
 String delimiter = ",";  
  
 try  
 {  
 FileReader fr = new FileReader(f1);  
 BufferedReader br = new BufferedReader(fr);  
 String line = "";  
  
 String[] tempArr;  
 double[] open = new double[15];  
 double[] close = new double[15];  
 double[] last = new double[15];  
  
 int i=0;  
 double sum=0,max=0,min=1000,profit=0;  
 line = br.readLine();  
 while((line = br.readLine()) != null)  
 {  
 tempArr = line.split(delimiter);  
 /\* for(String tempStr : tempArr)  
 {  
 System.out.println(tempArr[3]);  
 }\*/  
  
 open[i]= Double.parseDouble(tempArr[3]);  
 last[i]= Double.parseDouble(tempArr[6]);  
 close[i]= Double.parseDouble(tempArr[7]);  
 i++;  
  
 }  
 for(i=0; i<15; i++)  
 {  
 sum = sum+last[i];  
  
 if(max < close[i])  
 max = close[i];  
 if(min > close[i])  
 min = close[i];  
  
 if(open[i]<close[i])  
 profit = profit + (close[i]-open[i]);  
 else  
 profit = profit + (open[i]-close[i]);  
  
 }  
  
 System.*out*.println("Average: "+sum/15);  
 System.*out*.println("Max drawdown: "+(max-min));  
 System.*out*.println("Max return potential percentage: "+ (profit/open[0])\*100);  
 }  
 catch(IOException e)  
 {  
 e.printStackTrace();  
 }  
  
 }  
 }  
}

// stock exchange

package gl2;  
import java.io.BufferedWriter;  
import java.io.IOException;  
import java.util.ArrayList;  
  
  
public class stockExchange {  
 ArrayList<company> comp = new ArrayList<>();  
 String name;  
  
 stockExchange(String na1)  
 {this.name = na1;}  
  
 public void addComp(company comp1)  
 {  
 (this.comp).add(comp1);  
 }  
  
 public ArrayList<company> getComp()  
 {  
 return (this.comp);  
 }  
  
 public void showSector(String sec, BufferedWriter out) throws IOException  
 {  
 for(int i=0; i<comp.size(); i++)  
 {  
 if((comp.get(i).cat).equals(sec))  
 {  
 out.write(comp.get(i).ticker +", OHLC<> = "+ comp.get(i).open+" , " + comp.get(i).high+" , " + comp.get(i).lower+" , " + comp.get(i).close);  
 out.write("\n");  
  
 }  
 }  
 }  
  
 public void delComp(company comp1)  
 {  
 for(company c1: comp)  
 {  
 if (c1 == comp1) {  
  
 comp.remove(c1);  
 break;  
 }  
  
 }  
 }  
  
}

// company

package gl2;  
  
public class company {  
 String exchange;  
 String ticker;  
 String cat;  
 double open;  
 double high;  
 double lower;  
 double close;  
  
 double lowcirc;  
 double uppcirc;  
  
 company(String ex,String tick1, String cat1, double o1, double h1, double l1, double c1)  
 {  
 this.exchange = ex;  
 this.ticker = tick1;  
 this.cat = cat1;  
 this.open = o1;  
 this.high = h1;  
 this.lower = l1;  
 this.close = c1;  
  
 this.lowcirc = this.close - 0.1\*this.close;  
 this.uppcirc = this.close + 0.1\*this.close;  
 }  
}

// trader

package gl2;  
  
import java.util.ArrayList;  
import java.util.UUID;  
  
public class trader {  
 String name;  
 String ID;  
 Double balance;  
 ArrayList<order> stock = new ArrayList<>();  
  
 trader(String na1, double bal, ArrayList<order> stock1)  
 {  
 this.name = na1;  
 this.balance = bal;  
 this.stock = stock1;  
 this.ID = UUID.*randomUUID*().toString();  
 }  
  
 public String getID() {  
 return ID;  
 }  
  
  
}

// order

package gl2;  
  
import java.io.BufferedWriter;  
import java.io.IOException;  
import java.util.ArrayList;  
  
public class order {  
 String user;  
 String type;  
 String scrip;  
 int status;  
 int qty;  
 double rate;  
  
 order(stockExchange BSE,stockExchange NSE,ArrayList<trader> tr,orderBook oBook,String user1, String type1, String scrip1, int qty1, double rate1)  
 {  
 this.user = user1;  
 this.type = type1;  
  
 this.scrip = scrip1;  
 this.qty = qty1;  
 this.rate = rate1;  
 this.status = 0;  
  
 if(rate1!=0)  
 {  
 valid(BSE,NSE,tr,oBook);  
 }  
  
 }  
 order(String user1, String type1, String scrip1, int qty1, double rate1){  
 this.user = user1;  
 this.type = type1;  
  
 this.scrip = scrip1;  
 this.qty = qty1;  
 this.rate = rate1;  
 }  
  
 public void valid(stockExchange BSE, stockExchange NSE,ArrayList<trader> tr,orderBook oBook)  
 {  
 ArrayList<company> Bcomp = BSE.getComp();  
 ArrayList<company> Ncomp = NSE.getComp();  
  
 //System.out.println(this.scrip);  
 for(int i=0; i<Bcomp.size(); i++)  
 {  
 if((Bcomp.get(i)).ticker.equals(this.scrip))  
 {  
 if(this.rate < (Bcomp.get(i)).lowcirc)  
 this.status =10;  
  
 else if(this.qty\*this.rate > *gettrader*(tr,this.user).balance && this.type.equals("buy"))  
 this.status =11;  
  
 else if(this.rate > (Bcomp.get(i)).uppcirc)  
 this.status =12;  
  
 else {  
 oBook.addOrder(new order(this.user,this.type,this.scrip,this.qty,this.rate));  
 }  
 }  
 }  
  
 for(int i=0; i<Ncomp.size(); i++)  
 {  
 if((Ncomp.get(i)).ticker.equals(this.scrip))  
 {  
  
 if(this.rate < (Ncomp.get(i)).lowcirc)  
 this.status =10;  
  
 else if(this.qty\*this.rate > *gettrader*(tr,this.user).balance && this.type.equals("buy"))  
 this.status =11;  
  
 else if(this.rate > (Ncomp.get(i)).uppcirc)  
 this.status =12;  
  
 else {  
 oBook.addOrder(new order(this.user,this.type,this.scrip,this.qty,this.rate));  
 }  
 }  
 }  
 }  
  
 public int getStatus()  
 {  
 return this.status;  
 }  
  
 public String disStock()  
 {  
 return (this.scrip+":"+this.qty);  
 }  
  
 public static trader gettrader(ArrayList<trader> tr, String user)  
 {  
 int i;  
 for( i=0; i<tr.size(); i++) {  
 // System.out.println(i + " " + tr.get(i).name + " USER "+user);  
 if ((tr.get(i).name).equals(user)) {  
 {  
 // System.out.println(i + " " + tr.get(i).name);  
 return tr.get(i);  
 }  
 }  
 }  
 return tr.get(0);  
 }  
  
}  
  
class orderBook  
{  
 ArrayList<order> book = new ArrayList<>();  
  
 public ArrayList<order> getBook() {  
 return book;  
 }  
  
 public String getOrderName(int i){  
 return book.get(i).scrip;  
 }  
  
 public void addOrder(order o1){  
 book.add(o1);  
  
 }  
 public void printBook(BufferedWriter out) throws IOException {  
 for(int i=0; i<book.size(); i++)  
 {  
 out.write(book.get(i).type +" Order "+ book.get(i).scrip+":"+book.get(i).qty+" at "+book.get(i).rate);  
 out.write("\n");  
 }  
 }  
  
 public void transaction(ArrayList<trader> tr,BufferedWriter out)throws IOException  
 { //funds check  
  
 for(int i=0; i<book.size();i++)  
 {  
 // System.out.println(book.get(i).scrip);  
 for(int j=i+1; j<book.size();j++)  
 {  
  
 if(book.get(i).scrip.equals(book.get(j).scrip ))//&& book.get(i).type!=book.get(j).type)  
 {  
 if(book.get(i).type.equals("sell") )  
 {  
  
 int k1=0, k2=0;  
 trader buyer = book.get(j).*gettrader*(tr,(book.get(j)).user);  
 trader seller = book.get(i).*gettrader*(tr,(book.get(i)).user);  
 // System.out.println("buyer: "+buyer.name +"stock: "+buyer.stock.size());  
 // System.out.println("seller: "+seller.name);  
  
 for(k1=0; k1< buyer.stock.size(); k1++) {  
 if ((buyer.stock.get(k1).scrip).equals(book.get(j).scrip)) {  
  
 break;  
 }  
 }  
 for(k2=0; k2< seller.stock.size(); k2++)  
 {  
 //System.out.println(seller.stock.get(k2).scrip);  
 if((seller.stock.get(k2).scrip).equals(book.get(j).scrip))  
 {  
 break;  
 }  
 }  
 buyer.balance = buyer.balance - book.get(i).rate\*book.get(i).qty;  
 seller.balance = seller.balance + book.get(i).rate\*book.get(i).qty;  
 // System.out.println("k1: "+k1 + " k2: "+k2);  
  
 (seller.stock.get(k2)).user=buyer.name;  
 buyer.stock.add(seller.stock.get(k2));  
 seller.stock.remove(k2);  
 // 25 qty of scrip:SBI sold for INR 195; Buyer: Jaydeep, Seller: Nusrat  
 out.write(book.get(i).qty+" qty of scrip:"+book.get(k1).scrip+" sold for INR "+book.get(i).rate+"; buyer: "+buyer.name+", Seller: "+seller.name);  
 out.write("\n");  
  
 book.remove(i);  
 //book.remove(j);  
 }  
 else if(book.get(i).type.equals("buy") && book.get(j).rate<=book.get(i).rate)  
 {  
  
 int k1=0, k2=0;  
 trader buyer = book.get(i).*gettrader*(tr,(book.get(i)).user);  
 trader seller = book.get(j).*gettrader*(tr,(book.get(j)).user);  
 // System.out.println("buyer: "+buyer.name +"stock: "+buyer.stock.size());  
 // System.out.println("seller: "+seller.name);  
  
 for(k1=0; k1< buyer.stock.size(); k1++) {  
 if ((buyer.stock.get(k1).scrip).equals(book.get(i).scrip)) {  
  
 break;  
 }  
 }  
 for(k2=0; k2< seller.stock.size(); k2++)  
 {  
 //System.out.println(seller.stock.get(k2).scrip);  
 if((seller.stock.get(k2).scrip).equals(book.get(i).scrip))  
 {  
 break;  
 }  
 }  
 buyer.balance = buyer.balance - book.get(j).rate\*book.get(j).qty;  
 seller.balance = seller.balance + book.get(j).rate\*book.get(j).qty;  
 // System.out.println("k1: "+k1 + " k2: "+k2);  
  
 (seller.stock.get(k2)).user=buyer.name;  
 buyer.stock.add(seller.stock.get(k2));  
 seller.stock.remove(k2);  
 // 25 qty of scrip:SBI sold for INR 195; Buyer: Jaydeep, Seller: Nusrat  
 out.write(book.get(j).qty+" qty of scrip:"+book.get(k1).scrip+" sold for INR "+book.get(j).rate+"; buyer: "+buyer.name+", Seller: "+seller.name);  
 out.write("\n");  
  
 book.remove(i);  
 //book.remove(j);  
 }  
 }  
  
  
 }  
 }  
 // else  
 // {System.out.println("NO ");}  
  
 }  
  
  
}

OUTPUT

